PRÁCTICA 3

UNIVERSIDAD DE GRANADA

FLORIN EMANUEL TODOR GLIGA
PEDRO ANTONIO PRIETO FERNÁNDEZ
MIGUEL VELASCO FERNÁNDEZ
GABRIEL OLIVEROS JIMÉNEZ
LAURA RIERA OJEA
ALEXIS CALVENTE GALVÍN

ÍNDICE

1. Descripción del Sistema

Nuestra empresa ha sido encargada de desarrollar una plataforma en línea que facilite el comercio electrónico entre compradores y vendedores, ofreciendo una experiencia similar a la de las grandes plataformas existentes en el mercado. Este sistema está diseñado para gestionar eficazmente los procesos de registro de usuarios, gestión de productos, control de stock, procesos de compra y venta, métodos de pago y sistema de reseñas, asegurando una experiencia de usuario óptima y segura.

Para el acceso en la plataforma es obligatorio estar registrado en ella, sin dicho requisito **no** se permitirá acceder a la plataforma.

A continuación vamos a describir los siguientes subsistemas que forman el sistema de información de la plataforma.

Gestión de Usuarios:

El sistema permite a los usuarios **registrarse o iniciar sesión** en la plataforma, implementando una gestión que considera diferentes escenarios:

- **Registro de Usuarios**
  - Los usuarios pueden **crear una cuenta** proporcionando información como nombre de usuario, correo electrónico y contraseña.
  - El sistema valida que el correo electrónico no esté asociado a ninguna cuenta existente, ya sea activa o deshabilitada. Si el correo electrónico ya está en uso, se informa al usuario y se solicita que utilice uno diferente.

- **Inicio de sesión:**
  - Los **usuarios registrados** pueden acceder al sistema ingresando su correo electrónico y contraseña. Si las credenciales son correctas y la cuenta está activa, se concede acceso.
  - Si la cuenta está **deshabilitada**:Se informa al usuario que su cuenta ha sido deshabilitada y que no es posible acceder ni registrarse nuevamente con el mismo correo electrónico. Se le invita a registrarse utilizando un correo electrónico diferente si desea volver a utilizar el sistema.

- **Deshabilitación de cuenta:**
  - Los usuarios autenticados tienen la opción de **deshabilitar permanentemente su cuenta.** Al confirmar esta acción, el sistema cambia el **estado de la cuenta a deshabilitada**.
  - Si está deshabilitada: La cuenta no puede ser reactivada y el usuario no podrá registrarse nuevamente con el mismo correo electrónico.

- **Recuperación de contraseña:**

  - Si un usuario olvida su contraseña, puede **solicitar su recuperación** ingresando su correo electrónico. Si la cuenta está activa el sistema envía un token para restablecer la contraseña de manera segura. Si está deshabilitada se informa al usuario que no es posible recuperar la contraseña y se le indica que debe registrarse con un nuevo correo distinto.

- **Modificación de datos personales:**
  - Los usuarios autenticados pueden **actualizar su información personal**, como nombre de usuario, correo electrónico y dirección.

Gestión de Productos:

La plataforma ofrece a los usuarios la opción de buscar productos para su compra o, en el caso de los vendedores, añadir productos al catálogo. El sistema implementa una gestión inteligente de productos que considera diferentes escenarios:

- Productos sin stock/deshabilitados:
- Se muestran en la plataforma para informar a los usuarios de su existencia, pero no permiten ser añadidos al carrito.
- Si un producto sin stock estaba previamente añadido al carrito, se muestra al usuario pero no permite completar el pago del mismo.

Para gestionar el stock, cada producto tiene una variable "cantidad" que indica el número de unidades disponibles. Para que un producto se considere en stock, su cantidad debe ser mayor que cero.
Al añadir productos al carrito o realizar un pedido, el sistema verifica esta variable para asegurar la disponibilidad y evitar inconvenientes durante el proceso de compra. Si un vendedor desea retirar un producto del catálogo, puede establecer su cantidad a cero (eliminar stock), indicando que el producto ya no está disponible para la venta, o también puede reducir un poco el stock disponible para la venta. El sistema disminuirá automáticamente el stock disponible cuando se realice una venta de ese producto.

Gestión de Carrito de Compra:

El subsistema de **Gestión de Carrito de Compras** permite a los usuarios interactuar con los productos seleccionados para su compra, permitiendo añadir productos, borrarlos, modificar la cantidad de productos en el carrito y ver los precios y subtotal del carrito. Este subsistema se relaciona estrechamente con otros subsistemas, como el de **Gestión de Productos** y **Gestión de Pedidos**, para asegurar una integración fluida.

El subsistema permite a los usuarios agregar productos al carrito especificando una cantidad. Esta funcionalidad depende de la interacción con el subsistema de **Gestión de Productos**, que se encarga de verificar el stock disponible en la base de datos antes de permitir la adición; en caso de no haber suficiente stock, el sistema notifica un error al usuario. Una vez que los productos se encuentran en el carrito, el usuario puede visualizarlos junto con el subtotal calculado. Para ello, el subsistema consulta la base de datos para obtener información actualizada de los productos, la cual es gestionada por el

subsistema de **Gestión de Productos** también. Si un producto se encuentra añadido al carrito, pero deja de estar disponible (no queda stock, o queda inhabilitado el producto por la desactivación de la cuenta del vendedor), no se ve alterado el carrito, pero a la hora de proceder con el pedido se notificará al usuario de que el producto ya no está disponible.

El usuario también tiene la opción de modificar la cantidad de productos en el carrito. Para esto, se verifica nuevamente el stock en el subsistema de **Gestión de Productos**, asegurando que la nueva cantidad es válida y está disponible. Además, los usuarios pueden eliminar productos de su carrito. Finalmente, el subsistema ofrece la posibilidad de vaciar el carrito de compras, lo que implica eliminar todos los productos agregados de forma masiva. Cuando se borran los productos del carrito, no se ve modificado el stock de los productos, solo desaparecen de la tabla del carrito en la base de datos. Es cuando se realiza el pedido (subsistema de **Gestión de Pedidos)** cuando se ve alterado el stock de los productos que se compran.

Gestión de Pedidos:

El **Subsistema de Gestión de Pedidos** asegura que el proceso de compra sea eficiente y proporciona al usuario control sobre sus pedidos a lo largo de todo su ciclo de vida. Las funcionalidades clave se implementan con controles precisos para garantizar una experiencia de usuario transparente y ordenada:

-Realizar pedido: Al confirmar los productos en el carrito, el sistema permite al usuario seleccionar el método de pago, método de envío, estado del pedido y la dirección. Esta funcionalidad asegura que la información del pedido quede registrada en la base de datos de manera correcta, y el sistema proporciona una confirmación de la operación exitosa. Además se restan del stock los productos de este pedido.

-Consulta del historial de pedidos: El sistema permite a los usuarios visualizar un historial de sus pedidos previos, accediendo a los registros almacenados en la base de datos. Esta funcionalidad facilita el seguimiento y consulta de compras anteriores.

-Cancelación de pedidos: Si el usuario decide cancelar un pedido, el sistema actualiza el estado de dicho pedido a "cancelado" en la base de datos. El stock vuelve a su estado original.

-Selección de método de envío: Los usuarios tienen la opción de seleccionar entre varios métodos de envío (express,normal,frágil), proporcionando flexibilidad en función de la urgencia o coste. El sistema registra la opción seleccionada en el pedido.

-Confirmación de la recepción del pedido: El sistema permite que el usuario confirme la recepción de los productos. Si el usuario no confirma manualmente la recepción dentro de tres días desde la entrega, el sistema actualiza automáticamente el estado del pedido a "completado".

-Selección de método de pago: El sistema ofrece diversas modalidades de pago, lo que permite al usuario escoger la opción que mejor se ajuste a sus preferencias. Esto contribuye a una experiencia de compra más fluida y personalizada.

Gestión de Métodos de Pago:

Una vez autenticados, los usuarios pueden visualizar y gestionar sus métodos de pago a través de un botón dedicado en la interfaz de usuario.

- El sistema permite **añadir nuevos métodos de pago**, como tarjetas de crédito o cuentas de PayPal, asegurando que la información se almacene de forma segura y confidencial. Hay que comprobar que los detalles proporcionados sean válidos y que no se rechace la tarjeta durante la verificación.

- **Eliminar métodos de pago** durante el proceso de pago. Es importante destacar que la eliminación de métodos de pago solo es posible en el momento de realizar una transacción, no en otros momentos, ya que la interfaz de métodos de pago solo aparece a la hora de ir a realizar una compra. También es importante mencionar que solo el usuario propietario podrá eliminarlo.

- El usuario también puede **solicitar ver sus métodos de pago** guardados en el sistema (solo el usuario propietario), al igual que también puede **ver el historial de transacciones** monetarias para cada pedido.

- Al **realizar el pago del pedido**, es importante comprobar que el estado del pedido sea el correcto al igual que la cantidad a pagar coincida con el del pedido. También destacar que tan solo se puede utilizar un método de pago.

Esta gestión centralizada de métodos de pago facilita transacciones más rápidas y eficientes, mejorando la experiencia del usuario al reducir pasos innecesarios como por ejemplo no tener que añadir un método de pago cada vez que quiera pagar un pedido.

Gestión de Reseñas:

La plataforma cuenta con un sistema de reseñas que permite a los usuarios compartir sus experiencias y opiniones sobre el pedido realizado:

- Realizar reseñas: Una vez que el usuario confirma la recepción del pedido, se habilita la opción para valorar el pedido y dejar un comentario. Un usuario no podrá realizar una reseñas hasta que no se haya comprobado que el pedido que quiere valorar existe(es decir, que el pedido se encuentre almacenado en nuestra base de datos). Solo se podrá realizar una reseña por usuario hacia un pedido (id_pedido), si se elimina tampoco se podrá realizar una reseña. Se podrá modificar la reseña existente o eliminada, pero no añadir una nueva.

- Visualización de reseñas: Las valoraciones y comentarios pueden ser vistos por cualquier usuario en el perfil del usuario , fomentando la transparencia y confianza en la comunidad. Como las reseñas se realizan acerca del pedido y no del producto en concreto, puede que haya varias reseñas que sean comunes a varios vendedores(en el mismo pedido había productos vendidos por diferentes usuarios). Dentro de un usuario podemos ver primero todas las reseñas que tiene un usuario y posteriormente ver alguna reseña de un pedido en concreto.

-Modificación y eliminación de reseñas: Para realizar ambas funcionalidades se debe de comprobar la existencia en la base de datos de la reseña, es decir, que en la base de datos se encuentre valoración != null. En caso de que así sea, se podrá eliminar (colocar los campos de valoración y comentario en null l) o modificar la valoración y comentario.

- Contribución a la comunidad: Este sistema ayuda a otros usuarios a tomar decisiones informadas y permite a los vendedores recibir retroalimentación sobre pedidos.

Conclusión

Este sistema integral de comercio electrónico está diseñado para satisfacer las necesidades tanto de compradores como de vendedores, ofreciendo funcionalidades avanzadas y una gestión eficiente de los diferentes procesos involucrados. Con un enfoque en la usabilidad, seguridad y transparencia, la plataforma busca proporcionar una experiencia excepcional que promueva la confianza y fidelización de los usuarios, posicionándose como una opción líder en el mercado de comercio electrónico.

2. Requisitos Funcionales

2.1. Subsistema de Gestión de Usuarios

Autor: Alexis Calvente Galvín

Administra el proceso de registro y acceso de los usuarios al sistema.

- **RF1.1 Registro de usuarios:** Permite a nuevos usuarios crear una cuenta proporcionando información básica (nombre de usuario, correo electrónico y contraseña). El sistema valida los datos, asegurando que el correo electrónico no esté asociado a ninguna cuenta existente (activa o deshabilitada**)**, y confirma el registro exitoso.
- **RF1.2 Dar de baja usuario:** Permite a los usuarios autenticados deshabilitar permanentemente su cuenta, cambiando su estado a "inactivo" en el sistema. Una vez deshabilitada, la cuenta no puede ser reactivada y el usuario no podrá registrarse nuevamente con el mismo correo electrónico.
- **RF1.3 Modificación de datos de usuario:** Permite a los usuarios autenticados actualizar su información personal (nombre, correo electrónico, contraseña, dirección). El sistema valida y almacena los cambios realizados.
- **RF1.4 Recuperación de contraseña:** Facilita a los usuarios recuperar el acceso en caso de olvidar su contraseña, enviando un enlace o código de verificación al correo electrónico registrado para establecer una nueva contraseña.
- **RF1.5 Iniciar sesión:** Permite a los usuarios registrados acceder al sistema ingresando su correo electrónico y contraseña. El sistema verifica las credenciales y, si son correctas y la cuenta está activa, concede acceso. Si la cuenta está deshabilitada, se le informará que su cuenta ha sido deshabilitada y que no es posible acceder ni registrarse nuevamente con el mismo correo electrónico.

<table>
<tr><td colspan="4">Requisito funcional : Registro de usuario</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF1.1</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permit a nuevos usuarios crear una cuenta en el sistema.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar registro.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE1.1</td></tr>
<tr><td>Composición</td><td>● Nombre de usuario: Cadena de caracteres (50).
● Correo electrónico: Cadena de caracteres (50).
● Contraseña: Cadena de caracteres (50).
● Dirección : Cadena de caracteres (20)</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW1.1</td></tr>
<tr><td>Composición</td><td>● ID de usuario: Entero. (20)
● Nombre de usuario: Cadena de caracteres. (50)
● Correo electrónico: Cadena de caracteres. (50)
● Contraseña: Cadena de caracteres. (50)
● Dirección: Cadena de caracteres. (200)
● Fecha de registro: Fecha y hora.
● Estado de la cuenta: Activo/Inactivo</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación de registro con su id de usuario.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS1.1</td></tr>
<tr><td>Composición</td><td>Id de usuario.</td></tr>
<tr><td rowspan="2">Restricciones semánticas</td><td rowspan="2"></td><td>Número de referencia</td><td>RS1.1</td></tr>
<tr><td>Descripción</td><td>Validación estricta de correo electrónico único. Si el correo electrónico proporcionado ya existe en la base de datos, ya sea asociado a una cuenta activa o deshabilitada, no se realiza el registro y se informa al usuario que debe utilizar un correo</td></tr>
</table>

			electrónico diferente.
		Requisito funcional relacionado	RF1.1
		Requisito/s de datos relacionados	• RDE1.1 • RDW1.1

<table>
<tr><td colspan="4">Requisito funcional : Dar de baja usuario.</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF1.2</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite a los usuarios autenticados deshabilitar permanentemente su cuenta, cambiando su estado a "inactivo" en el sistema.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar la deshabilitación de la cuenta..</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE1.2</td></tr>
<tr><td>Composición</td><td>• ID de usuario: Entero. (20)
• Contraseña: Cadena de caracteres (50)</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW1.2</td></tr>
<tr><td>Composición</td><td>● Estado de la cuenta: Se actualiza a Inactivo.
● Fecha de desactivación: Fecha y hora en que se realiza la baja.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación de deshabilitación.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="6">Restricciones semánticas</td><td rowspan="6"></td><td>Número de referencia</td><td>RS1.2.1</td></tr>
<tr><td>Descripción</td><td>Verificación de contraseña al dar de baja. Si la contraseña proporcionada no coincide con la almacenada, no se procede a dar de baja al usuario y se informa del error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF1.2</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDE1.2
● RDW1.2</td></tr>
<tr><td>Número de referencia</td><td>RS1.2.2</td></tr>
<tr><td>Descripción.</td><td>Imposibilidad de reutilizar el correo electrónico. Una vez que la cuenta es deshabilitada, el usuario no podrá utilizar el mismo correo electrónico para registrarse nuevamente. Si desea volver a utilizar el sistema, deberá registrarse con un correo</td></tr>
</table>

			electrónico diferente.
		Requisito funcional relacionado	RF1.2
		Requisito/s de datos relacionados	● RDE1.2 ● RDW1.2
		Número de referencia.	**RS1.2.3**
		Descripción.	Deshabilitación permanente. Al dar de baja la cuenta, esta se deshabilita de forma permanente y no puede ser reactivada. Toda la información asociada al usuario se mantiene según las políticas de retención de datos, pero el acceso no es posible.
		Requisito funcional relacionado	RF1.2
		Requisito/s de datos relacionados	● RDE1.2 ● RDW1.2

<table>
<tr><td colspan="4">Requisito funcional : Modificación de datos de usuario.</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF1.3</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite a los usuarios autenticados actualizar su información personal.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar modificación de datos.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE1.3</td></tr>
<tr><td>Composición</td><td>• ID de usuario: Entero
• Datos a modificar (opcional, uno o varios):
- Nombre de usuario: Cadena de caracteres (50).
- Correo electrónico: Cadena de caracteres (50).
- Contraseña: Cadena de caracteres (50).
- Dirección: Cadena de caracteres (200).</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDR1.3</td></tr>
<tr><td>Composición</td><td>● ID de usuario.
● Datos actuales del usuario (los mismos que en RDW1.1)</td></tr>
<tr><td>Número de referencia</td><td>RDW1.3</td></tr>
<tr><td>Composición</td><td>Datos actualizados: Se actualizan los datos modificados.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación de modificación.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS1.3</td></tr>
<tr><td>Composición</td><td>Datos actualizados: Los datos que han sido modificados.</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS1.3</td></tr>
<tr><td>Descripción</td><td>Autenticación requerida para modificar datos. Solo el usuario autenticado puede modificar sus propios datos. Si el ID de usuario proporcionado no coincide con la sesión activa, se deniega la operación.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF1.3</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDE1.3
● RDR1.3</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional : Recuperación de contraseña.</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF1.4</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Se facilita a los usuarios recuperar el acceso a su cuenta en caso de olvidar su contraseña.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar recuperación de contraseña.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE1.4</td></tr>
<tr><td>Composición</td><td>● Correo electrónico: Cadena de caracteres (50).</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDR1.4</td></tr>
<tr><td>Composición</td><td>● Correo electrónico (50).
● Estado de la cuenta: Activo/Inactivo.</td></tr>
<tr><td>Número de referencia</td><td>RDW1.4</td></tr>
<tr><td>Composición</td><td>● Token de recuperación: Cadena de caracteres (50).
● Fecha de expiración del token: Fecha y hora.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir instrucciones para restablecer contraseña.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS1.4</td></tr>
<tr><td>Composición</td><td>Token de recuperación (50).</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS1.4</td></tr>
<tr><td>Descripción</td><td>Token de recuperación válido. El token generado debe ser único y tener una validez limitada en el tiempo. Si el token es inválido o ha expirado, no se permite restablecer la contraseña.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF1.4</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDW1.4
● RDR1.4</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional : Iniciar sesión.</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF1.5</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite a los usuarios registrados acceder al sistema proporcionando su correo electrónico y contraseña.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Ingresar correo electrónico y contraseña para iniciar sesión.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE1.5</td></tr>
<tr><td>Composición</td><td>● Correo electrónico: Cadena de caracteres (50).
● Contraseña: Cadena de caracteres (20).</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR1.5</td></tr>
<tr><td>Composición</td><td>● Correo electrónico (50).
● Contraseña (50).
● Estado de la cuenta (activo/inactivo).</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación de inicio de sesión o notificación de error.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS1.5</td></tr>
<tr><td>Descripción</td><td>● Si las credenciales son incorrectas, no se permite el acceso al sistema y se informa al usuario del error en las credenciales.
● Si la cuenta está deshabilitada, se informa al usuario que su cuenta ha sido deshabilitada y que no es posible acceder ni registrarse nuevamente con el mismo correo electrónico.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF1.5</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDE1.5
● RDR1.5</td></tr>
</table>

2.2. Subsistema de Gestión de Productos

Autor: Miguel Velasco Fernández

Permite a los usuarios añadir y administrar los productos que desean vender.

- **RF2.1 Subir nuevo producto:** Facilita la adición de nuevos productos ingresando id del producto, nombre, cantidad y precio. Los datos se registran en la base de datos, y el sistema confirma la operación o notifica un error.
- **RF2.2 Editar producto:** Permite modificar la información de productos existentes. Actualiza los datos en la base de datos y confirma la actualización o informa de un error.
- **RF2.3 Eliminar producto:** Posibilita la eliminación de la cantidad de producto disponible para la venta. La información del producto no se elimina de la base de datos, solo se deshabilita su vista al usuario.
- **RF2.4 Ver lista de productos propios:** Muestra al usuario(ya sea comprador o vendedor) un listado de los productos que pertenecen al usuario requerido(un usuario se puede buscar a sí mismo) tras consultar la base de datos.
- **RF2.5 Filtrar productos por precio:** Permite visualizar productos según el precio dado, mostrando los resultados filtrados tras la consulta a la base de datos.

<table>
<tr><th colspan="4">Requisito funcional: Subir Producto</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 2.1</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Un usuario sube un nuevo producto</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por agente externo</td><td colspan="2">Añadir un nuevo producto al stock de una tienda</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 2.1</td></tr>
<tr><td>Composición</td><td>Datos de entrada para subir un nuevo producto:
Nombre Producto: cadena caracteres (30)
ID Producto: cadena caracteres (10)
ID Usuario: cadena caracteres (10)
Cantidad: double(>0)
Precio: double</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 2.1</td></tr>
<tr><td>Composición</td><td>Datos que componen producto
Mismos datos que RDE 2.1</td></tr>
<tr><td></td><td></td><td>Número de referencia</td><td>RDR 2.1</td></tr>
<tr><td></td><td></td><td>Composición</td><td>ID Usuario</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por agente externo</td><td colspan="2">Confirmación subida nuevo producto</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td></td><td></td></tr>
<tr><td>Composición</td><td></td></tr>
<tr><td rowspan="3">Restricciones semánticas</td><td rowspan="3"></td><td>Número de referencia</td><td>RS 2.1</td></tr>
<tr><td>Descripción</td><td>Si no existe ningún usuario con ese ID, no se añade el producto y se devuelve error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 2.1</td></tr>
</table>

		Requisito/s de datos relacionados	RDW 2.1 y RDR 2.1

<table>
<tr><th colspan="4">Requisito funcional: Editar producto</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 2.2</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite modificar los datos de un producto que tiene subido el usuario</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Modificar los datos de un producto ya existente</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 2.2</td></tr>
<tr><td>Composición</td><td>Datos de entrada para modificar un producto:
● Mismos datos que RDE 2.1</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 2.2</td></tr>
<tr><td>Composición</td><td>Datos que componen producto
● Mismos datos que RDE 2.1</td></tr>
<tr><td></td><td></td><td>Número de referencia</td><td>RDR 2.2</td></tr>
<tr><td></td><td></td><td>Composición</td><td>ID Producto, ID Usuario</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Se devuelve una confirmación de que los datos han sido modificados</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td></td></tr>
<tr><td>Composición</td><td></td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 2.2.1</td></tr>
<tr><td>Descripción</td><td>Si no existe ningún producto con ese ID, no se realiza la modificación de los datos</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 2.2</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDR 2.2 y RDW 2.2</td></tr>
<tr><td></td><td></td><td>Número de referencia</td><td>RS 2.2.2</td></tr>
</table>

		Descripción	Si el ID usuario no se corresponde con el del usuario que subió el producto o no existe ese ID usuario, no permite el cambio y devuelve error
		Requisito funcional relacionado	RF 2.2

<table>
<tr><th colspan="5">Requisito funcional: Eliminar stock</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF 2.3</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Permite al sistema reducir el stock de un producto cuando este sea vendido y además permite al vendedor reducir o eliminar por completo el stock de un producto(en caso de que el vendedor no quiera seguir vendiendo el producto)</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Elimina la cantidad requerida por el vendedor o el sistema</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 2.3</td></tr>
<tr><td>Composición</td><td>Datos de entrada para eliminar un producto:
ID Producto: cadena caracteres (10)
Nombre Producto: cadena caracteres (30)
Cantidad: entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 2.3</td></tr>
<tr><td>Composición</td><td>Datos que componen producto
Cantidad: entero</td></tr>
<tr><td></td><td></td><td>Número de referencia</td><td>RDR 2.3</td></tr>
<tr><td></td><td></td><td>Composición</td><td>ID Producto, ID Usuario</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Confirmación de que se ha reducido la cantidad disponible del producto</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td></td></tr>
<tr><td>Composición</td><td></td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 2.3.1</td></tr>
<tr><td>Descripción</td><td>Si no existe ningún producto con ese ID, no se deshabilitan los datos y se devuelve error</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 2.3</td></tr>
<tr><td>Requisito/s de datos</td><td>RDR 2.3 y RDW 2.3</td></tr>
</table>

		relacionados	
		Número de referencia	RS 2.3.2
		Descripción	Si la cantidad de producto disponible es menor que la cantidad que el usuario quiere quitar de la venta, entonces la cantidad disponible(stock) se queda a 0.
		Requisitos funcional relacionado	RF 2.3
		Requisitos de datos relacionados	RDR 2.3 y RDW 2.3
		Número de referencia	RS 2.3.3
		Descripción	Si el ID usuario no se corresponde con el del usuario que subió el producto o el ID usuario no existe, no permite el cambio y devuelve error
		Requisitos funcional relacionado	RF 2.3
		Requisitos de datos relacionados	RDR 2.3 y RDW 2.3

<table>
<tr><th colspan="5">Requisito funcional: Mostrar productos</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF 2.4</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Muestra la lista de productos a la venta que tiene un usuario</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Mostrar al usuario un listado de productos</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 2.4</td></tr>
<tr><td>Composición</td><td>Datos de entrada para ver la lista de productos:
ID Usuario: cadena caracteres (10)</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR 2.4</td></tr>
<tr><td>Composición</td><td>Datos que componen producto
Mismos datos que RDE 2.1</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Confirmación del resultado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS 2.4</td></tr>
<tr><td>Composición</td><td>Listado de registros, cada uno con la misma composición que RDE 2.1</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 2.4.1</td></tr>
<tr><td>Descripción</td><td>Si no existe ningún usuario con ese ID, no se leen los datos y se devuelve error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 2.4</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDE 2.4</td></tr>
</table>

<table>
<tr><th colspan="4">Requisito funcional: Mostrar con filtro por precio</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 2.5</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Filtrar productos por su precio para mostrarselos a un usuario</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Mostrar al usuario un listado de los productos que cumplen el filtro</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 2.5</td></tr>
<tr><td>Composición</td><td>Datos de entrada con los que filtrar los productos:
● Precio: int</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR 2.5</td></tr>
<tr><td>Composición</td><td>Datos que componen producto
● Mismos datos que RDE 2.1</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación del resultado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS 2.5</td></tr>
<tr><td>Composición</td><td>Listado de registros, cada uno con la misma composición que RDE 2.1 y que cumplen los filtros aplicados</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 2.5</td></tr>
<tr><td>Descripción</td><td>Si no existe ningún producto que cumpla con los filtros, se devuelve un registro vacío y se devuelve error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 2.5</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDR 2.5</td></tr>
</table>

2.3. Subsistema de Gestión de Carrito de Compras

Autor: Gabriel Oliveros Jiménez

Gestiona el carrito de compras donde los usuarios pueden agregar o eliminar productos.

- **RF3.1 Añadir producto al carrito:** Permite agregar productos al carrito especificando la cantidad. Los datos se almacenan en la base de datos, confirmando la adición o notificando un error.
- **RF3.2 Ver carrito de compras:** Muestra los productos añadidos al carrito consultando la base de datos y muestra el subtotal.
- **RF3.3 Modificar cantidad de un producto en el carrito:** Facilita el ajuste de la cantidad de productos en el carrito, actualizando la información en la base de datos y confirmando la modificación.
- **RF3.4 Eliminar producto del carrito:** Permite quitar productos del carrito, eliminando los datos correspondientes y confirmando la acción.
- **RF3.5 Vaciar carrito de compras:** Ofrece la opción de eliminar todos los productos del carrito, confirmando el vaciado tras borrar los datos de la base de datos.

<table>
<tr><td colspan="4">Requisito funcional: Añadir producto al carrito</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF3.1</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite agregar productos al carrito especificando la cantidad. Los datos se almacenan en la base de datos, confirmando la adición o notificando un error.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Seleccionar cantidad de producto en la página del producto y usar botón de añadir al carrito.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 3.1</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● Cantidad: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDW 3.1</td></tr>
<tr><td>Composición</td><td>● Mismo que RDE3.1</td></tr>
<tr><td>Número de referencia</td><td>RDR 3.1</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación de la adición del producto o notificación de error.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="5">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS3.1.1</td></tr>
<tr><td>Descripción</td><td>Si la cantidad solicitada excede el stock disponible, no se realiza la inserción y se devuelve un error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF3.1</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDW3.1</td></tr>
<tr><td></td><td>Número de</td><td>RS3.1.2</td></tr>
</table>

		referencia	
		Descripción	No se puede agregar un producto que ya esté en el carrito, salta mensaje de error. En lugar de eso, debe modificarse la cantidad.
		Requisito funcional relacionado	RF3.3
		Requisito/s de datos relacionados	RDE3.3, RDW3.3

<table>
<tr><td colspan="4">Requisito funcional: Ver carrito de compras</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF3.2</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Muestra los productos añadidos al carrito, consultando la base de datos y muestra el subtotal.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar visualización del carrito.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE3.2</td></tr>
<tr><td>Composición</td><td>● ID de usuario: número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR3.2</td></tr>
<tr><td>Composición</td><td>● ID de usuario: número entero
Para cada producto:
● ID del producto: número entero
● Nombre: cadena de caracteres (15)
● Precio</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Mostrar los productos añadidos al carrito y el subtotal.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS3.2</td></tr>
<tr><td>Composición</td><td>● Nombre: cadena de caracteres (15)
● Cantidad: número entero
● Precio: número como flotante</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS3.3</td></tr>
<tr><td>Descripción</td><td>El carrito debe mostrar el subtotal exacto.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF3.2</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDR3.2, RDS3.2</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional: Modificar cantidad de un producto en el carrito.</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF3.3</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite ajustar la cantidad de unidades de un producto en el carrito.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar modificación de cantidad de un producto.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE3.3</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● Nueva cantidad: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDW3.3</td></tr>
<tr><td>Composición</td><td>● Mismo que RDE3.3</td></tr>
<tr><td>Número de referencia</td><td>RDR 3.3</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación de la modificación de la cantidad.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS3.3</td></tr>
<tr><td>Descripción</td><td>La cantidad no puede ser negativa ni mayor que el stock disponible.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF3.3</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>● RDW3.3</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional: Eliminar producto del carrito</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF3.4</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite eliminar productos del carrito, eliminando los datos correspondientes y confirmando la acción.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar eliminación de un producto del carrito.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE3.4</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDW3.4</td></tr>
<tr><td>Composición</td><td>● ID de usuario: número entero (para ver tabla de carrito correspondiente al usuario)
● Eliminar entrada del producto en la tabla del carrito</td></tr>
<tr><td>Número de referencia</td><td>RDR 3.4</td></tr>
<tr><td>Composición</td><td>● ID del producto: número entero
● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación de la eliminación</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Descripción</td><td>NA</td></tr>
<tr><td>Requisito funcional relacionado</td><td>NA</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>NA</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional: Vaciar carrito de compras</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF3.5</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Elimina todos los productos del carrito.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar vaciado completo del carrito. (Pulsar botón correspondiente)</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE3.5</td></tr>
<tr><td>Composición</td><td>● ID de usuario: número entero</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDW3.5</td></tr>
<tr><td>Composición</td><td>● ID de usuario: número entero (para ver tabla de carrito correspondiente al usuario)
● Eliminar tabla carrito</td></tr>
<tr><td>Número de referencia</td><td>RDR 3.5</td></tr>
<tr><td>Composición</td><td>● Id de usuario número entero</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación del vaciado del carrito.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS3.5</td></tr>
<tr><td>Descripción</td><td>Si el carrito ya está vacío, se notifica al usuario.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF3.5</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDW3.5</td></tr>
</table>

2.4 Subsistema de Gestión de pedidos

Autor: Pedro Antonio Prieto Fernández

Administra los pedidos realizados por los usuarios tras finalizar una compra.

- **RF4.1 Realizar pedido:** Permite confirmar los productos en el carrito y seleccionar el método de pago para realizar un pedido. Registra los datos del pedido en la base de datos y confirma la operación.
- **RF4.2 Ver historial de pedidos:** Muestra al usuario sus pedidos anteriores consultando la base de datos.
- **RF4.3 Cancelar pedido:** Permite cancelar un pedido seleccionado, actualizando su estado a "cancelado" en la base de datos, confirmando la cancelación y el vendedor se encargará de devolver el producto.
- **RF4.4 Opción para elegir el método de envío:** Te permite elegir entre diferentes métodos de envío estándar, premium, entrega rápida.
- **RF4.5 Confirmar recepción del pedido:** Permite al usuario confirmar la recepción del pedido, actualizando su estado a "completado" en la base de datos, pasados 3 días de la entrega se da el pedido por recepcionado aunque el cliente no active esta funcionalidad.
- **RF4.6 Opción para elegir el método de pago:** Permite elegir entre las diferentes modalidades que se disponen

<table>
<tr><th colspan="4">Requisito funcional: Realizar pedido</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 4.1</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite confirmar los productos en el carrito. Registra los datos del pedido en la base de datos y confirma la operación. Haciendo que disminuya el stock de los productos que se incluyen en el pedido</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar inserción</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 4.1</td></tr>
<tr><td>Composición</td><td>● Dirección: cadena de caracteres (30)
● ID Pedido: cadena de caracteres (10)
● Estado_Pedido: cadena de caracteres (10)
● Tipo de pago: cadena de caracteres (20)
● Método de envío: cadena de caracteres (10)
● ID Usuario: número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 4.1 - RDR 4.1</td></tr>
<tr><td>Composición</td><td>● RDW 4.1 será igual que RDE 4.1
● RDR 4.1
-Carrito: lista de ID producto, cadena de caracteres (10)</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación resultado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
</table>

Restricciones semánticas		**Número de referencia**	RS 4.1
		Descripción	El usuario debe estar dado de alta de lo contrario da error.
		Requisito funcional relacionado	RF 4.1
		Requisito/s de datorelacionados	
		Número de referencia	RS 4.2
		Descripción	El producto debe tener stock.
		Requisito funcional relacionado	RF 4.1
		Requisito/s de datos relacionados	
		Número de referencia	RS 4.3
		Descripción	El producto debe estar habilitado
		Requisito funcional relacionado	RF 4.1
		Requisito/s de datos relacionados	

<table>
<tr><th colspan="5">Requisito funcional: Ver historial de pedidos</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF 4.2</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Muestra al usuario sus pedidos anteriores consultando la base de datos.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Solicitar listado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 4.2</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR 4.2</td></tr>
<tr><td>Composición</td><td>● ID Pedido: cadena de caracteres (10)
● Carrito: lista de ID producto, cadena de caracteres (10)
● Estado: cadena de caracteres (10)
● ID Usuario: número entero</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Mostrar listado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS 4.2</td></tr>
<tr><td>Composición</td><td>LIstado de todos los pedidos</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td></td></tr>
<tr><td>Descripción</td><td></td></tr>
<tr><td>Requisito funcional relacionado</td><td></td></tr>
<tr><td>Requisito/s de datos relacionados</td><td></td></tr>
</table>

<table>
<tr><th colspan="4">Requisito funcional: Cancelar pedido</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 4.3</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite cancelar un pedido seleccionado, actualizando su estado a "cancelado" en la base de datos, confirmando la cancelación. También se reestablece el stock de los productos del producto cancelado.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Seleccionar pedido</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 4.3</td></tr>
<tr><td>Composición</td><td>● Estado: cadena de caracteres (10)

Para cada producto del pedido:

● ID Producto: número entero
● Cantidad producto: número entero
● ID Pedido: número entero
● ID Usuario: número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 4.3</td></tr>
<tr><td>Composición</td><td>● Estado: cadena de caracteres (10)

Para cada producto del pedido:

● ID Producto: número entero
● Nueva cantidad producto: número entero</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Cancelación pedido</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td>Restricciones semánticas</td><td></td><td>Número de referencia</td><td></td></tr>
</table>

		Descripción	
		Requisito funcional relacionado	
		Requisito/s de datos relacionados	

<table>
<tr><th colspan="5">Requisito funcional: Opción para elegir el método de envío</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF 4.4</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Te permite elegir entre diferentes métodos de envío estándar, premium, entrega rápida.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Elegir método de envío</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 4.4</td></tr>
<tr><td>Composición</td><td>● Método de envío: cadena de caracteres (10)
● ID Usuario: número entero
● ID Pedido: número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 4.4</td></tr>
<tr><td>Composición</td><td>Mismo que RDE 4.4</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Confirmación envío elegido</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td></td></tr>
<tr><td>Descripción</td><td></td></tr>
<tr><td>Requisito funcional relacionado</td><td></td></tr>
<tr><td>Requisito/s de datos relacionados</td><td></td></tr>
</table>

<table>
<tr><th colspan="4">Requisito funcional: Confirmación recepción del pedido</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF 4.5</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite al usuario confirmar la recepción del pedido, actualizando su estado a "completado" en la base de datos, pasados 3 días de la entrega se da el pedido por recepcionado aunque el cliente no active esta funcionalidad.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmar recepción</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE 4.5</td></tr>
<tr><td>Composición</td><td>● Estado: cadena de caracteres (10)
● ID usuario: Número entero
● ID pedido: Número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 4.5</td></tr>
<tr><td>Composición</td><td>Mismo que RDE 4.5</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmación de recepción del pedido</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 4.4</td></tr>
<tr><td>Descripción</td><td>Si el usuario no confirma la recepción del pedido en 3 días, el sistema lo da por recepcionado</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 4.5</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td></td></tr>
</table>

<table>
<tr><th colspan="5">Requisito funcional: Opción para elegir el método de pago</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF 4.6</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Se selecciona un método de pago dentro de los disponibles</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Elegir método de pago</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDW 4.6</td></tr>
<tr><td>Composición</td><td>● ID del método de pago: Entero.
● Tipo de método de pago: Cadena de caracteres. (10)
● ID pedido: número entero
● ID usuario: número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR 4.6</td></tr>
<tr><td>Composición</td><td>Mismo que RDE 4.6</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Método de pago aceptado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>NA</td></tr>
<tr><td>Composición</td><td>NA</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS 4.5</td></tr>
<tr><td>Descripción</td><td>El método de pago debe de ser un método de pago válido que exista en la base de datos.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF 4.6</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td></td></tr>
</table>

2.5. Subsistema de Reseñas

Autor: Florín Emanuel Todor Gliga

Facilita a los usuarios la valoración y opinión sobre pedidos.

- **RF5.1 Añadir reseña sobre pedido:** Permite publicar una valoración y comentario sobre un pedido, almacenando la opinión en la base de datos y confirmando la publicación.
- **RF5.2 Editar reseña:** Ofrece la opción de modificar una reseña existente, actualizando los datos en la base de datos y confirmando la actualización.
- **RF5.3 Eliminar reseña:** Permite eliminar una reseña publicada, borrando los datos correspondientes y confirmando la eliminación.
- **RF5.4 Ver reseñas de un pedido:** Muestra las reseñas asociadas a un producto específico tras consultar la base de datos.
- **RF5.5 Ver reseñas de un Usuario:** Muestra todas las reseñas asociadas a un usuario.

<table>
<tr><td colspan="4">Requisito funcional : Añadir reseña sobre un pedido</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF5.1</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite publicar una valoración (1-5) , añadir un comentario sobre un pedido y confirmar la inserción.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar inserción de valoración y comentario.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE5.1</td></tr>
<tr><td>Composición</td><td>Datos almacenados de inserción de reseña sobre un pedido:
- ID_Reseña: Número entero(1-10)
- Valoración : Número entero (1-5)
- Comentario: Cadena de caracteres(500)
- ID Pedido: Número entero
- ID Usuario: Número entero</td></tr>
<tr><td colspan="2" rowspan="3">Datos internos
Requisito de datos</td><td>Número de referencia</td><td>RDW5.1</td></tr>
<tr><td>Composición</td><td>Los mismos datos que el RDE5.1</td></tr>
<tr><td>Número de referencia</td><td>(comprobación de existencia)
RDR5.1 Datos de Lectura</td></tr>
<tr><td colspan="2"></td><td>Composición</td><td>- ID Pedido: Número entero
- ID Usuario: Número entero
- Estado_Pedido : String</td></tr>
<tr><td rowspan="2">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmar envío de reseña.</td></tr>
<tr><td rowspan="5">Restricciones semánticas</td><td rowspan="5"></td><td>Número de referencia</td><td>RS5.1.1</td></tr>
<tr><td>Número de referencia</td><td>RS5.1.2</td></tr>
<tr><td>Descripción</td><td>RS5.1.1
Si no existe el ID Pedido o el ID Usuario (debe de estar registrado), no se inserta la reseña y se devuelve un error.
RS5.1.2
Comprobar si existe la recepción del pedido, en caso de que no exista no se puede realizar la reseña y se devuelve un error.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF5.1</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDE5.1 y RDR5.1</td></tr>
</table>

<table>
<tr><td colspan="5">Requisito funcional: Editar reseña</td></tr>
<tr><td>Número de referencia</td><td colspan="4">RF5.2</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Ofrece la opción de modificar una reseña existente y confirmar la modificación.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">modificar una reseña existente.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE5.2</td></tr>
<tr><td>Composición</td><td>Datos almacenados de modificación de reseña sobre un producto.
- ID_Reseña: Número entero(1-10)
- Valoración : Número entero (1-5)
- Comentario: Cadena de caracteres(500)
- ID Pedido: Número entero
- ID Usuario: Número entero</td></tr>
<tr><td colspan="2" rowspan="4">Datos internos
Requisito de datos</td><td>Número de referencia</td><td>RDW5.2</td></tr>
<tr><td>Composición</td><td>Los mismos datos que RDE5.2.</td></tr>
<tr><td>Número de referencia</td><td>(comprobación de existencia)
RDR5.2 Datos de Lectura</td></tr>
<tr><td>Composición</td><td>- ID_Reseña: Número entero
- ID Pedido: Número entero
- ID Usuario: Número entero
- Valoración: Número entero (1-5)</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Confirmar modificación de reseña .</td></tr>
<tr><td>Requisito de datos</td><td colspan="3">No es necesario.</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS5.2</td></tr>
<tr><td>Descripción</td><td>Si no existe una reseña con ID_Reseña, o no existe ningún pedido con ID Pedido o ningún usuario con ID Usuario, no se puede modificar, es decir, si la valoración está en null no hay valoración. Tenemos que comparar el de lectura en la base con el de escritura, si no existe un campo de Valoración != null no podemos modificar.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF5.2</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDE5.2 y RDR5.2</td></tr>
</table>

<table>
<tr><td colspan="4">Requisito funcional : Eliminar reseña</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF5.3</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite eliminar una reseña publicada, escribiendo null en los campos de valoración y comentario</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Eliminar reseña publicada.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE5.3</td></tr>
<tr><td>Composición</td><td>- ID_Reseña: Número entero
- Valoración : null
- Comentario: null
- ID Pedido: Número entero
- ID Usuario: Número entero</td></tr>
<tr><td colspan="2" rowspan="4">Datos internos
Requisito de datos</td><td>Número de referencia</td><td>RDW5.3</td></tr>
<tr><td>Composición</td><td>Los mismos datos que RDE5.3</td></tr>
<tr><td>Número de referencia</td><td>(comprobación de existencia)
RDR5.3 Datos de Lectura</td></tr>
<tr><td>Composición</td><td>- ID_Reseña: Número entero
- ID Pedido: Número entero
- ID Usuario: Número entero
- Valoración: Número entero (1-5)</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmar la eliminación de la valoración.</td></tr>
<tr><td>Requisito de datos</td><td colspan="2">No es necesario.</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS5.3</td></tr>
<tr><td>Descripción</td><td>Tenemos que comprobar que en el campo de valoración sea != null, en caso de serlo se coloca tanto valoración como comentario en null. Si no existe una reseña con ID_Reseña, o no existe ningún pedido con ID Pedido o ningún usuario con ID Usuario, no se puede eliminar.</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF5.3</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>RDW5.3 y RDR5.3</td></tr>
</table>

Requisito funcional : Ver reseñas de un pedido			
Número de referencia	RF5.4		
Descripción del requisito	Muestra las reseñas asociadas a un pedido específico tras consultar la base de datos. Esta funcionalidad se utiliza cuando se muestran todas las reseñas de un usuario, es decir, si el agente externo quiere ver las reseñas de un pedido específico de dicha lista de reseñas.		
Entrada	**Agente externo**	Usuario	
	Acción iniciada por el agente externo	Solicitar ver (lectura) las reseñas de un producto.	
	Requisito de datos	**Número de referencia**	**RDE5.4**
		Composición	- ID Pedido: Número entero
Datos internos	**Requisito de datos**	**Número de referencia**	**RDR5.4**
		Composición	- ID Pedido: Número entero - Valoración: Número entero (1-5) - Comentario : Cadena de caracteres (500)
Salida	**Agente externo**	Usuario	
	Acción iniciada por el agente externo	Confirmar resultado	
	Requisito de datos	**Número de referencia**	**RDS5.4**
		Composición	- Los mismos datos que **RDW5.1**
Restricciones semánticas		**Número de referencia**	**RS5.4**
		Descripción	Lista de registros, cada uno de ellos con los mismos datos de **RDS5.4**
		Requisito funcional relacionado	**RF5.4**
		Requisito/s de datos relacionados	**RDS5.4**

<table>
<tr><td colspan="4">Requisito funcional : Ver reseñas de un usuario</td></tr>
<tr><td>Número de referencia</td><td colspan="3">RF5.5</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Muestra las reseñas asociadas a un usuario específico tras consultar la base de datos.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar ver (lectura) las reseñas de un producto.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE5.5</td></tr>
<tr><td>Composición</td><td>- ID Usuario: Número entero</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR 5.5</td></tr>
<tr><td>Composición</td><td>- ID Pedido: Número entero
- ID Usuario: Número entero

por cada reseña.
- Valoración : Número entero (1-5)
- Comentario: Cadena de caracteres (500)</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Confirmar resultado</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS5.5</td></tr>
<tr><td>Composición</td><td>Para cada reseña que tenga un usuario en específico, se muestra :

- Valoración: Número entero(1-5)
- Comentario: Cadena de caracteres(500)</td></tr>
<tr><td rowspan="4">Restricciones semánticas</td><td rowspan="4"></td><td>Número de referencia</td><td>RS5.5</td></tr>
<tr><td>Descripción</td><td>Lista de registros, cada uno de ellos con los mismos datos de RDS5.5</td></tr>
<tr><td>Requisito funcional relacionado</td><td>RF5.5</td></tr>
<tr><td>Requisito/s de datos relacionados</td><td>• RDS5.5</td></tr>
</table>

2.6. Subsistema de Gestión de Pagos

Autor: Laura Riera Ojea

Gestiona los métodos de pago y las transacciones asociadas.

- **RF6.1 Añadir método de pago:** Permite agregar métodos de pago introduciendo los detalles necesarios (tarjeta de crédito o PayPal), almacenando la información en la base de datos y confirmando la adición.
- **RF6.2 Eliminar método de pago:** Facilita la eliminación de métodos de pago seleccionados, borrando los datos correspondientes y confirmando la acción.
- **RF6.3 Ver métodos de pago guardados:** Muestra al usuario sus métodos de pago almacenados tras consultar la base de datos.
- **RF6.4 Realizar pago de un pedido:** Permite efectuar el pago de un pedido utilizando un método seleccionado, registrando la transacción en la base de datos y confirmando el pago o notificando un error.
- **RF6.5 Ver historial de transacciones:** Proporciona al usuario el historial de pagos realizados, mostrando las transacciones registradas en la base de datos.

<table>
<tr><th colspan="5">Requisito funcional</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF6.1: Añadir método de pago</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Permite agregar métodos de pago introduciendo los detalles necesarios (tarjeta de crédito o PayPal).</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Solicitar agregar un nuevo método de pago.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td colspan="2">RDE6.1</td></tr>
<tr><td>Composición</td><td colspan="2">● Tipo de método de pago: Cadena de caracteres ("Tarjeta de crédito", "PayPal").
● Detalles del método de pago:
➔ Si es tarjeta de crédito:
❖ Número de tarjeta: Cadena numérica (16 dígitos).
❖ Fecha de expiración: Fecha (MM/AA).
❖ Código CVV: Cadena numérica (3 o 4 dígitos)
➔ Si es PayPal:
❖ Correo electrónico de PayPal: Cadena de caracteres (100).</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td colspan="2">RDW6.1</td></tr>
<tr><td>Composición</td><td colspan="2">● ID del método de pago: Entero.
● ID de usuario: Entero.
● Tipo de método de pago: Cadena de caracteres.
● Número de tarjeta (si aplica).
● Fecha de expiración (si aplica).
● Nombre del titular (si aplica).
● Correo electrónico de PayPal (si aplica).
● Fecha de registro: Fecha y hora.</td></tr>
<tr><td rowspan="2">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente</td><td colspan="3">Recibir confirmación del método de pago.</td></tr>
</table>

	externo		
	Requisito de datos	**Número de referencia**	**RDS6.1**
		Composición	● **ID del método de pago:** Entero.
Restricciones semánticas		**Número de referencia**	**RS6.1**
		Descripción	**Validación del método de pago:** Si los detalles proporcionados del método de pago no son válidos o la tarjeta es rechazada durante la verificación, no se agrega el método y se notifica al usuario con un mensaje de error.
		Requisito funcional relacionado	**RF6.1**
		Requisito/s de datos relacionados	● **RDW6.1**

<table>
<tr><th colspan="4">Requisito funcional</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF6.2: Eliminar método de pago.</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Se borran los datos correspondientes al método de pago y confirma la acción.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar eliminación de un método de pago.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE6.2</td></tr>
<tr><td>Composición</td><td>● ID del método de pago: Entero
● Usado: Booleno</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Composición</td><td>RDR6.2</td></tr>
<tr><td>Número de referencia</td><td>● Comprobar si se ha usado método de pago: Se comprueba si usado es true.</td></tr>
<tr><td>Composición</td><td>RDW6.2</td></tr>
<tr><td>Número de referencia</td><td>● Eliminación del método de pago: Se elimina el registro correspondiente al ID.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación de eliminación.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS6.2</td></tr>
<tr><td>Composición</td><td>● Mensaje de confirmación: Cadena de caracteres ("El método de pago ha sido eliminado exitosamente.").
● Mensaje de error: Cadena de caracteres (“El método no se ha podido eliminar”)</td></tr>
<tr><td rowspan="2">Restricciones semánticas</td><td rowspan="2"></td><td>Número de referencia</td><td>RS6.2</td></tr>
<tr><td>Descripción</td><td>Verificación de propiedad del método de pago: Solo el usuario propietario puede eliminar sus métodos de pago. Si el ID proporcionado no</td></tr>
</table>

			corresponde a un método de pago del usuario, se deniega la acción y se informa al usuario. **Método usado:** Si el método de pago ya se había usado antes, no se podrá borrar el método de pago y saldrá el mensaje de error.
		Requisito funcional relacionado	**RF6.2**
		Requisito/s de datos relacionados	• **RDW6.2** • **RDR6.2**

<table>
<tr><th colspan="5">Requisito funcional</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF6.3: Ver métodos de pago guardados.</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Muestra al usuario sus métodos de pago almacenados</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Solicitar visualización de métodos de pago guardados.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td colspan="2">RDE6.3</td></tr>
<tr><td>Composición</td><td colspan="2">● NA</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td colspan="2">RDR6.3</td></tr>
<tr><td>Composición</td><td colspan="2">● ID del método de pago: Entero.
● Tipo de método de pago: Cadena de caracteres.
● Terminación de la tarjeta: (4 dígitos)(si aplica)
● Fecha de registro: Fecha y hora.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Visualizar los métodos de pago guardados.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td colspan="2">RDS6.3</td></tr>
<tr><td>Composición</td><td colspan="2">● Lista de métodos de pago obtenida de RDR6.3</td></tr>
<tr><td rowspan="2">Restricciones semánticas</td><td rowspan="2"></td><td>Número de referencia</td><td colspan="2">RS6.3</td></tr>
<tr><td>Descripción</td><td colspan="2">Privacidad de datos. Solo el usuario autenticado puede ver sus propios métodos de pago. Si se detecta un intento de acceder a métodos de pago de otros usuarios, se deniega la acción y se registra el evento.</td></tr>
</table>

		Requisito funcional relacionado	**RF6.3**
		Requisito/s de datos relacionados	• **RDR6.3** • **RDS6.3**

<table>
<tr><th colspan="4">Requisito funcional</th></tr>
<tr><td>Número de referencia</td><td colspan="3">RF6.4: Realizar pago de un pedido.</td></tr>
<tr><td>Descripción del requisito</td><td colspan="3">Permite efectuar el pago de un pedido utilizando un método seleccionado, registrando la transacción en la base de datos y confirmando el pago o notificando un error.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Solicitar realizar el pago de un pedido.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE6.4</td></tr>
<tr><td>Composición</td><td>● ID del pedido: Entero.
● ID del método de pago: Entero.
● Cantidad a pagar: Decimal.</td></tr>
<tr><td rowspan="4">Datos internos</td><td rowspan="4">Requisito de datos</td><td>Número de referencia</td><td>RDR6.4</td></tr>
<tr><td>Composición</td><td>● Detalles del pedido:
➔ Estado del pedido: Cadena de caracteres.
➔ Cantidad total a pagar: Decimal.
● Detalles del método de pago:
➔ ID del método de pago: Entero.</td></tr>
<tr><td>Número de referencia</td><td>RDW6.4</td></tr>
<tr><td>Composición</td><td>● ID de transacción: Entero.
● ID del pedido: Entero.
● ID del usuario: Entero.
● ID del método de pago: Entero.
● Cantidad a pagar: Decimal.
● Fecha y hora de la transacción.</td></tr>
<tr><td rowspan="3">Salida</td><td>Agente externo</td><td colspan="2">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="2">Recibir confirmación del pago o notificación de error.</td></tr>
<tr><td>Requisito de datos</td><td>Número de referencia</td><td>RDS6.4</td></tr>
</table>

		Composición	● **Mensaje de confirmación o error:** Cadena de caracteres. ● **Detalles de la transacción (en caso de éxito):** ➔ **ID de la transacción:** Entero. ➔ **Cantidad pagada:** Decimal. ➔ **Fecha y hora.**
Restricciones semánticas		**Número de referencia**	**RS6.4**
		Descripción	● **Validación de cantidad y estado del pedido**: El sistema debe verificar que el pedido está pendiente de pago y que la cantidad a pagar coincide con la cantidad total del pedido. ● **Autorización del método de pago**: Solo se puede utilizar un método de pago que pertenezca al usuario autenticado.
		Requisito funcional relacionado	**RF6.4**
		Requisito/s de datos relacionados	● **RDE6.4** ● **RDR6.4** ● **RDW6.4**

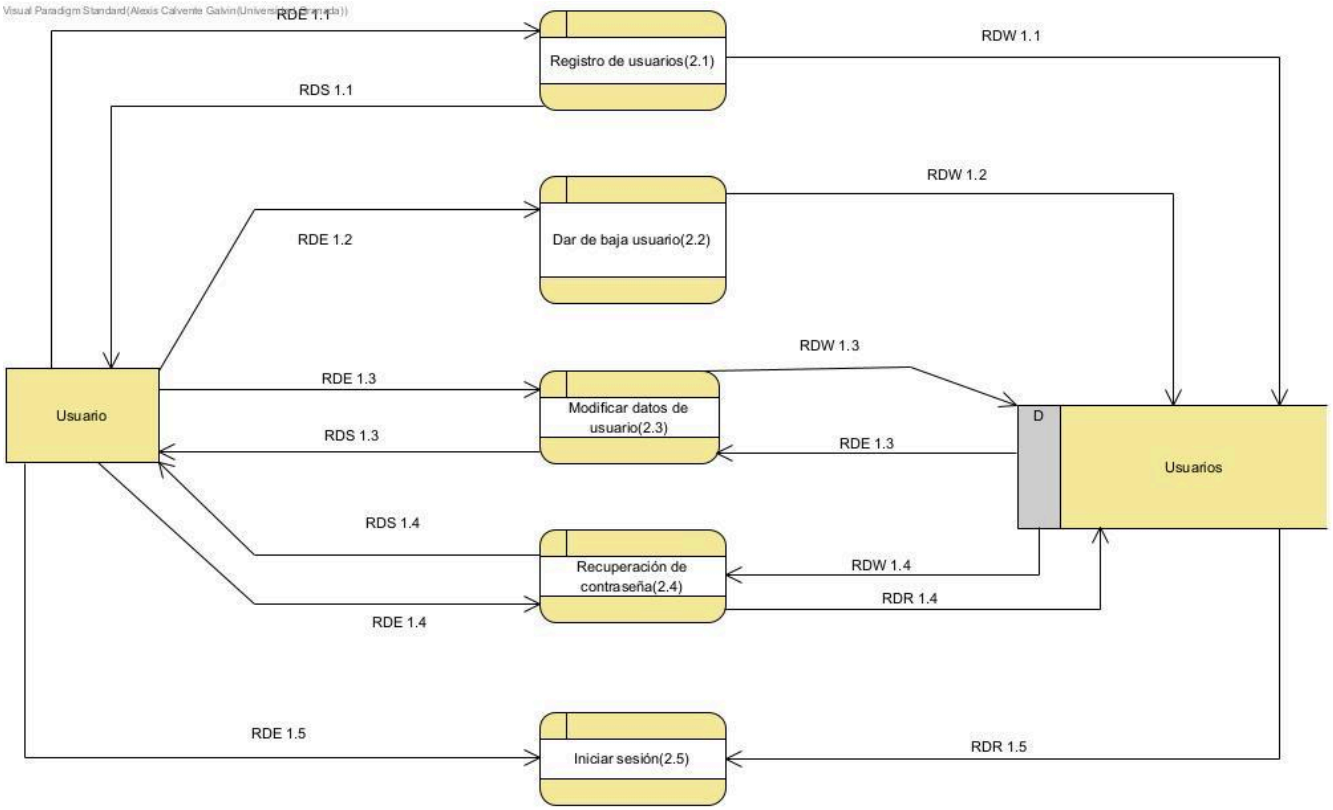
<table>
<tr><th colspan="5">Requisito funcional</th></tr>
<tr><td>Número de referencia</td><td colspan="4">RF6.5: Ver historial de transacciones</td></tr>
<tr><td>Descripción del requisito</td><td colspan="4">Proporciona al usuario el historial de pagos realizados, mostrando las transacciones registradas en la base de datos.</td></tr>
<tr><td rowspan="4">Entrada</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Solicitar visualización del historial de transacciones.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDE6.5</td></tr>
<tr><td>Composición</td><td>● NA</td></tr>
<tr><td rowspan="2">Datos internos</td><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDR6.5</td></tr>
<tr><td>Composición</td><td>● ID de transacción: Entero.
● ID del pedido: Entero.
● Cantidad pagada: Decimal.
● Fecha y hora de la transacción: Fecha y hora.
● Estado de la transacción: Cadena de caracteres
● Método de pago utilizado: Cadena de caracteres.</td></tr>
<tr><td rowspan="4">Salida</td><td>Agente externo</td><td colspan="3">Usuario</td></tr>
<tr><td>Acción iniciada por el agente externo</td><td colspan="3">Visualizar historial de transacciones.</td></tr>
<tr><td rowspan="2">Requisito de datos</td><td>Número de referencia</td><td>RDS6.5</td></tr>
<tr><td>Composición</td><td>● Lista de transacciones obtenida de RDR6.5</td></tr>
<tr><td rowspan="2">Restricciones semánticas</td><td rowspan="2"></td><td>Número de referencia</td><td>RS6.5</td></tr>
<tr><td>Descripción</td><td>● Privacidad y seguridad: Solo el usuario</td></tr>
</table>

			autenticado puede acceder a su propio historial de transacciones. Si se detecta un intento de acceso no autorizado, se deniega la acción y se registra el evento de seguridad.
		Requisito funcional relacionado	**RF6.5**
		Requisito/s de datos relacionados	• **RDS6.5** • **RDR6.5**

DFD1s

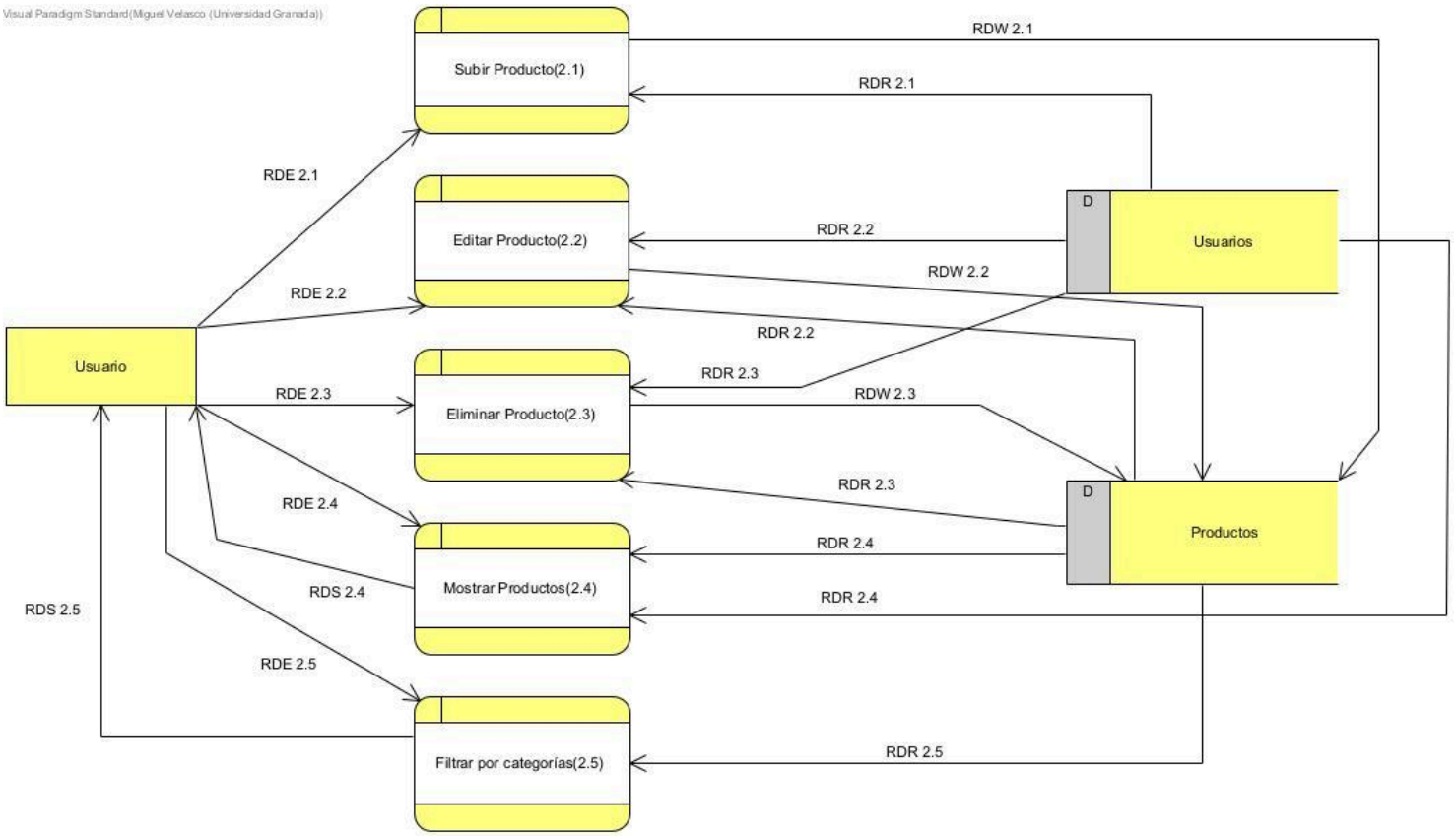
1.Subsistema de usuarios

Autor: Alexis Calvente Galvín



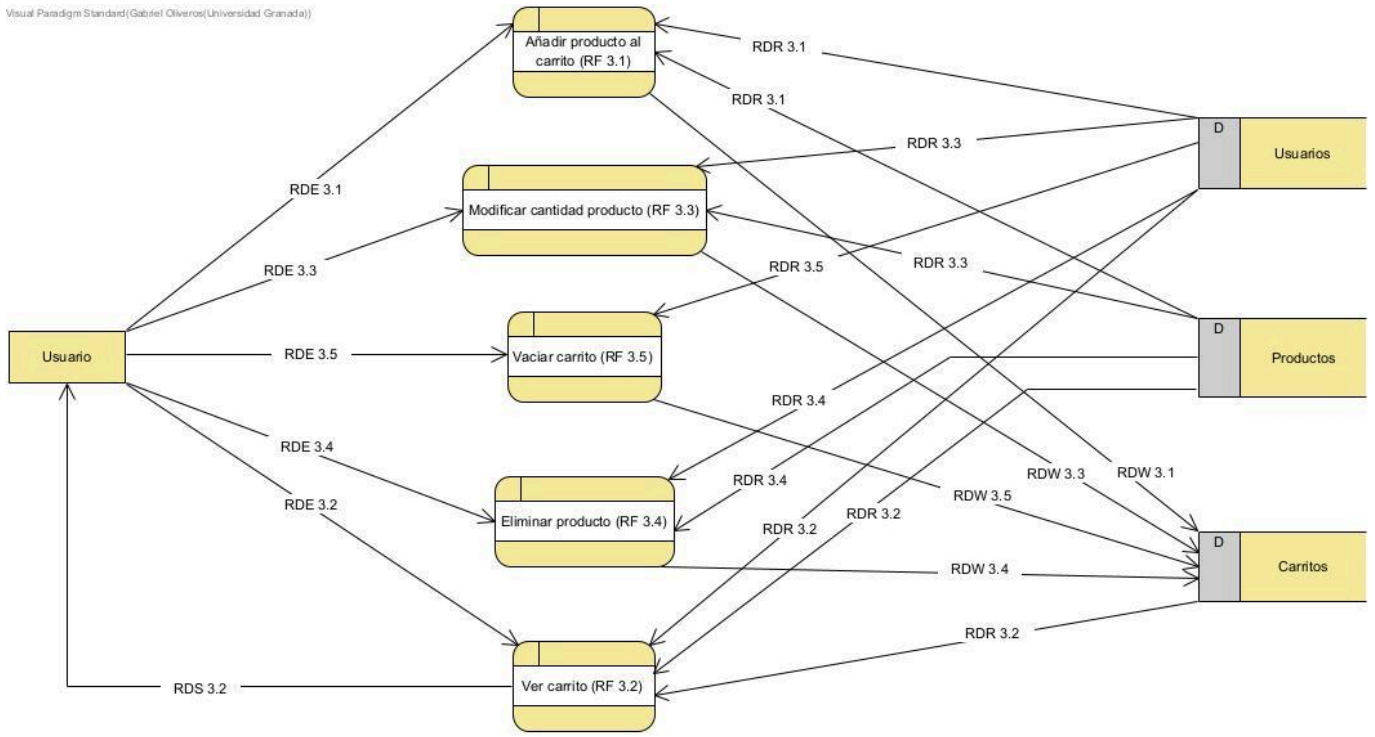
2. Subsistema de Productos

Autor: Miguel Velasco Fernández



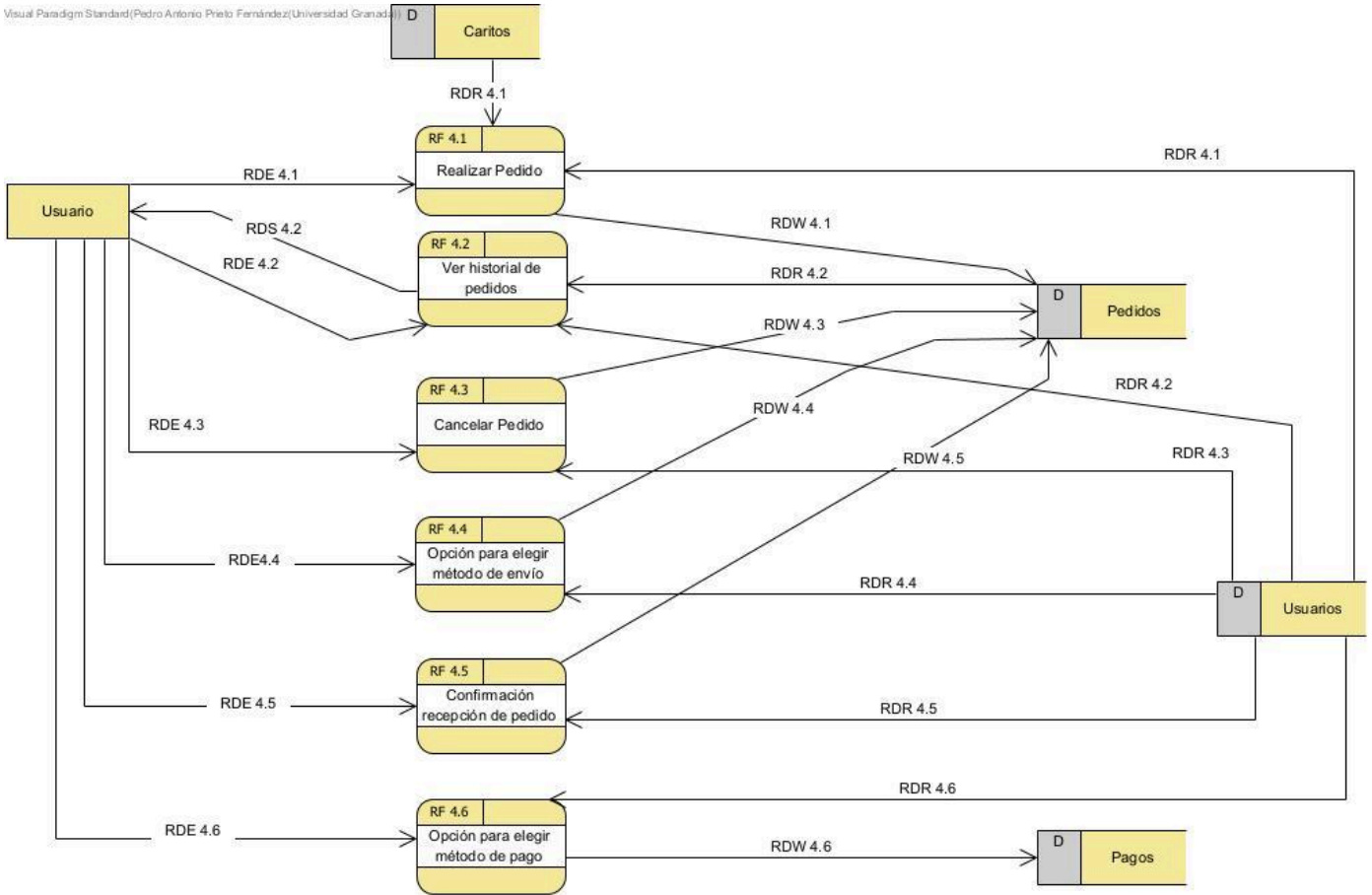
3. Subsistema de Gestión de Carrito de Compras

Autor: Gabriel Oliveros Jiménez



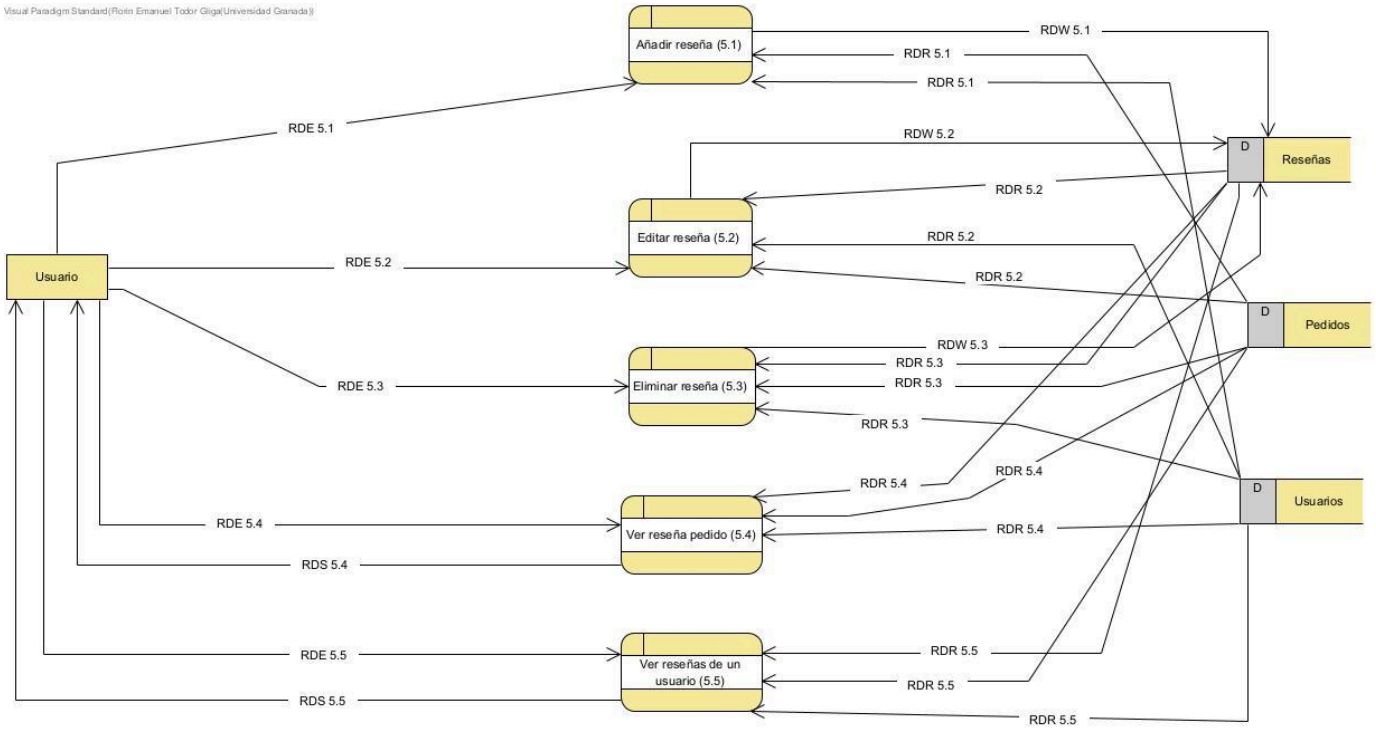
4. Subsistema de Gestión de pedidos

Autor: Pedro Antonio Prieto Fernández



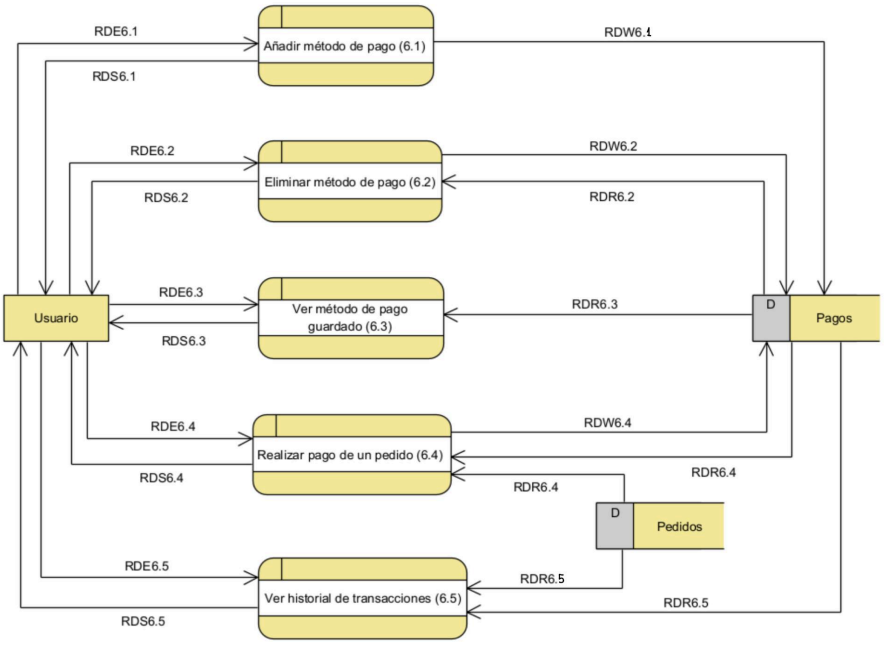
5. Subsistema de Reseñas

Autor: Florín Emanuel Todor Gliga

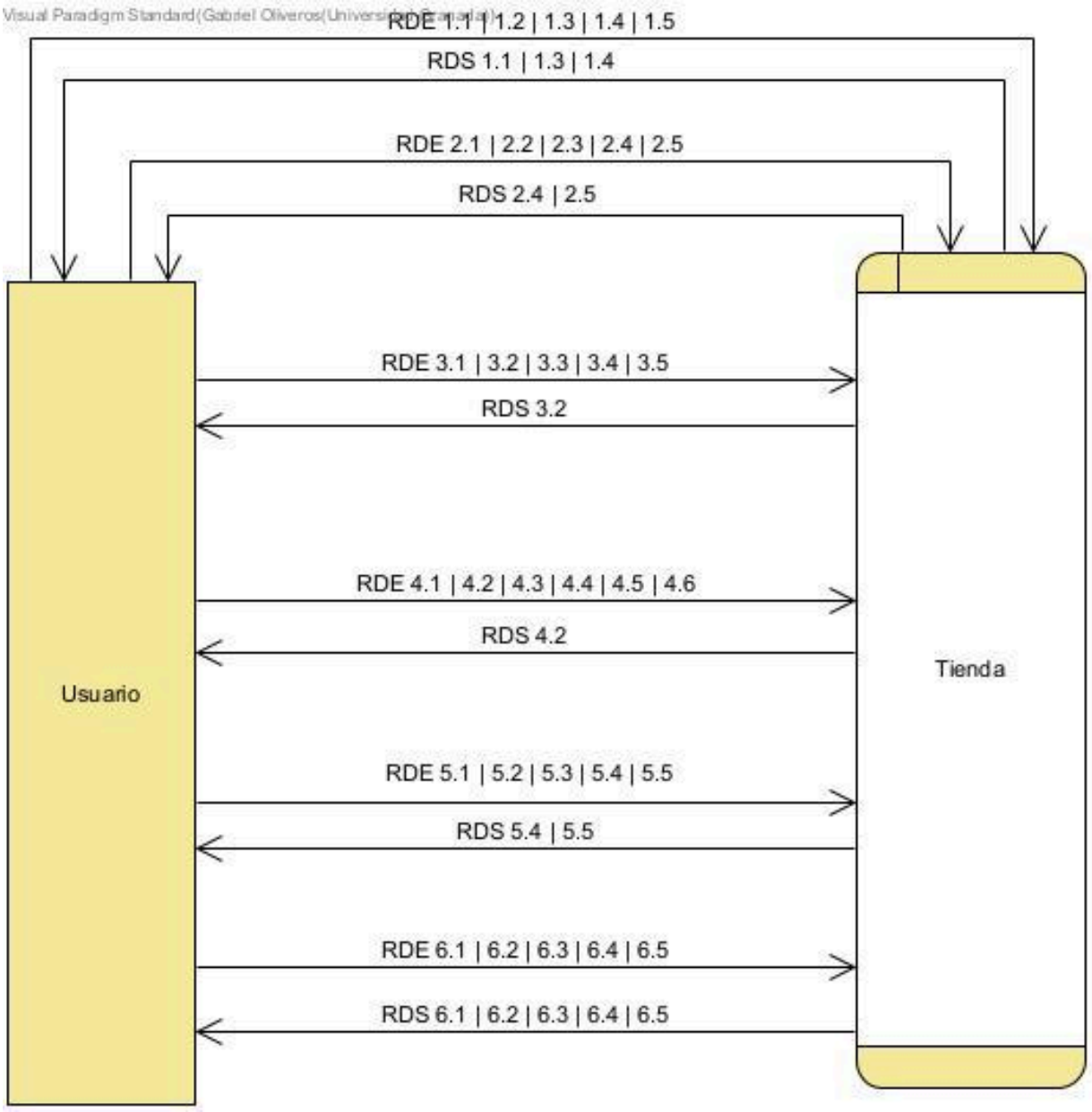


6. Subsistema de Pago

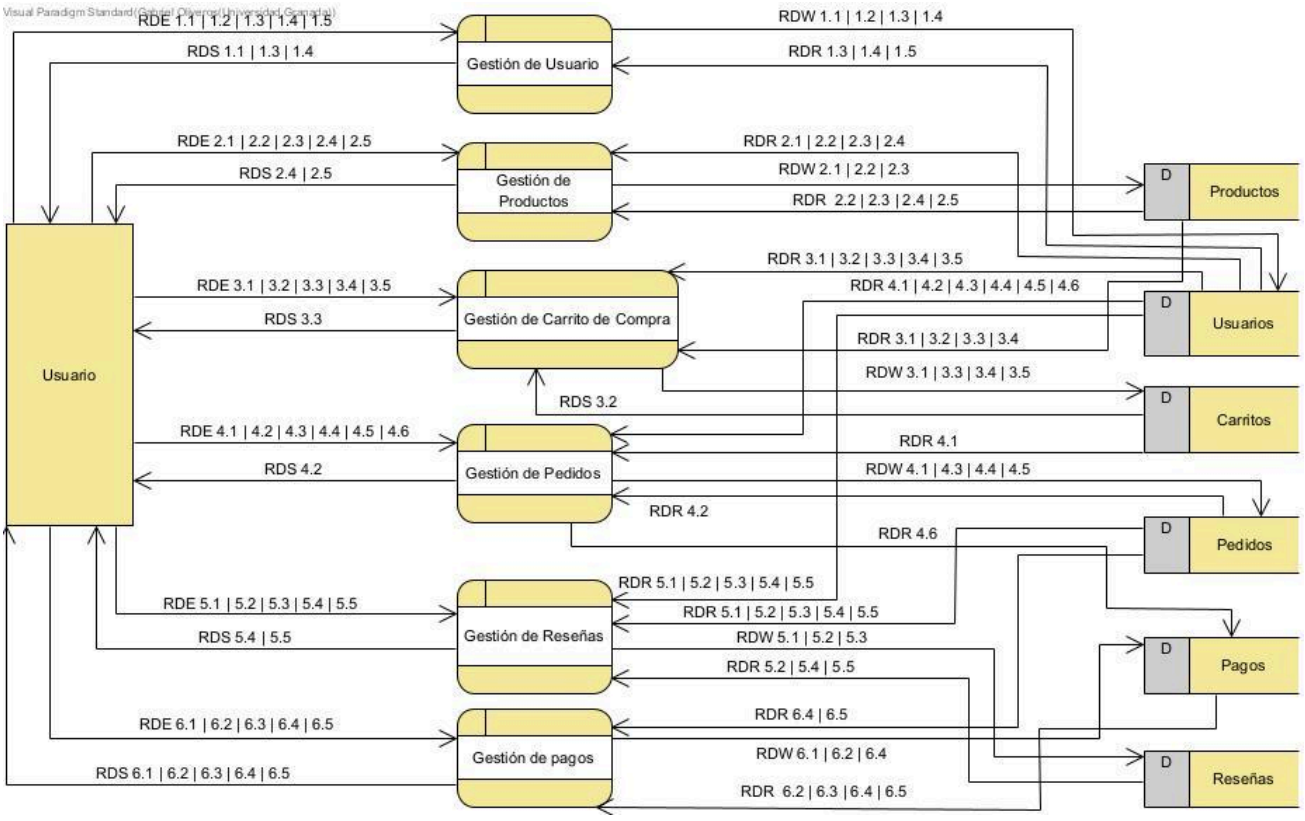
Autor: Laura Riera Ojea



Caja Negra

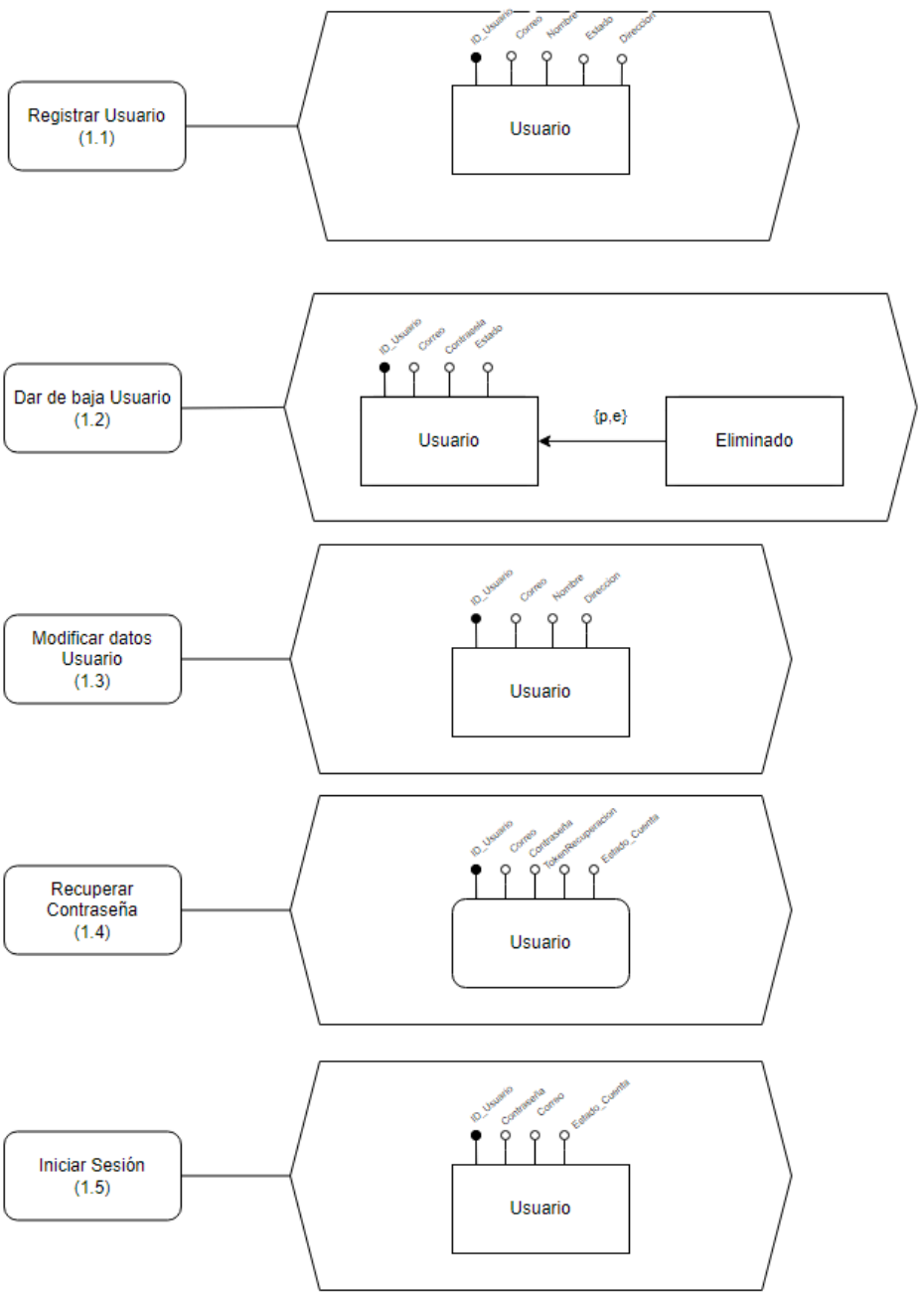

RDE 1.1 | 1.2 | 1.3 | 1.4 | 1.5
RDS 1.1 | 1.3 | 1.4
RDE 2.1 | 2.2 | 2.3 | 2.4 | 2.5
RDS 2.4 | 2.5
RDE 3.1 | 3.2 | 3.3 | 3.4 | 3.5
RDS 3.2
RDE 4.1 | 4.2 | 4.3 | 4.4 | 4.5 | 4.6
RDS 4.2
RDE 5.1 | 5.2 | 5.3 | 5.4 | 5.5
RDS 5.4 | 5.5
RDE 6.1 | 6.2 | 6.3 | 6.4 | 6.5
RDS 6.1 | 6.2 | 6.3 | 6.4 | 6.5
Usuario
Tienda

DFD0 Sistema

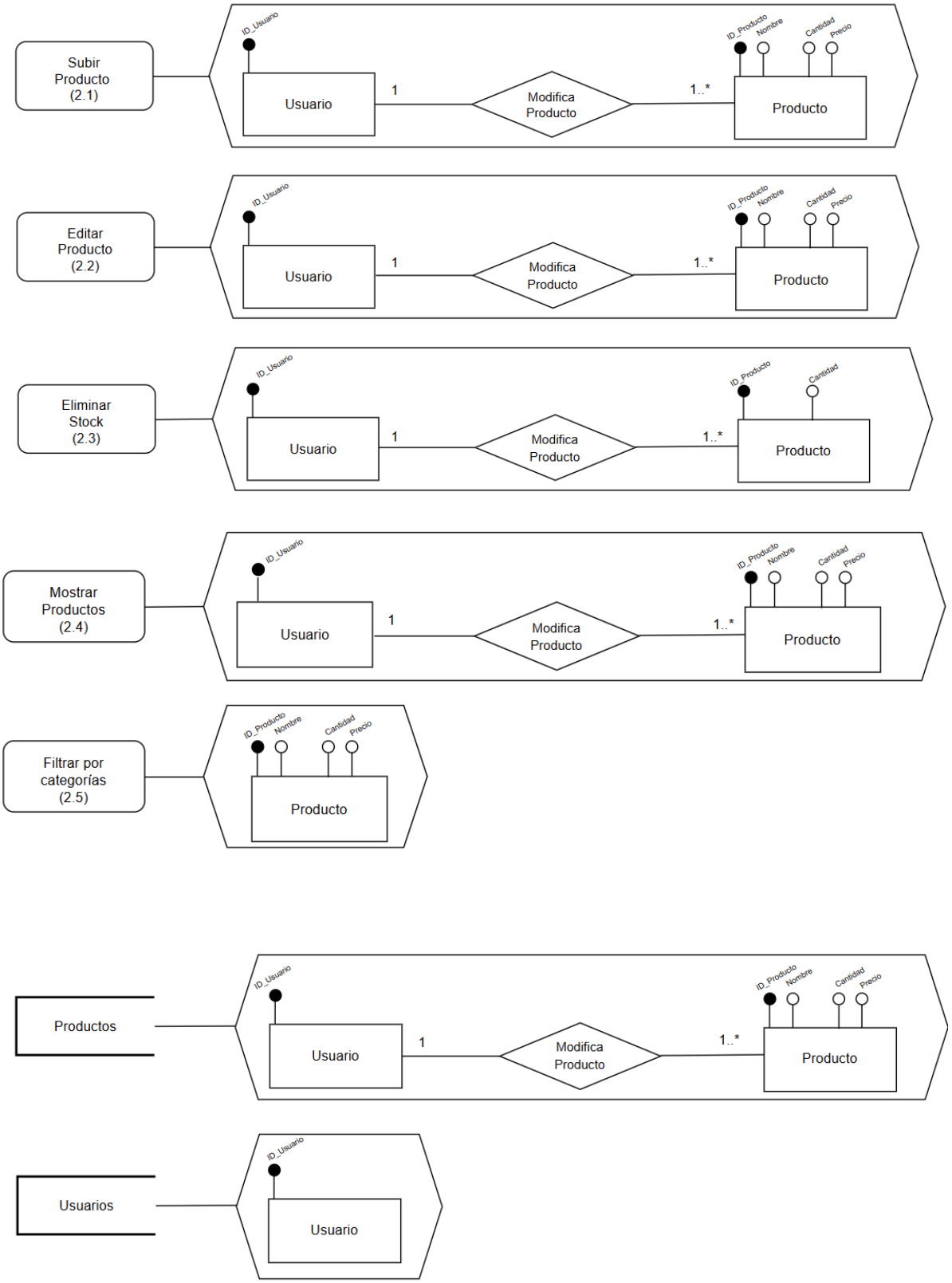


ESQUEMAS EXTERNOS

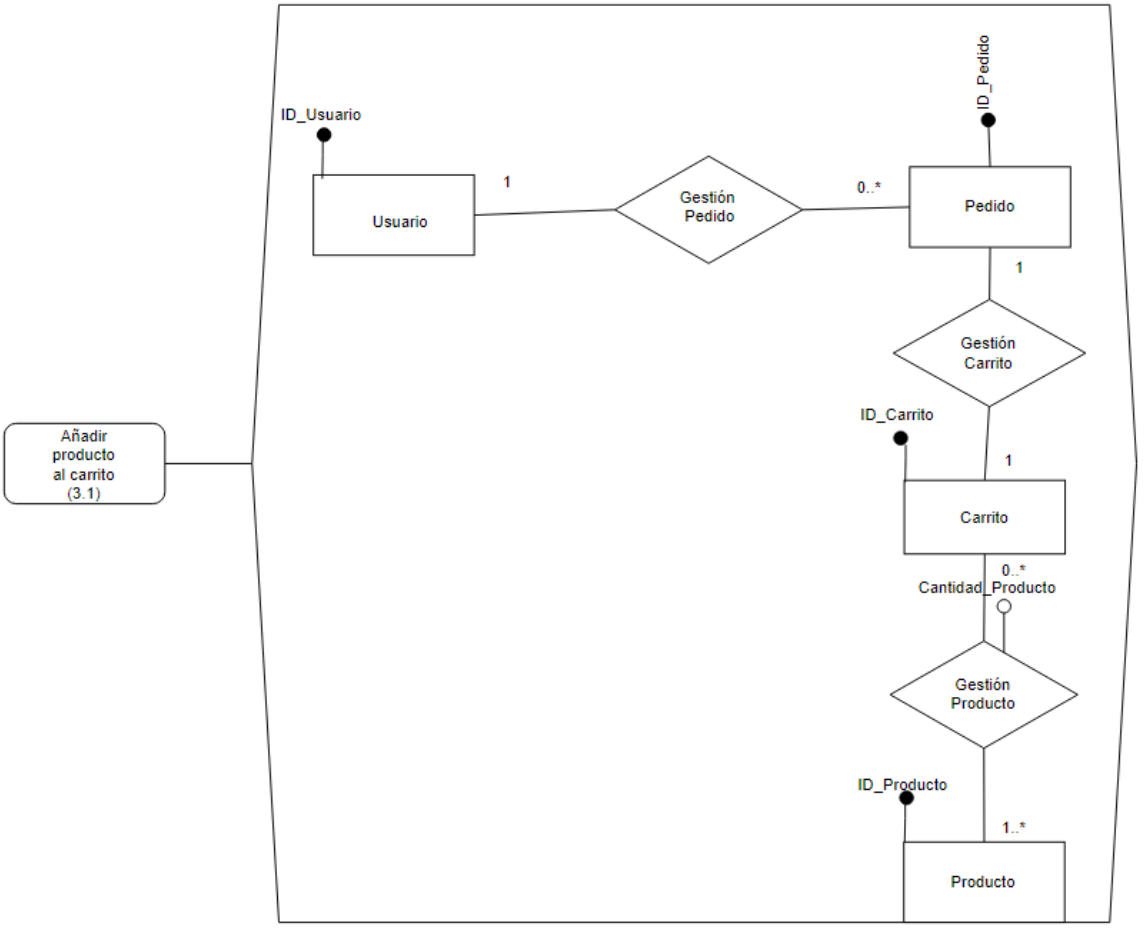
Subsistema Gestión Usuario (Alexis Calvente Galvin)

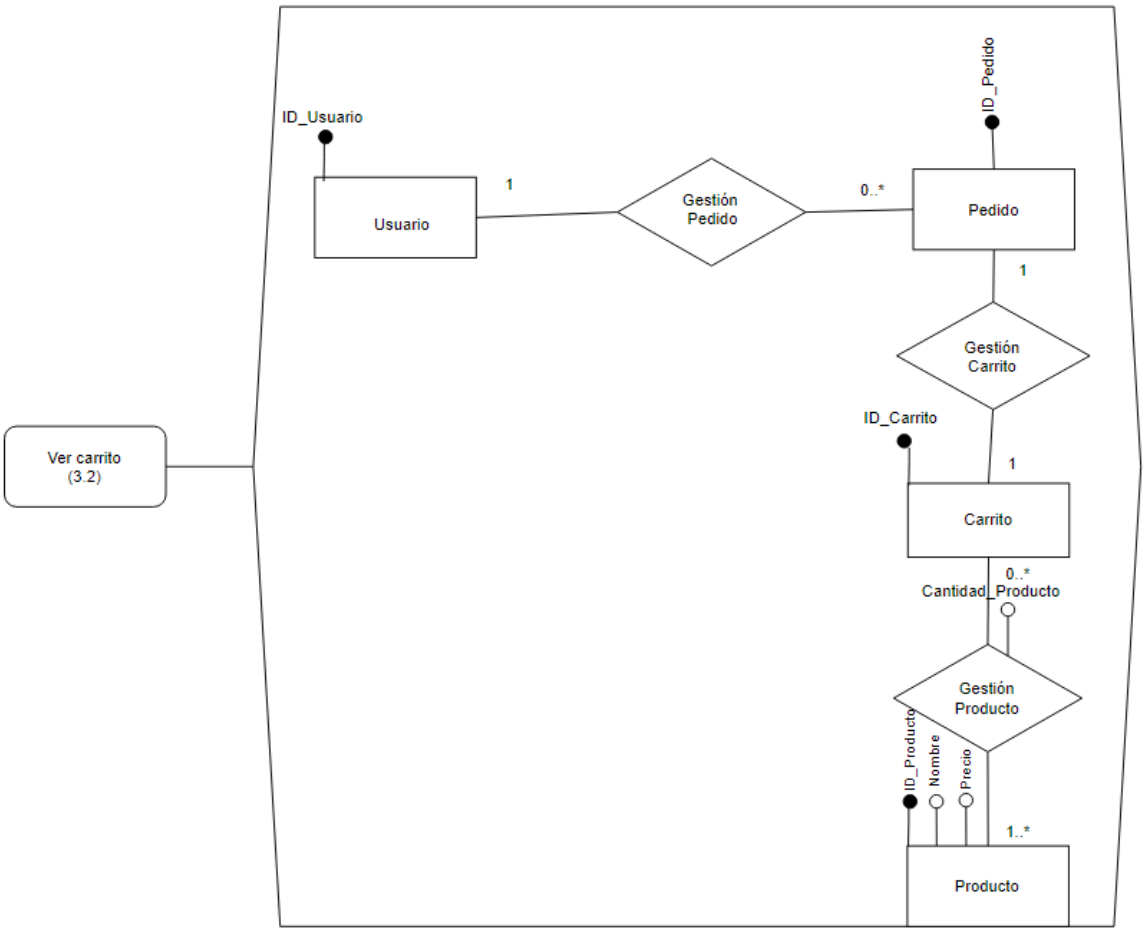


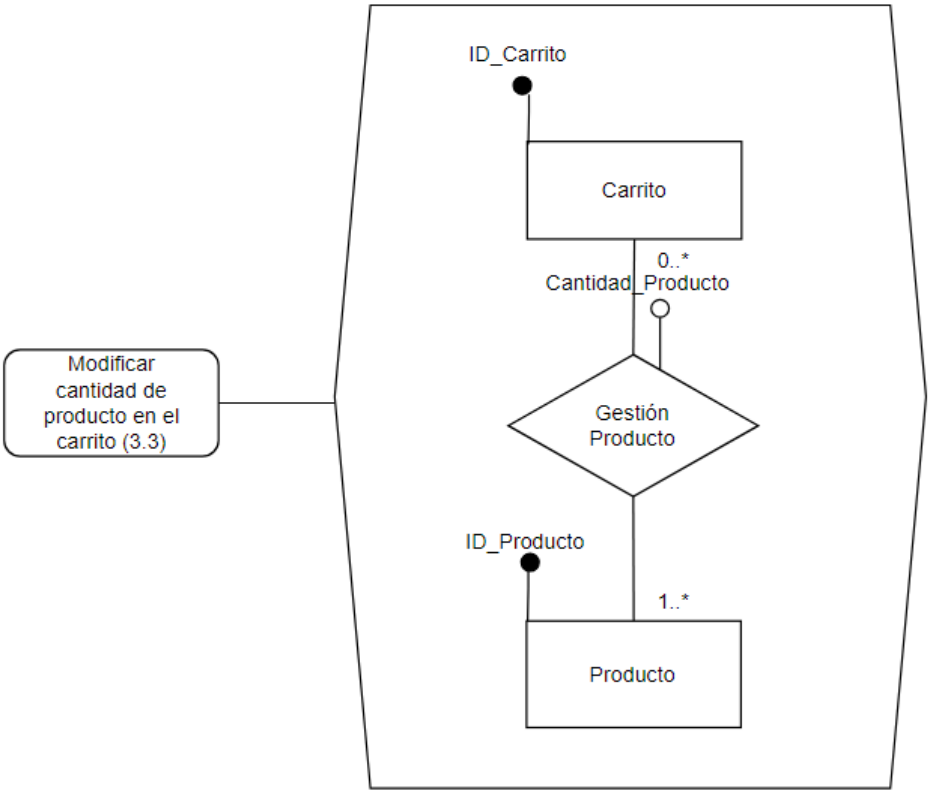
Subsistema Gestión Producto (Miguel Velasco Fernández)

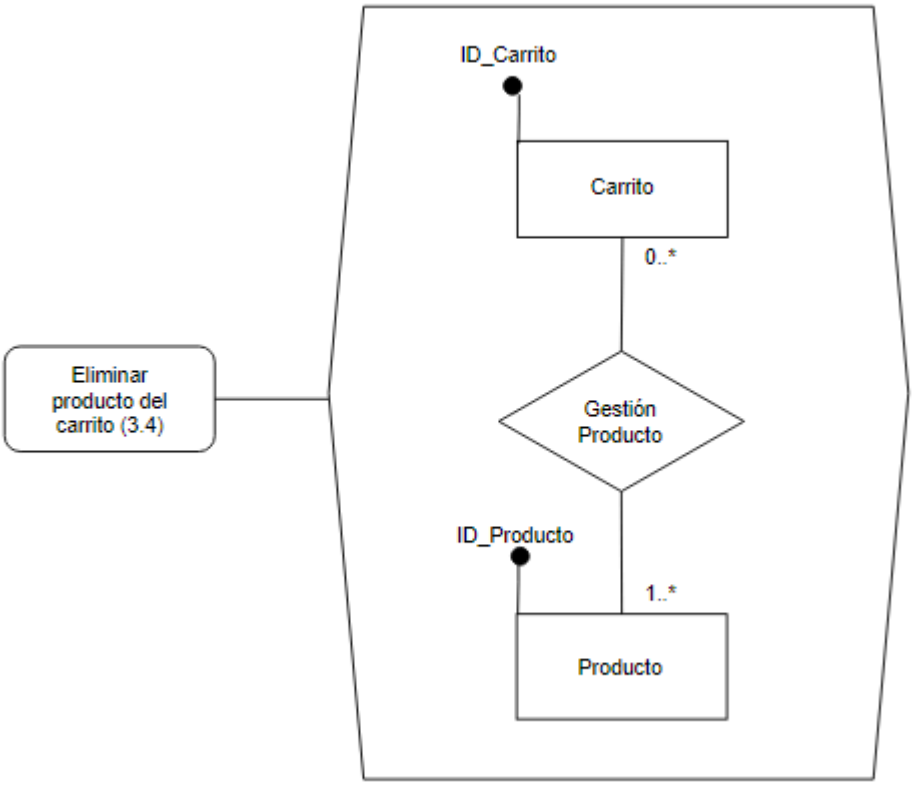


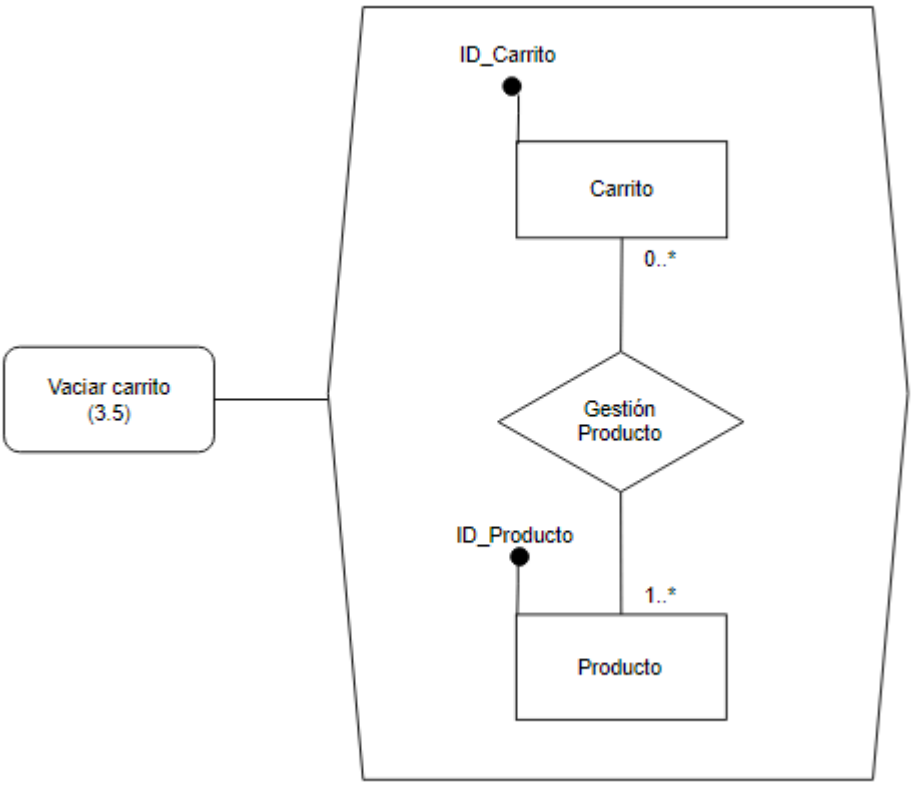
Subsistema Gestión Carrito (Gabriel Oliveros Jiménez)

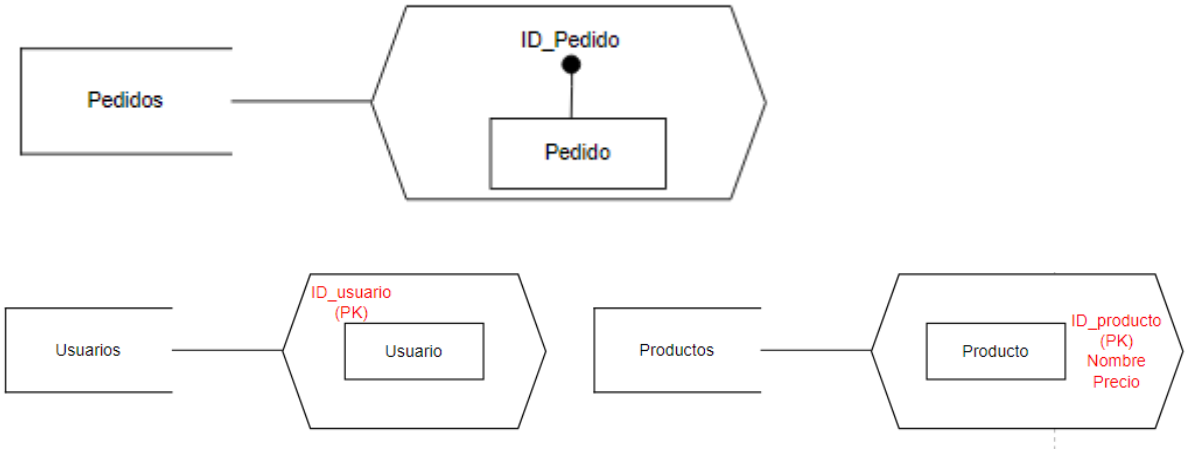



ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Pedido
1
Gestión Carrito
ID_Carrito
1
Carrito
0..*
Cantidad_Producto
Gestión Producto
ID_Producto
Nombre
Precio
1..*
Producto
Ver carrito (3.2)

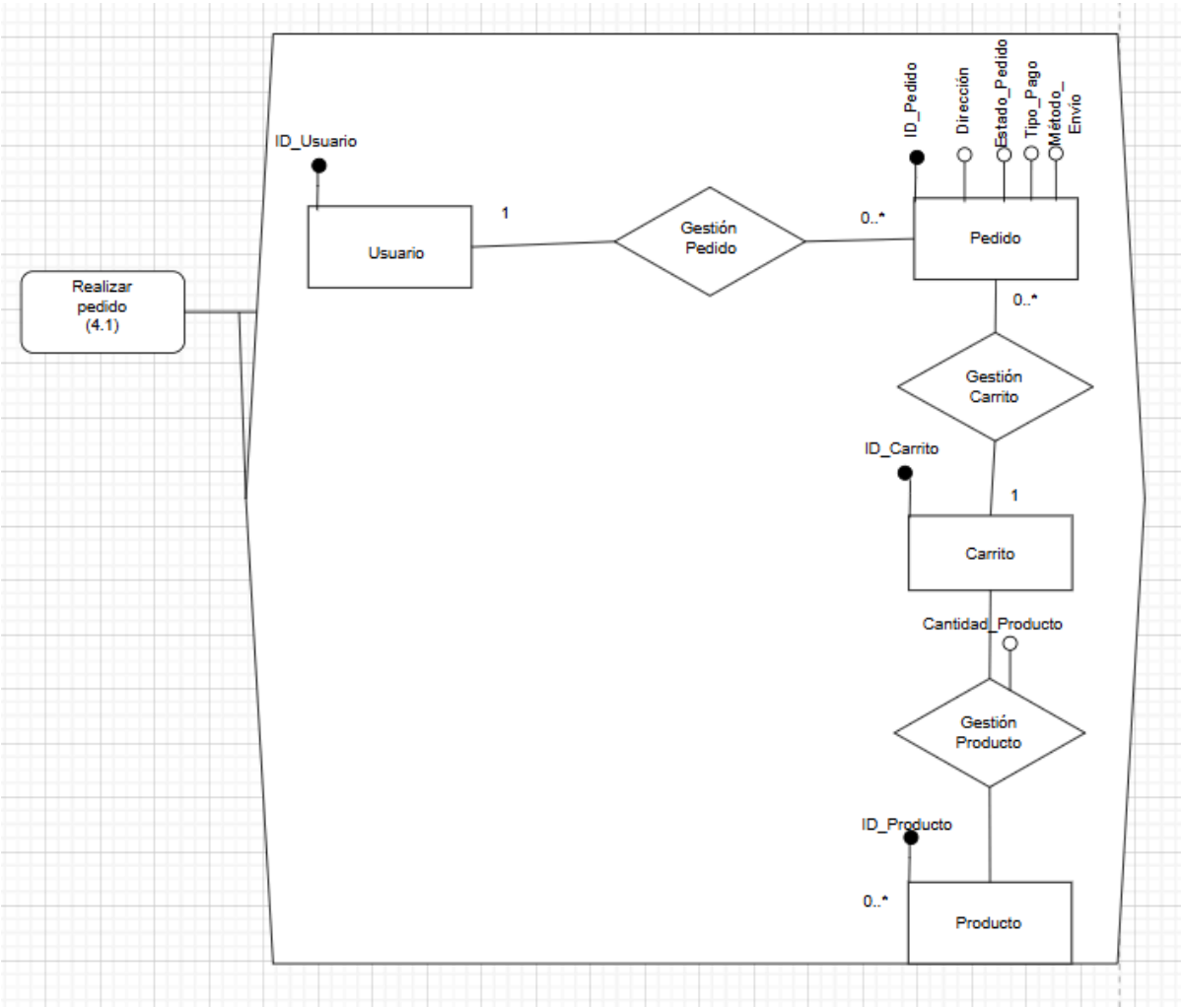

ID_Carrito
Carrito
0..*
Cantidad_Producto
Gestión Producto
ID_Producto
1..*
Producto
Modificar cantidad de producto en el carrito (3.3)

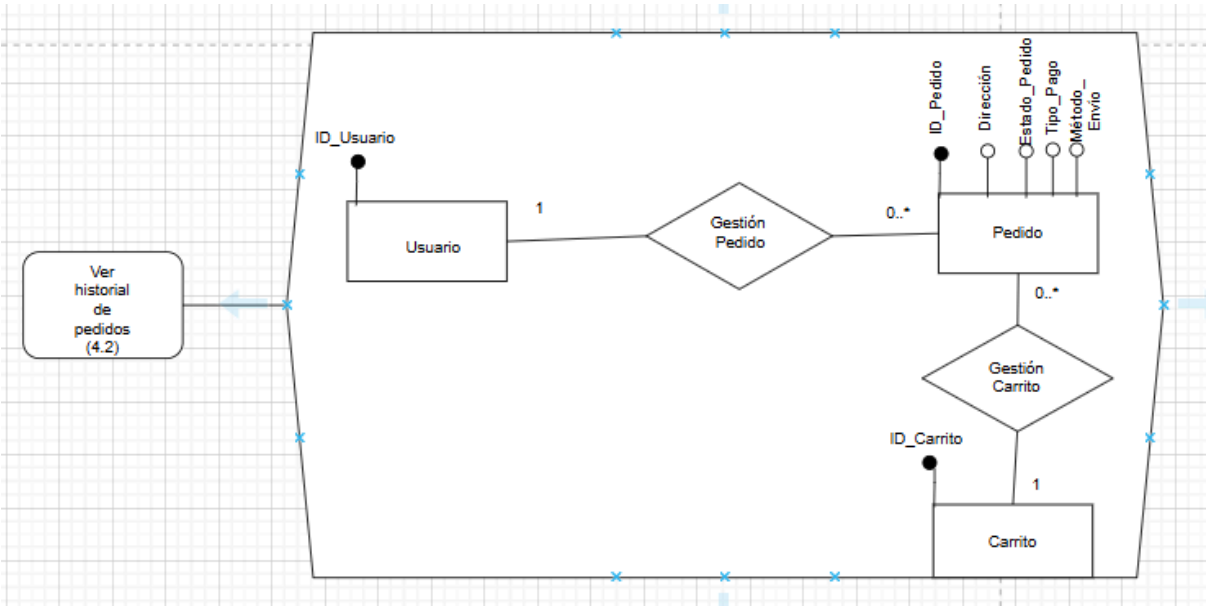

ID_Carrito
Carrito
0..*
Eliminar producto del carrito (3.4)
Gestión Producto
ID_Producto
1..*
Producto

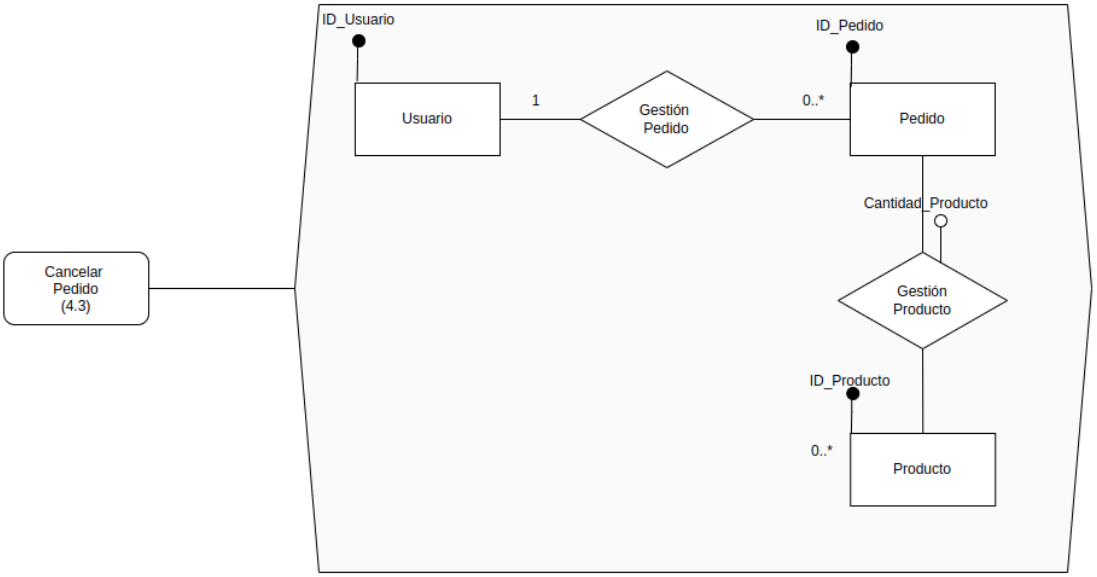

ID_Carrito
Carrito
0..*
Vaciar carrito (3.5)
Gestión Producto
ID_Producto
1..*
Producto

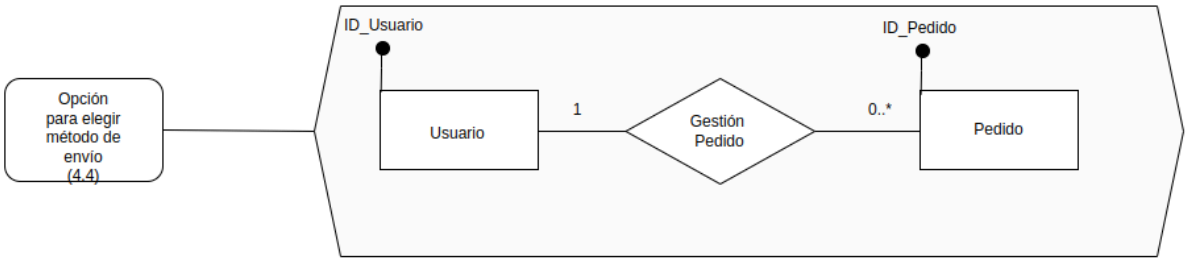


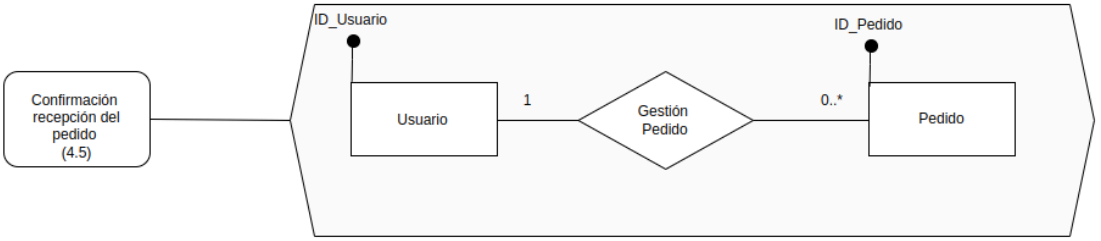
Subsistema Gestión Pedido (Pedro Antonio Prieto Fernández)

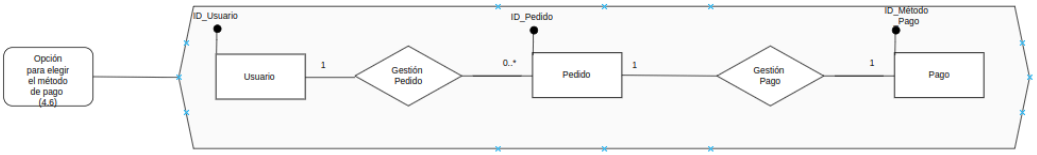



ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Dirección
Estado_Pedido
Tipo_Pago
Método_ Envío
Pedido
0..*
Gestión Carrito
ID_Carrito
1
Carrito
Ver historial de pedidos (4.2)

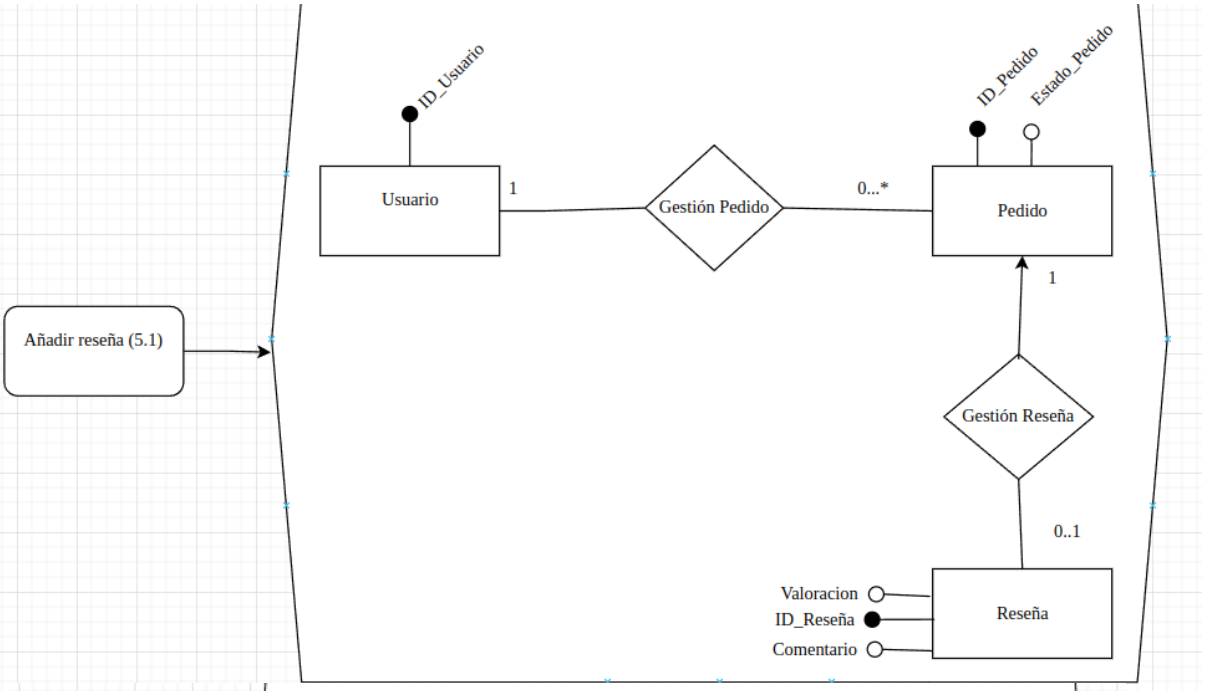

ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Pedido
Cantidad_Producto
Gestión Producto
ID_Producto
0..*
Producto
Cancelar Pedido (4.3)

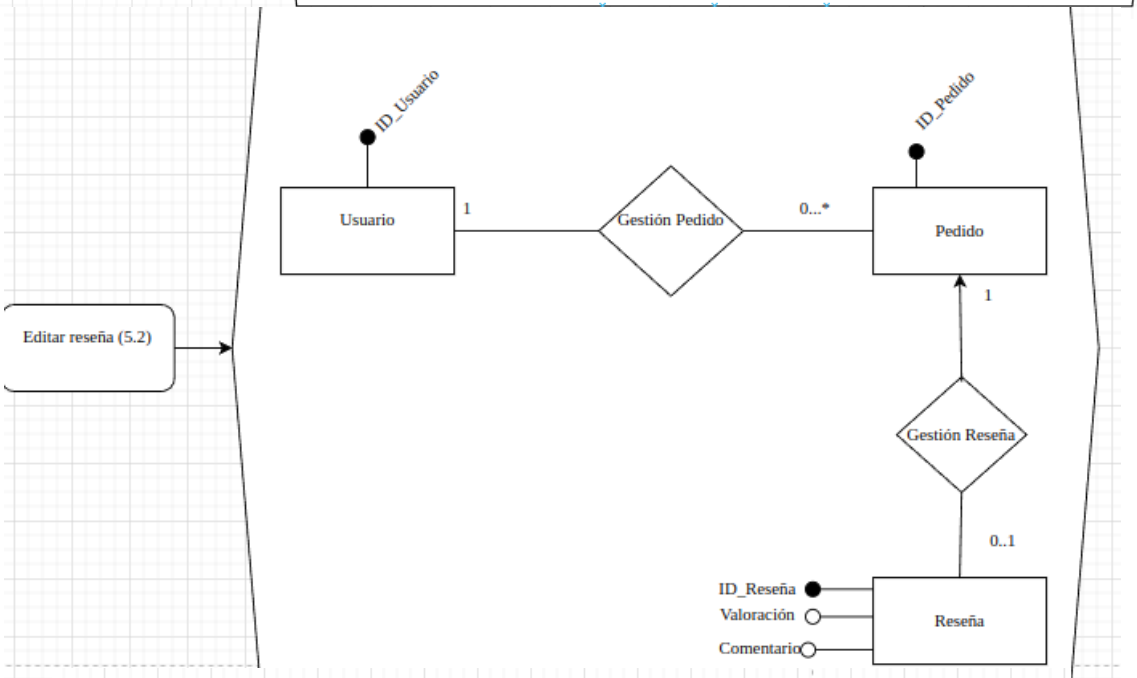

ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Pedido
Opción para elegir método de envío (4.4)

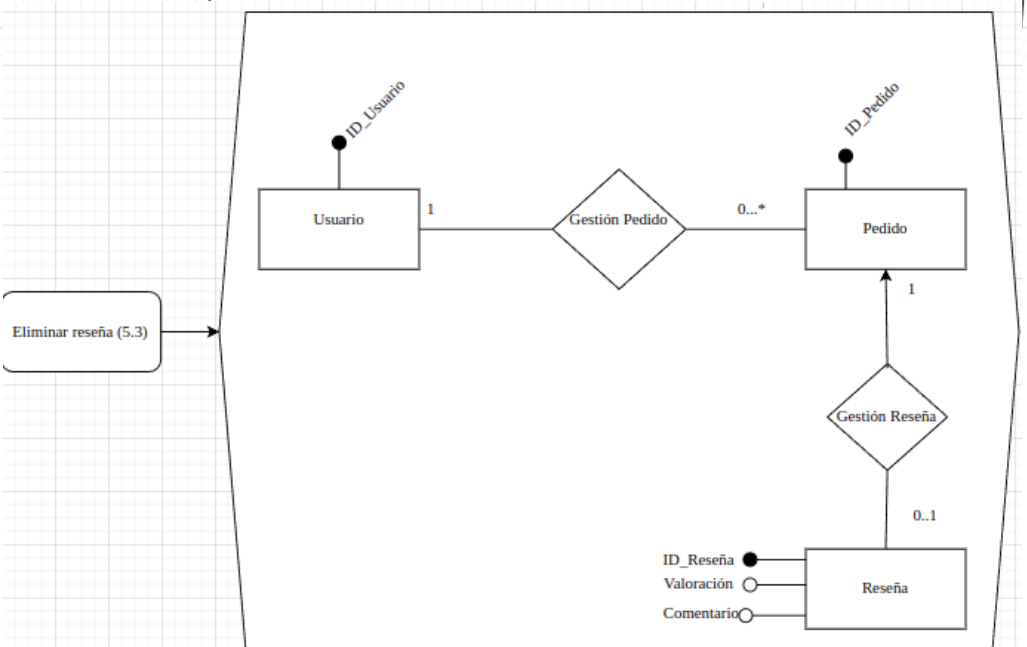

Confirmación recepción del pedido (4.5)
ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Pedido

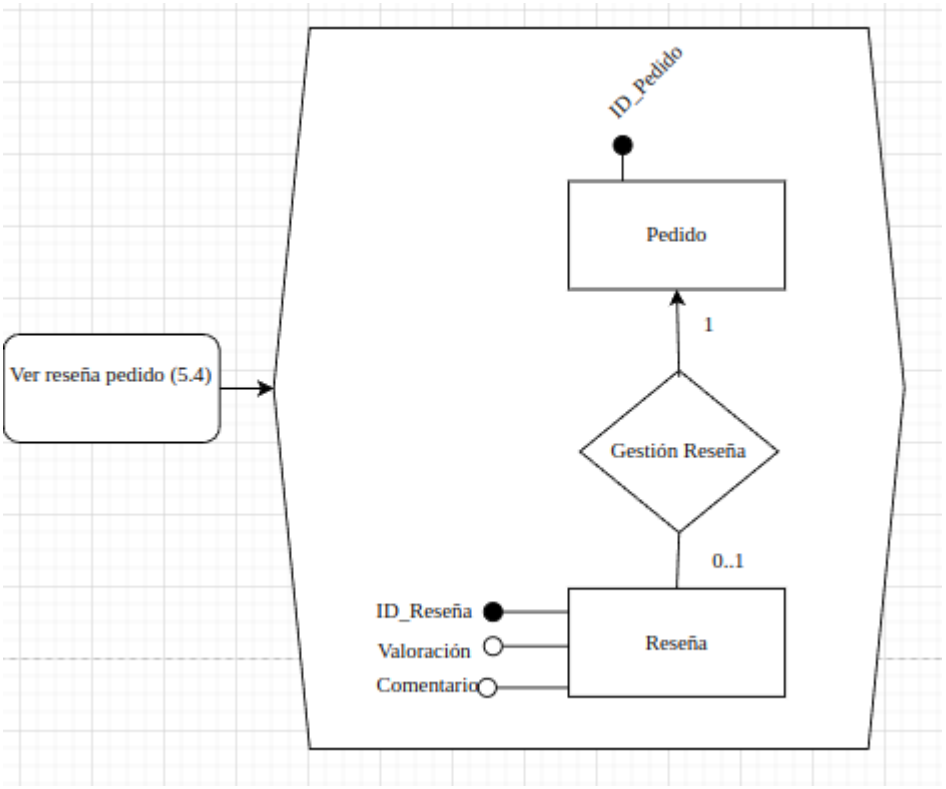

Opción para elegir el método de pago (4.6)
ID_Usuario
Usuario
1
Gestión Pedido
0..*
ID_Pedido
Pedido
1
Gestión Pago
1
ID_Método Pago
Pago

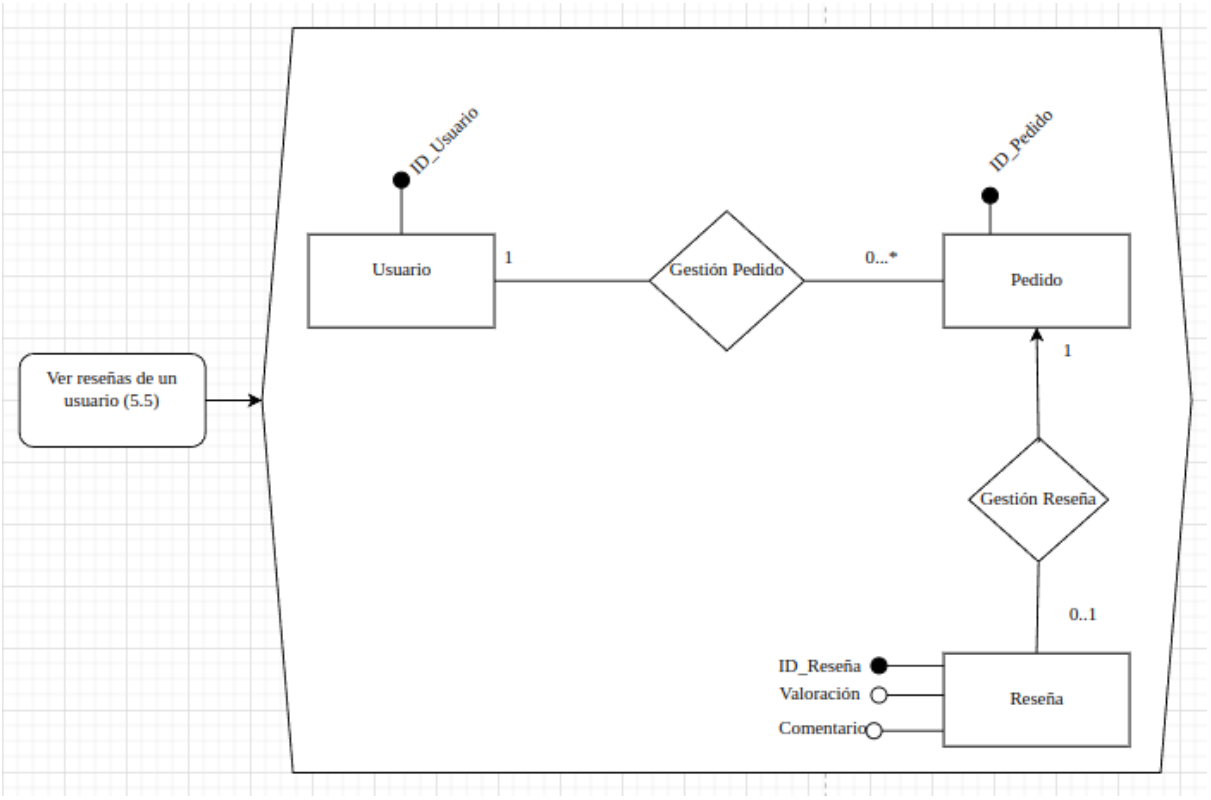
Subsistema Gestión Reseñas (Florín Emanuel Todor Gliga)

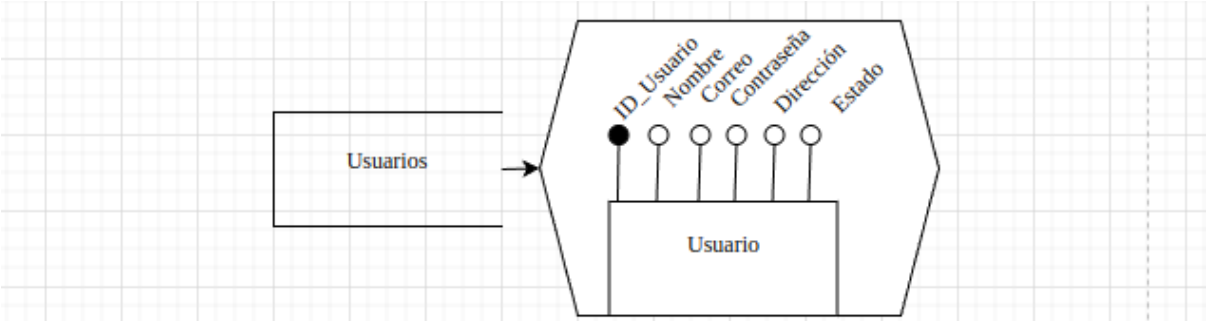


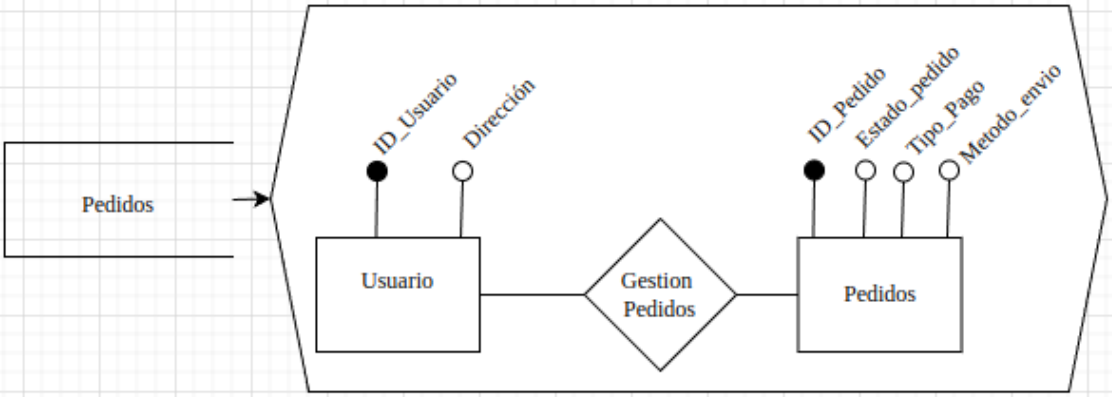


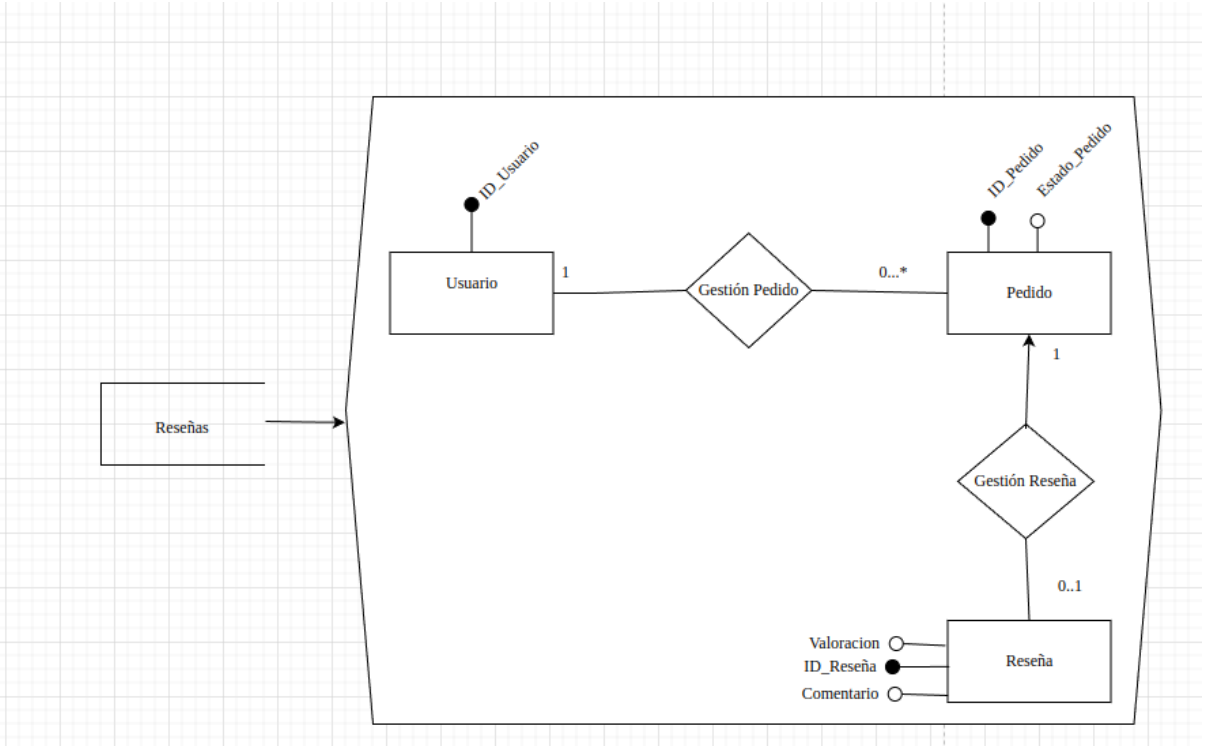



ID_Pedido
Pedido
1
Ver reseña pedido (5.4)
Gestión Reseña
0..1
ID_Reseña
Valoración
Comentario
Reseña

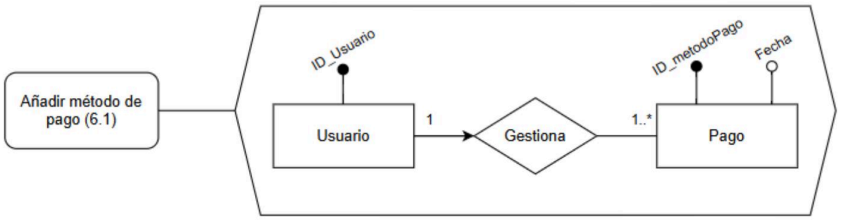

ID_Usuario
ID_Pedido
Usuario
1
Gestión Pedido
0...*
Pedido
Ver reseñas de un usuario (5.5)
1
Gestión Reseña
0..1
ID_Reseña
Valoración
Comentario
Reseña

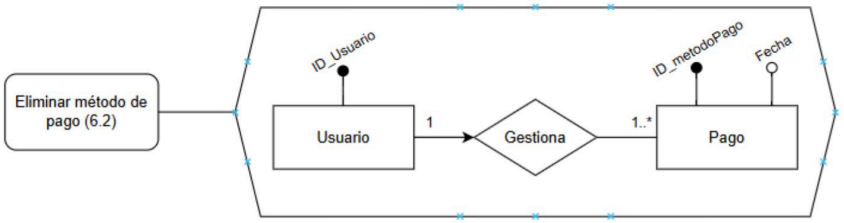

Usuarios
ID_Usuario
Nombre
Correo
Contraseña
Dirección
Estado
Usuario

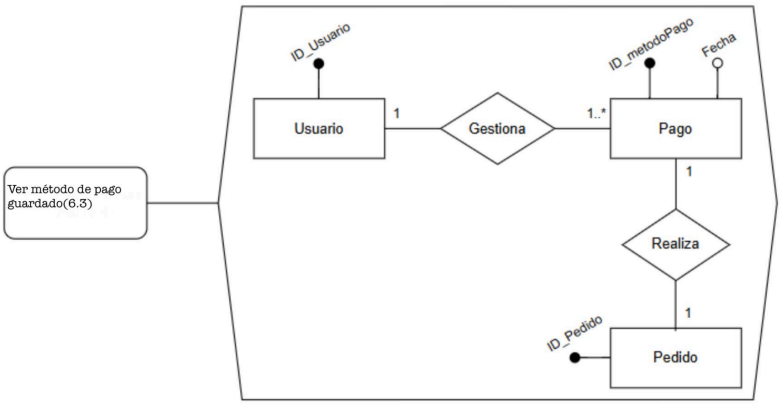

Pedidos
ID_Usuario
Dirección
ID_Pedido
Estado_pedido
Tipo_Pago
Metodo_envio
Usuario
Gestion Pedidos
Pedidos

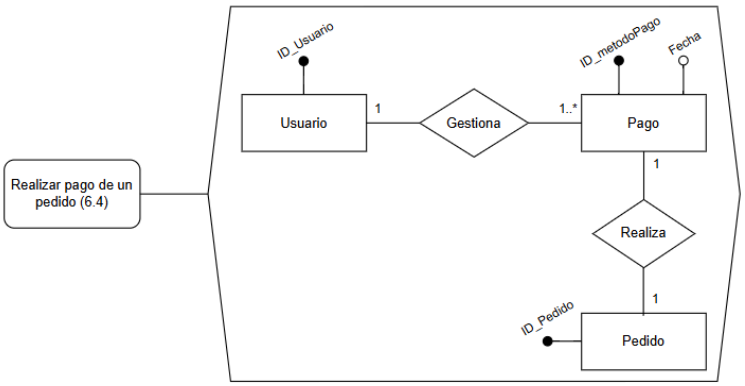

Reseñas
ID_Usuario
ID_Pedido
Estado_Pedido
Usuario
1
Gestión Pedido
0...*
Pedido
1
Gestión Reseña
0..1
Valoracion
ID_Reseña
Comentario
Reseña

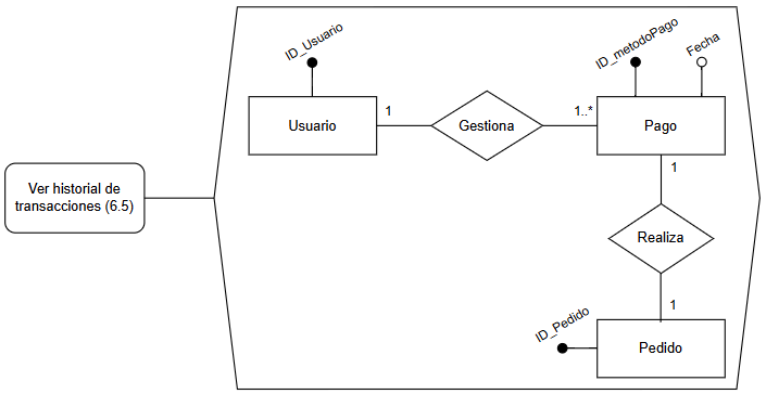
Subsistema Gestión Pagos (Laura Riera Ojea)





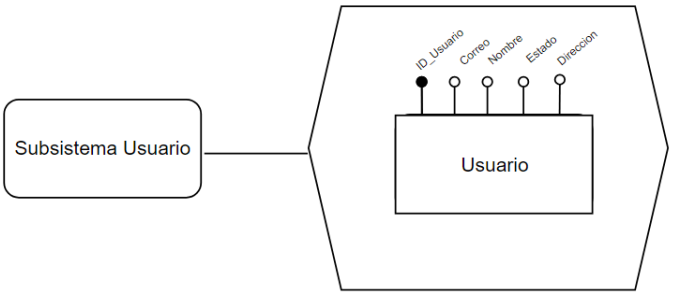




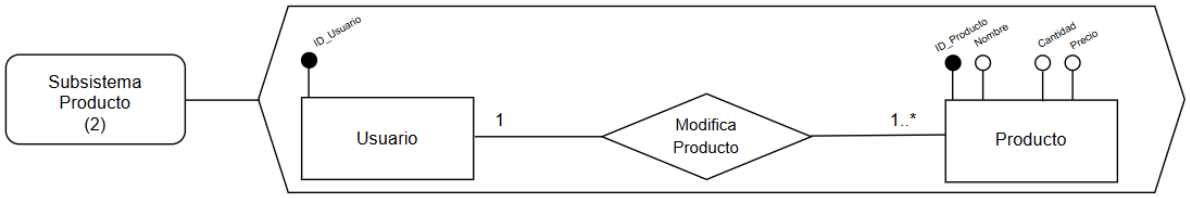


DFD0s EXTERNOS

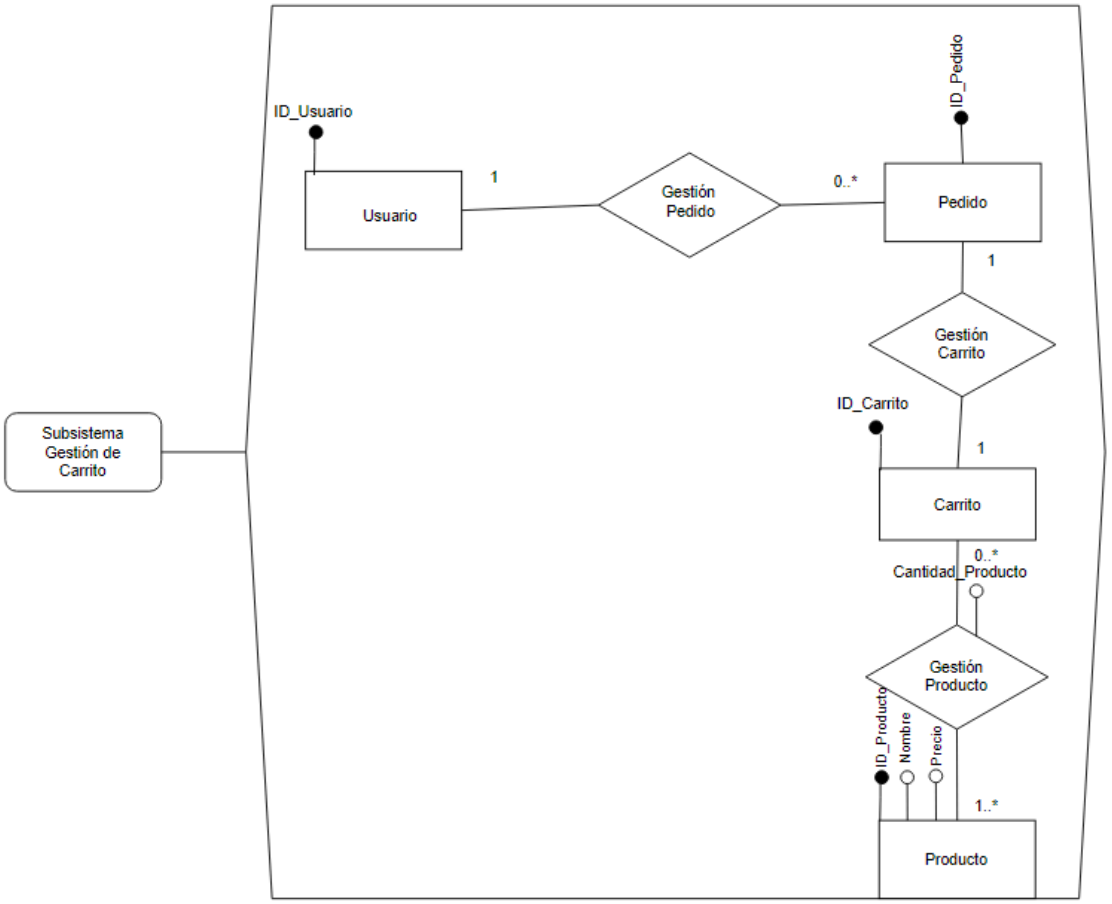
DFD0 SUBSISTEMA 1 (ALEXIS CALVENTE)

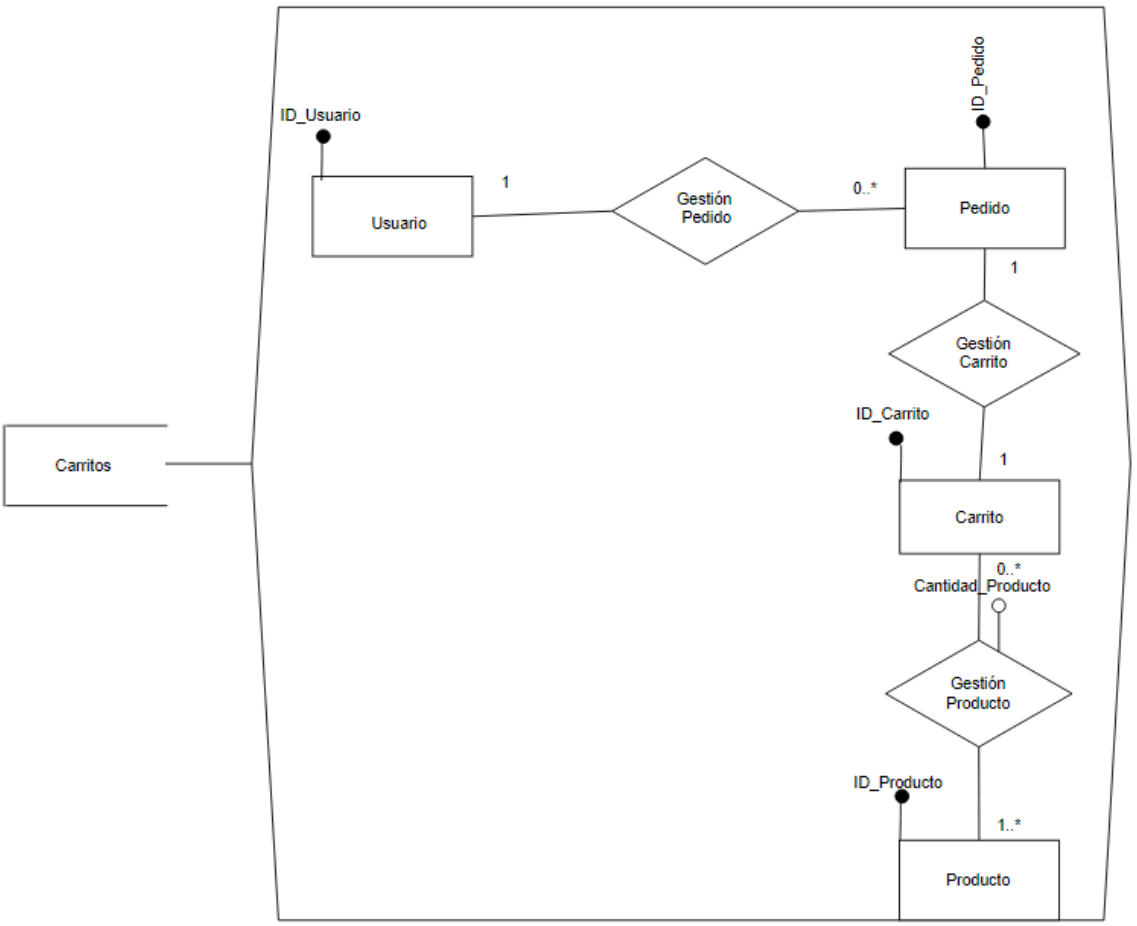


DFD0 SUBSISTEMA 2 (MIGUEL VELASCO)

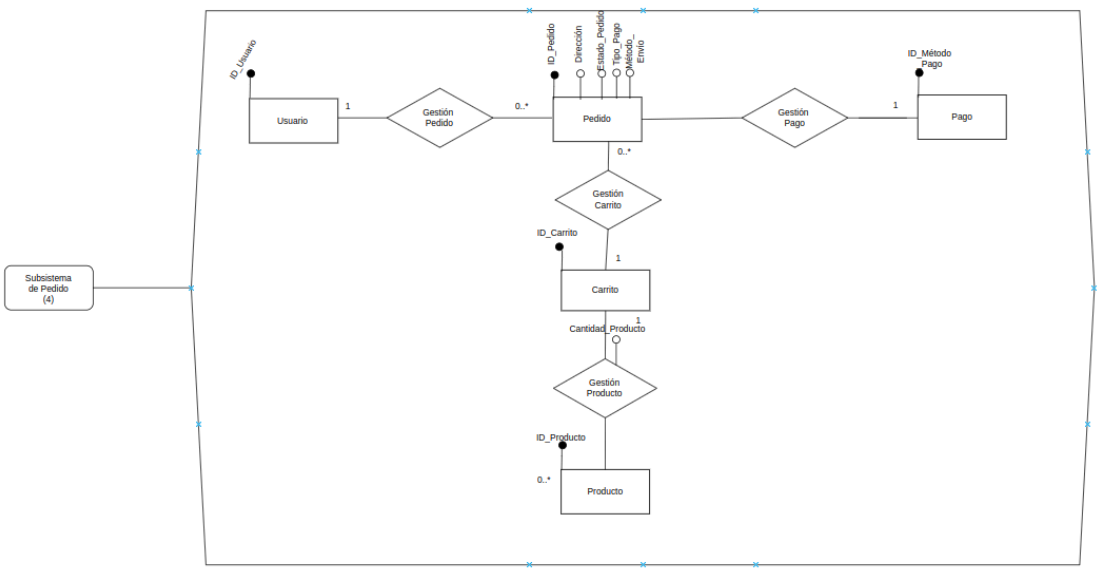


DFD0 SUBSISTEMA 3 (GABRIEL OLIVEROS)

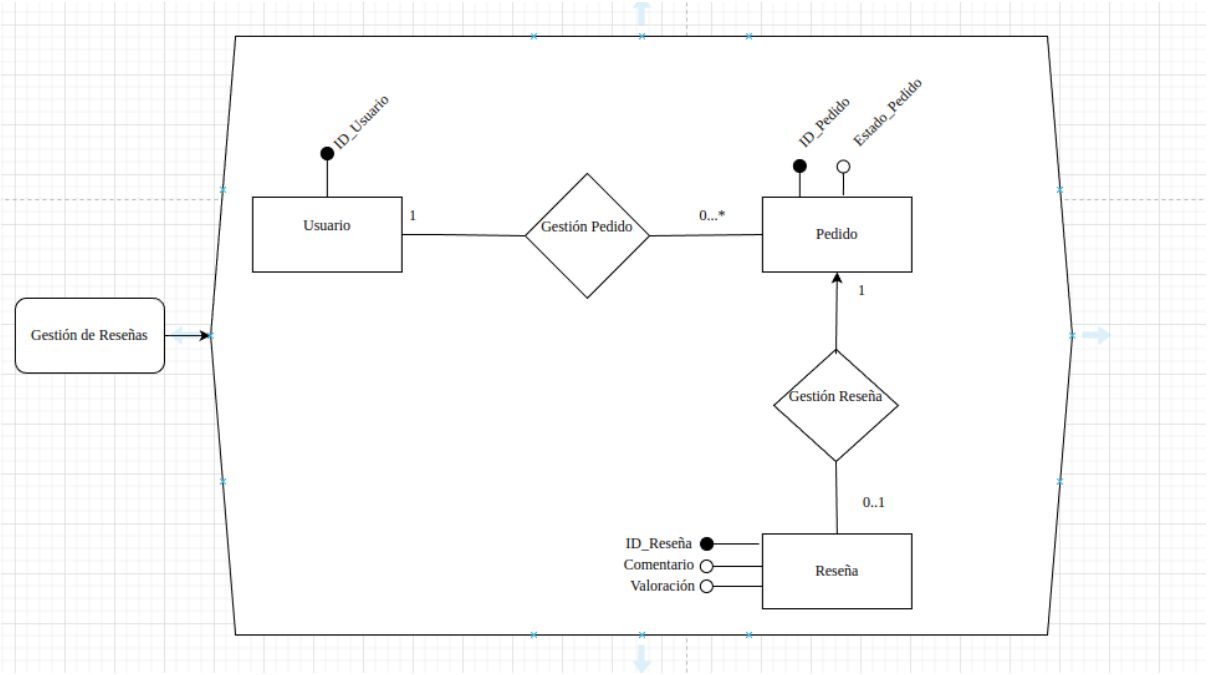




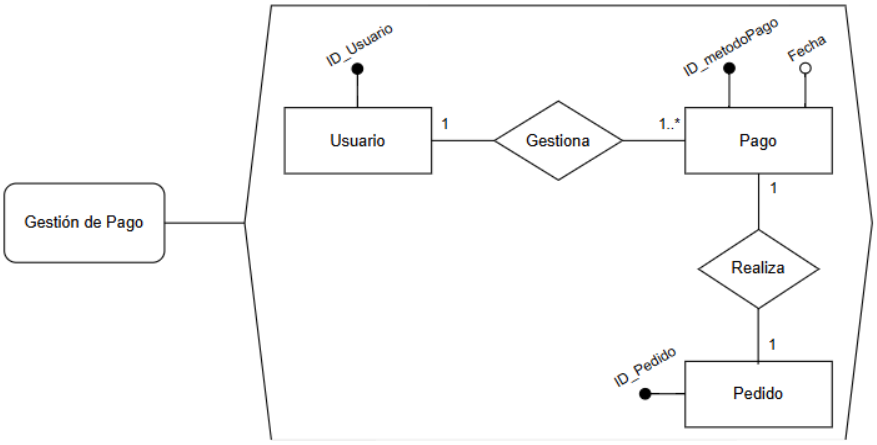
DFD0 SUBSISTEMA 4 (PEDRO ANTONIO PRIETO)



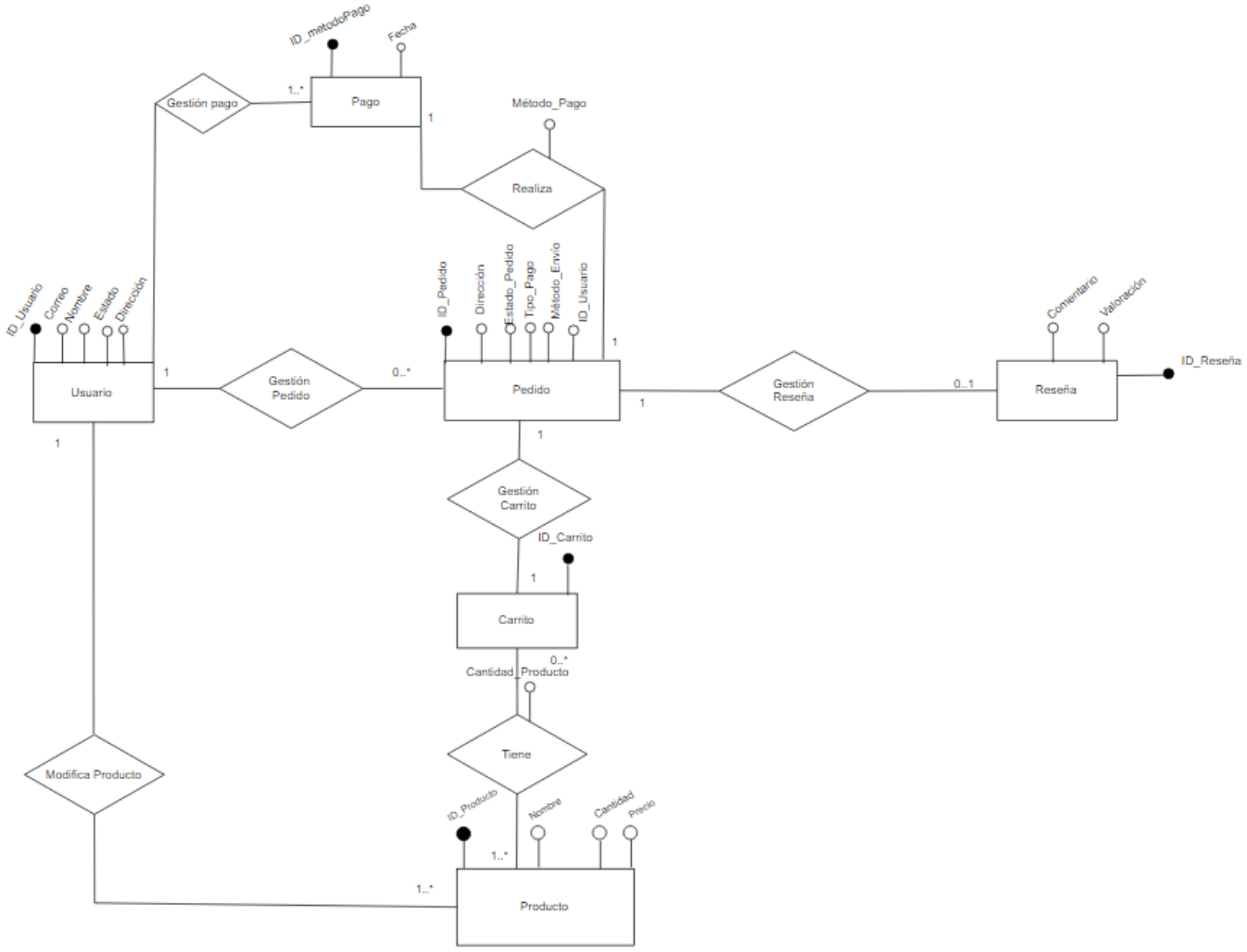
DFD0 SUBSISTEMA 5 (FLORIN EMANUEL TODOR)



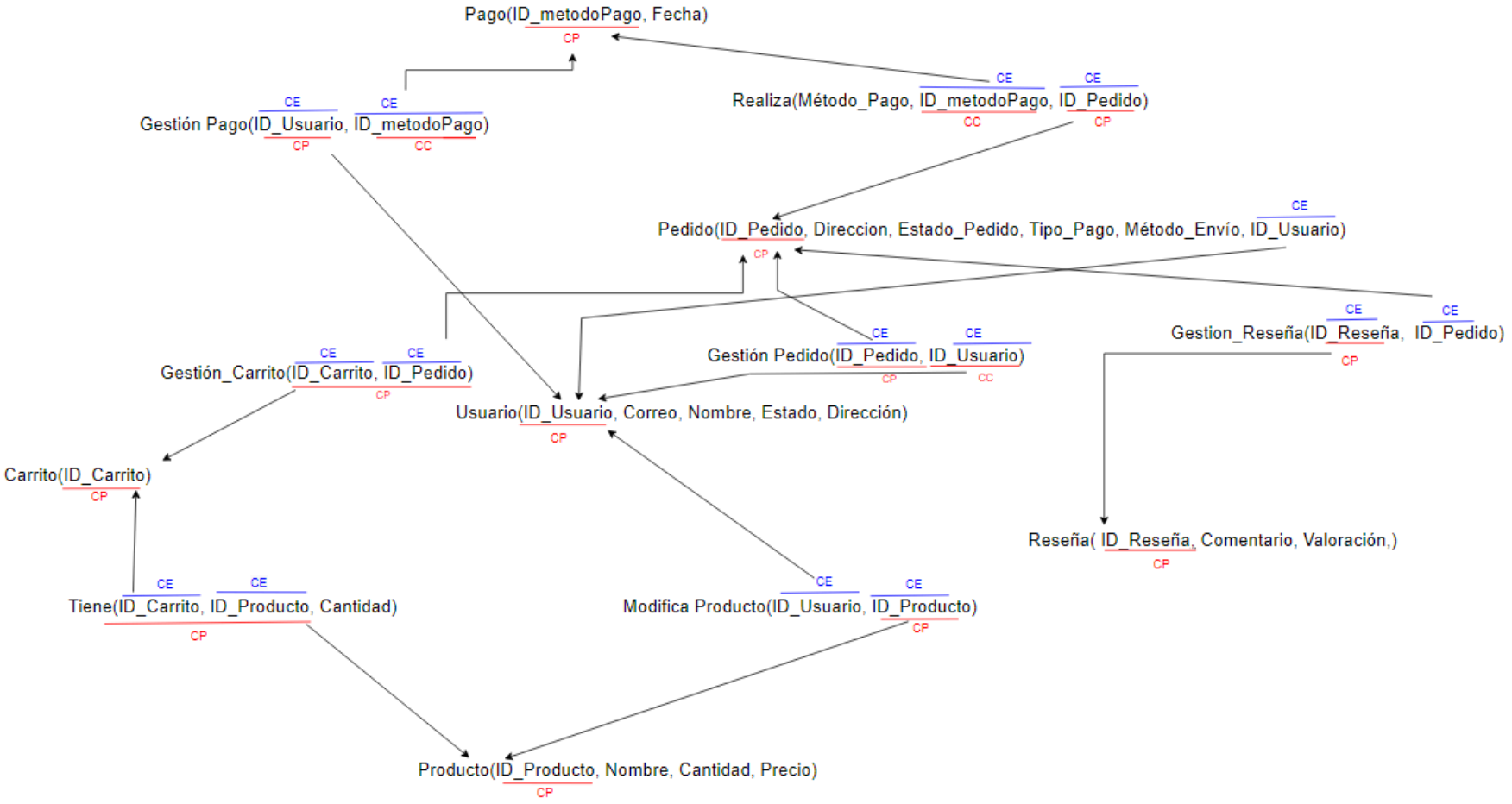
DFD0 SUBSISTEMA 6 (LAURA RIERA)



ESQUEMA EXTERNO ARMAZÓN



ESQUEMA PASO A TABLA



TABLAS Y NORMALIZACIÓN

Tabla 1:
Usuario(ID_Usuario, Correo, Nombre, Estado, Dirección)

- **Claves candidatas:** {ID_Usuario}
- **Dependencias funcionales:**
  - {ID_Usuario -> Correo}
  - {ID_Usuario -> Nombre}
  - {ID_Usuario -> Estado}
  - {ID_Usuario -> Dirección}

- **Normalización:** Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave, y por tanto, diremos que está en FNBC(y también estará en 3FN, 2FN y 1FN).

Tabla 2:
Pedido(ID_Pedido,Dirección, Estado_Pedido, Tipo_Pago, Método_Envío, ID_Usuario)

- **Claves candidatas:** {ID_Pedido}
- **Dependencias funcionales:**
  - {ID_Pedido -> Dirección}
  - {ID_Pedido -> Estado_Pedido}
  - {ID_Pedido -> Tipo_Pago}
  - {ID_Pedido -> Método_Envio}
  - {ID_Pedido -> ID_Usuario}

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave/clave_primaria , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN).

Tabla 3:
Producto(ID_Producto, Nombre, Cantidad, Precio)

- **Claves candidatas:** {ID_Producto}
- **Dependencias funcionales:**
  - {ID_Producto -> Nombre}
  - {ID_Producto -> Cantidad}
  - {ID_Producto -> Precio}
- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave, y por tanto, diremos que está en FNBC(y también estará en 3FN, 2FN y 1FN).

Tabla 4:
Carrito(ID_Carrito)

- **Claves candidatas:** {ID_Carrito}

- **Normalización**: Como en esta tabla solo hay un atributo, diremos que está en FNBC.

Tabla 5:
Pago(ID_metodoPago, Fecha, ID_Usuario, Tipo_MetodoPago, Numero_Tarjeta, Fecha_Expiracion, Codigo_CVV, Nombre_Titular, Correo_PayPal)

- **Claves candidatas:** {ID_metodoPago}
- **Dependencias funcionales:**
  - {ID_metodoPago -> Fecha}

- Normalización: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave, y por tanto, diremos que está en FNBC(y también estará en 3FN, 2FN y 1FN).

Tabla 6:
Reseña(ID_Reseña,Comentario, Valoración)

- **Claves candidatas:** {ID_Reseña}
- **Dependencias funcionales:**
  - {ID_Reseña -> Comentario}
  - {ID_Reseña -> Valoración}

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave/clave_primaria , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN).

Tabla 7:
Gestión Pago(ID_Usuario, ID_metodoPago)

- **Claves candidatas:** {ID_Usuario, ID_metodoPago}
- **Dependencias funcionales:**
  - {ID_Usuario -> ID_metodoPago}
  - {ID_metodoPago ->ID_Usuario}
  -

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave, y por tanto, diremos que está en FNBC(y también estará en 3FN, 2FN y 1FN). Podríamos decir también que como sólo cuenta con dos atributos y ambos son CC, la tabla estará en FNBC.

Tabla 8:
Gestión Pedido(ID_Pedido, ID_Usuario)

- **Claves candidatas:** {ID_Pedido, ID_Usuario}
- **Dependencias funcionales:**
  - {ID_Pedido -> ID_ Usuario}
  - {ID_Usuario -> ID_Pedido}

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN). Podríamos decir también que como sólo cuenta con dos atributos y ambos son CC, la tabla estará en FNBC.

Tabla 9:
Tiene(ID_Carrito, ID_Producto, Cantidad)

- **Claves candidatas:** {{ID_Carrito, ID_Producto}}
- **Dependencias funcionales:**
  - {ID_Carrito, ID_Producto -> Cantidad}

- Normalización: Para la única dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave, y por tanto, diremos que está en FNBC.

Tabla 10:
Gestion_Reseña(ID_Reseña, ID_Pedido)

- **Claves candidatas:** {ID_Reseña}
- **Dependencias funcionales:**
  - {ID_Reseña-> ID_Pedido}

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave/clave_primaria , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN).

Tabla 11:
Modifica Producto(ID_Usuario, ID_Producto)

- **Claves candidatas:** {ID_Producto}
- **Dependencias funcionales:**
  - {ID_Producto -> ID_Usuario }

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave/clave_primaria , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN).

Tabla 12:
Gestión Carrito(ID_Carrito, ID_Pedido)

- **Claves candidatas:** {ID_Carrito, ID_Pedido}

- **Normalización**: No hay dependencias funcionales no triviales. La tabla está en FNBC (y también estará en 3FN, 2FN y 1FN).

Tabla 13:
Realiza(ID_Pedido, ID_metodoPago)

- **Claves candidatas:** {ID_Pedido, ID_metodoPago}
- **Dependencias funcionales:**
  - {ID_Pedido-> ID_metodoPago}
  - {ID_metodoPago -> ID_Pedido}

- **Normalización**: Para toda dependencia funcional no trivial descrita anteriormente, a la izquierda hay una superclave/clave_primaria , y por tanto, diremos que está en FNBC (y también estará en 3FN, 2FN y 1FN)

PRÁCTICA 3

1. Tablas en SQL

```
-- Subsistema Usuario
CREATE TABLE usuario (
   ID_Usuario integer NOT NULL,
   Correo varchar(30),
   Nombre varchar(30),
   Estado CHAR(1) CHECK (Estado IN ('A', 'I')),
   Direccion varchar(50),
   Contraseña varchar(50),
   Fecha_Registro TIMESTAMP DEFAULT SYSTIMESTAMP,
   Fecha_Desactivacion TIMESTAMP,
   PRIMARY KEY(ID_Usuario),
   UNIQUE(Correo)
);

CREATE SEQUENCE pago_seq
START WITH 1      -- Valor inicial
INCREMENT BY 1    -- De cuánto en cuánto incrementa
NOCACHE           -- Opcional
NOCYCLE;

-- Subsistema Producto
CREATE TABLE producto (
   ID_Producto integer NOT NULL,
   NombreProducto varchar(30),
   Cantidad integer check ( cantidad >= 0 ),
   Precio float check ( precio >= 0 ),
   PRIMARY KEY(ID_Producto)
);

-- Subsistema Carrito
CREATE TABLE carrito (
   ID_Carrito integer NOT NULL,
   PRIMARY KEY(ID_Carrito)
);
```

```
-- Subsistema Pedido
CREATE TABLE pedido (
   ID_Pedido integer NOT NULL,
   Direccion varchar(30),
   Estado_Pedido varchar(10),
   Tipo_Pago varchar(30),
   Metodo_Envio varchar(10),
   ID_Usuario integer REFERENCES usuario(ID_Usuario),
   PRIMARY KEY(ID_Pedido)
);

-- Subsistema Reseña
CREATE TABLE reseña (
   ID_Resena INT PRIMARY KEY, -- Evitamos caracteres especiales
   Comentario VARCHAR2(500),
   Valoracion INT
);

-- Subsistema Pagos
CREATE TABLE pago (
   ID_metodoPago INT PRIMARY KEY,
   ID_Usuario int,
   Fecha DATE
);

-- Relación: Gestión de Pagos
CREATE TABLE GestionPago (
   ID_Usuario INT,
   ID_metodoPago INT,
   PRIMARY KEY (ID_Usuario),
   UNIQUE (ID_metodoPago),
   FOREIGN KEY (ID_Usuario) REFERENCES usuario(ID_Usuario),
   FOREIGN KEY (ID_metodoPago) REFERENCES pago(ID_metodoPago)
);

-- Relación: Realiza
CREATE TABLE Realiza (
   ID_metodoPago INT,
   ID_Pedido INT,
   Metodo_Pago VARCHAR(30),
   PRIMARY KEY (ID_Pedido),
   UNIQUE (ID_metodoPago),
   FOREIGN KEY (ID_Pedido) REFERENCES pedido(ID_Pedido),
   FOREIGN KEY (ID_metodoPago) REFERENCES pago(ID_metodoPago)
);

-- Relación: Gestión de Reseñas
CREATE TABLE Gestion_Reseña (
```

```
    ID_Resena INT, -- Evitamos caracteres especiales
    ID_Pedido INT,
    PRIMARY KEY (ID_Resena),
    FOREIGN KEY (ID_Resena) REFERENCES reseña(ID_Resena),
    FOREIGN KEY (ID_Pedido) REFERENCES pedido(ID_Pedido)
);



-- Relación: Modificación de Productos
CREATE TABLE modificaProducto (
    ID_Usuario integer REFERENCES usuario(ID_Usuario),
    ID_Producto integer REFERENCES producto(ID_Producto),
    PRIMARY KEY(ID_Producto)
);

-- Relación: Productos en Carrito
CREATE TABLE tiene (
    ID_Carrito integer REFERENCES carrito(ID_Carrito),
    ID_Producto integer REFERENCES producto(ID_Producto),
    Cantidad integer,
    PRIMARY KEY(ID_Carrito, ID_Producto)
);

-- Relación: Gestión de Carrito
CREATE TABLE GestionCarrito (
    ID_Carrito INTEGER NOT NULL,
    ID_Pedido INTEGER NOT NULL,
    PRIMARY KEY (ID_Carrito, ID_Pedido),
    FOREIGN KEY (ID_Carrito) REFERENCES CARRITO(ID_Carrito),
    FOREIGN KEY (ID_Pedido) REFERENCES PEDIDO(ID_Pedido)
);

-- Relación: Gestión de Pedido
CREATE TABLE GestionPedido (
    ID_Usuario integer REFERENCES usuario(ID_Usuario),
    ID_Pedido integer REFERENCES pedido(ID_Pedido),
    PRIMARY KEY(ID_Pedido),
    UNIQUE(ID_Usuario)
);

--Secuencia para los id_pedido
CREATE SEQUENCE seq_id_pedido START WITH 1 INCREMENT BY 1;
--Secuencia para los id_pedido
CREATE SEQUENCE seq_id_pedido START WITH 1 INCREMENT BY 1;
```

2. Descripción de las transacciones identificadas:

Transacciones subsistema de Usuario(Alexis Calvente Galvin)

1. registerUser(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para registrar un nuevo usuario en la Base de Datos.

Validaciones:

- Revisa la conexión.
- Verifica que el correo proporcionado no esté ya registrado.
- Obtiene el próximo ID disponible para el nuevo usuario.

Operaciones de inserción:

- INSERT INTO USUARIO para agregar al usuario con los datos proporcionados.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto (inserción exitosa).
- conn.rollback() en caso de excepción.

2. deleteUser(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para deshabilitar la cuenta de un usuario.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario exista.
- Comprueba que la contraseña proporcionada sea correcta.
- Verifica que la cuenta no esté ya deshabilitada.

Operaciones de actualización:

- UPDATE USUARIO para cambiar el estado del usuario a I (inactivo) y registrar la fecha de desactivación.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

3. updateUser(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para modificar los datos de un usuario.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario exista.
- Comprueba que el estado del usuario sea activo (A).

Operaciones de actualización:

- UPDATE USUARIO para modificar los datos del usuario (correo, nombre, dirección, contraseña).

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

4. recoverPassword(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para generar un token de recuperación de contraseña.

Validaciones:

- Revisa la conexión.
- Verifica que el correo proporcionado esté registrado y que la cuenta esté activa (A).

Operaciones de generación:

- Genera un token único y lo asocia al usuario en la Base de Datos.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

5. resetPassword(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para cambiar la contraseña de un usuario utilizando un token de recuperación.

Validaciones:

- Revisa la conexión.
- Verifica que el token proporcionado coincida con el token registrado para el correo.

Operaciones de actualización:

- UPDATE USUARIO para cambiar la contraseña y eliminar el token de recuperación.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

6. loginUser(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para autenticar a un usuario.

Validaciones:

- Revisa la conexión.
- Verifica que el correo y la contraseña proporcionados sean correctos.
- Comprueba que la cuenta esté activa (A) y no tenga fecha de desactivación.

Operaciones de lectura:

- SELECT USUARIO para obtener el ID del usuario autenticado.

Confirmación o anulación:

- No requiere conn.commit() ni conn.rollback(), ya que es una operación de lectura.
- En caso de error, se lanza una excepción.

Transacciones subsistema de Reseñas (Florín Emanuel Todor Gliga)

1. addReview(...)

Se inicia cuando el método es llamado y se comienzan a ejecutar las validaciones y las sentencias SQL.

Validaciones previas:

- Verifica que la conexión exista.
- Revisa si el usuario existe.
- Comprueba si el pedido corresponde al usuario y si está en estado 'Entregado'.
- Se asegura de que no exista ya una reseña para ese pedido.

Operaciones de inserción:

- INSERT INTO RESEÑA para crear la nueva reseña.
- INSERT INTO GESTION_RESEÑA para asociar la reseña al pedido.

Confirmación o anulación:

- Si todo sale bien, se hace conn.commit().
- Si cualquier validación o inserción falla (lanza excepción), se hace conn.rollback() y se relanza la excepción (throw e;).

2. editReview(...)

Se inicia cuando el método es llamado y se comienzan a ejecutar las validaciones y las sentencias SQL.

Validaciones:

- Comprueba la conexión.
- Verifica si el usuario existe.
- Comprueba que la reseña exista, que pertenezca al usuario y que no esté ya “eliminada” (valoración NULL).

Operación de modificación:

- UPDATE RESEÑA para cambiar la valoración y el comentario.

Confirmación o anulación:

- conn.commit() si todo va bien.
- conn.rollback() si se produce alguna excepción.

3. deleteReview(...)

Se inicia cuando el método es llamado y se comienzan a ejecutar las validaciones y las sentencias SQL.

Validaciones:

- Comprueba la conexión.
- Verifica si el usuario existe.
- Revisa que la reseña exista, que pertenezca al usuario, y que no esté ya eliminada (valoración NULL).

Operación de eliminación:

- UPDATE RESEÑA poniendo VALORACION = NULL y COMENTARIO = NULL.

Confirmación o anulación:

- conn.commit() si todo fue correcto.
- conn.rollback() si algo falla.

4. getReviewsByOrder(...)

Se inicia cuando el método es llamado y se comienzan a ejecutar las validaciones y las sentencias SQL.

Validaciones:

- Verifica la conexión.
- Comprueba que el pedido existe.

Operaciones de lectura:

- SELECT para obtener las reseñas asociadas al pedido.
- Si no hay reseñas, lanza una excepción.

Confirmación o anulación:

- Aunque es solo lectura, se llama igualmente a conn.commit() al final (para “cerrar” la transacción).
- Si ocurre un error, conn.rollback() y se relanza la excepción.

5. getReviewsByUser(...)

Se inicia cuando el método es llamado y se comienzan a ejecutar las validaciones y las sentencias SQL.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario existe.

Operaciones de lectura:

- SELECT para obtener las reseñas asociadas al usuario.
- Si no hay reseñas, se lanza una excepción.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto (lectura exitosa).
- conn.rollback() en caso de excepción.

Transacciones subsistema de Productos(Miguel Velasco Fernández)

1. addProduct(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para agregar un producto a la Base de Datos.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario existe.
- Comprueba que el precio y la cantidad sea mayor que 0. También comprueba que el producto no existe ya.

Operaciones de inserción:

- INSERT INTO Producto para agregar el producto a la tabla.
- INSERT INTO modificaProducto para crear la relación Usuario-Producto

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto (inserciones y validaciones exitosas).
- conn.rollback() en caso de excepción.

2. editProduct(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para editar un producto dentro de la Base de Datos.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario y el producto existen y que el usuario está asociado al producto que se quiere modificar.
- Comprueba que el precio y la cantidad nuevos sean mayores que 0.

Operaciones de actualización:

- UPDATE Producto para editar el producto de la tabla.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

3. deleteProduct(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para eliminar un producto dentro de la Base de Datos.En nuestro caso, eliminar es como cambiar la cantidad de producto a 0.

Validaciones:

- Revisa la conexión.
- Verifica que el usuario y el producto existen y que el usuario está asociado al producto que se quiere eliminar.
- Comprueba que el producto no haya sido eliminado ya.

Operaciones de actualización:

- UPDATE Producto para editarla cantidad de producto de la tabla.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

4. getProductsByUser(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para consultar un producto dentro de la Base de Datos según el usuario al que pertenece .

Validaciones:

- Revisa la conexión.
- Verifica que el usuario y el producto existen
- Comprueba que el usuario tenga productos asociados.

Operaciones de actualización:

- Select Producto para obtener los productos de la tabla asociados al usuario.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

5. getProductsByPrice(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para consultar un producto dentro de la Base de Datos según el precio que tiene.

Validaciones:

- Revisa la conexión.
- Verifica que el producto existe.
- Comprueba que haya productos con el precio especificado.

Operaciones de actualización:

- Select Producto para obtener los productos de la tabla con el precio requerido.

Confirmación o anulación:

- Se hace conn.commit() si todo es correcto.
- conn.rollback() en caso de excepción.

Transacciones subsistema de Carrito(Gabriel Oliveros Jiménez)

1. getProductosDelCarrito(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas que devuelven los productos que contiene el carrito. Se usa en el subsistema de pedidos.

Operaciones de lectura:

- SELECT ID_Producto, Cantidad FROM tiene WHERE ID_Carrito = ?: Se obtienen los productos y sus cantidades del carrito con el idCarrito dado.

Confirmación o anulación:

- No se realiza conn.commit() explícitamente porque es una operación de solo lectura y la conexión sigue abierta. Se deja la gestión de la transacción a la capa superior.
- SQLException se propaga en caso de error.

2. crearRelacionCarritoPedido(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para crear un pedido vacío y una relación carrito-pedido con el pedido creado. Se llama al método como parte de la creación de un nuevo carrito.

Validaciones:

- Verifica la conexión.

Operaciones:

- Obtiene el CURRVAL de la secuencia seq_id_pedido. Si falla, intenta obtener NEXTVAL para inicializar la secuencia.
- Incrementa la secuencia seq_id_pedido usando NEXTVAL.
- INSERT INTO PEDIDO: Inserta un nuevo pedido con el idPedido obtenido de la secuencia y el idUsuario.
- INSERT INTO GESTIONCARRITO: Inserta un registro para asociar el idCarrito con el idPedido.

Confirmación o anulación:

- conn.rollback(): Esto asegura que, si una parte de la operación falla, se deshagan los cambios previos.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback). Se relanzan las excepciones en puntos clave.

3. addCarritoEntry(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para añadir un producto al carrito. Si el carrito no existe, se crea.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.

Operaciones:

- Obtiene el CURRVAL de la secuencia seq_id_carrito. Si falla, intenta obtener NEXTVAL para inicializar la secuencia.
- Incrementa la secuencia seq_id_carrito usando NEXTVAL.
- INSERT INTO CARRITO: Inserta un nuevo carrito con el idCarrito obtenido.

- Llama a crearRelacionCarritoPedido(idUsuario, idCarrito) para asociar el carrito a un nuevo pedido.

Confirmación o anulación:

- conn.commit(): Confirma la transacción si todo se ejecuta correctamente.
- conn.rollback(): Se realiza un rollback explícito si hay una excepción.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback). Se relanzan las excepciones en puntos clave.

4. getCarritoId(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para obtener el id del carrito a partir del id del usuario.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.

Operaciones de lectura:

- SELECT MAX(ID_Pedido) FROM pedido WHERE ID_Usuario = ?: Obtiene el ID del pedido más reciente del usuario.
- SELECT ID_Carrito FROM GestionCarrito WHERE ID_Pedido = ?: Obtiene el idCarrito asociado al idPedido más reciente.

Confirmación o anulación:

- No se realiza conn.commit() explícitamente. Se asume que la transacción se maneja en un nivel superior.
- Exception se propaga en caso de error, incluyendo SQLException.

5. addProductToCart(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para agregar un producto al carrito.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.
- Verifica que el producto exista.
- Verifica que la cantidad solicitada no exceda el stock disponible.
- Verifica que el producto no esté ya en el carrito.

Operaciones:

- getCarritoId(idUsuario): Obtiene el ID del carrito. Si el usuario no tiene carrito, llama a la función addCarritoEntry para crear uno nuevo.
- INSERT INTO TIENE: Inserta el producto, el ID del carrito y la cantidad en la tabla TIENE.

Confirmación o anulación:

- conn.commit(): Confirma la transacción si todo se ejecuta correctamente.
- conn.rollback(): Se realiza un rollback explícito si hay una excepción.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback).

6. viewCart(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para mostrar los productos del carrito.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.

Operaciones de lectura:

- getCarritoId(idUsuario): Obtiene el ID del carrito del usuario.
- SELECT ... FROM TIENE ... JOIN PRODUCTO ...: Obtiene los productos del carrito, incluyendo su nombre, precio y cantidad. Calcula el total por producto y el subtotal del carrito.

Confirmación o anulación:

- No se realiza conn.commit() explícitamente, al ser una operación de lectura.
- Exception se propaga en caso de error.

7. modifyCartQuantity(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para modificar la cantidad de un producto en el carrito.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.
- Verifica que el producto exista.
- Verifica que la nueva cantidad sea válida (no negativa y no mayor al stock).
- Verifica que el producto esté en el carrito.

Operaciones:

- getCarritoId(idUsuario): Obtiene el ID del carrito.
- UPDATE TIENE: Actualiza la cantidad del producto en el carrito.

Confirmación o anulación:

- conn.commit(): Confirma la transacción.
- conn.rollback(): Se realiza un rollback explícito si hay una excepción.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback).

8. removeProductFromCart(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para eliminar un producto del carrito.

Validaciones:

- Verifica la conexión.
- Verifica que el usuario exista.
- Verifica que el producto esté en el carrito.

Operaciones:

- getCarritoId(idUsuario): Obtiene el ID del carrito.
- DELETE FROM TIENE: Elimina el producto del carrito.

Confirmación o anulación:

- conn.commit(): Confirma la transacción.

- conn.rollback(): Se realiza un rollback explícito si hay una excepción.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback).

9. emptyCart(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para vaciar el carrito.

Validaciones:

- Verifica la conexión.
- Verifica si el carrito ya está vacío.

Operaciones:

- getCarritoId(idUsuario): Obtiene el ID del carrito.
- DELETE FROM TIENE: Elimina todos los productos del carrito.

Confirmación o anulación:

- conn.commit(): Confirma la transacción.
- conn.rollback(): Se realiza un rollback explícito si hay una excepción.
- Exception se propaga en caso de error (incluyendo SQLException tras el rollback).

Transacciones subsistema de Pedido(Pedro Antonio Prieto Fernández)

1. realizarPedido(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para completar un pedido

Validaciones:

- Verifica la conexión
- Verifica el método de envío
- Verifica la existencia del usuario
- Verifica la existencia del carrito
- Verifica si el carrito está vacío
- Verifica el stock y el estado de los productos

Operaciones:

- Iniciar transacción
- Obtener el máximo ID de pedido
- Actualizar el pedido
- Establece el estado del pedido como "procesando"
- Insertar en GestionPedido
- Actualizar el stock de los productos:
- Añadir una nueva entrada al carrito

Confirmación o anulación:

- conn.commit(): Si todas las operaciones se completan sin errores, se confirma la transacción.

- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción

2. verHistorialPedidos(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para obtener y devolver el historial de pedidos de un usuario.

Validaciones:

- Verifica la conexión: Se asume que la conexión conn es válida y está establecida.
- Verifica implícitamente la existencia del usuario: No se realiza una comprobación explícita de la existencia del usuario. Si el usuario no existe se devolverá una lista vacía

Operaciones:

- Obtener los pedidos del usuario:
- Para cada pedido encontrado:
    - Obtener los productos asociados a cada pedido:
    - Crear un objeto Pedido.
    - Agregar el pedido a la lista:

Confirmación o anulación:

No se realiza conn.commit() ni conn.rollback() explícitamente: Se trata de una operación de solo lectura

3. cancelarPedido(...)

Es la función que actúa dentro del programa para realizar las operaciones SQL requeridas para cancelar un pedido existente.

Validaciones:

- Verifica la conexión
- Verifica la existencia del pedido
- Verifica la pertenencia del pedido al usuario
- Verifica el estado del pedido:
    - Comprueba si el estado del pedido es "entregado". Si es así, lanza una SQLException indicando que no se puede cancelar un pedido entregado.
    - Comprueba si el estado del pedido es "cancelado". Si es así, lanza una SQLException indicando que el pedido ya está cancelado.

Operaciones:

- Iniciar transacción
- Actualizar el estado del pedido:
- Obtener los productos y cantidades del pedido
- Restablecer el stock de los productos:

Confirmación o anulación:

- conn.commit(): Si todas las operaciones se completan sin errores, se confirma la transacción.
- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción para deshacer cualquier cambio realizado en la base de datos.

4. elegirMetodoEnvio(...)

Este método permite al usuario elegir o modificar el método de envío asociado a un pedido específico.

Validaciones:

- Verifica la conexión:
- Verifica la pertenencia del pedido al usuario

Operaciones:

- Iniciar transacción
- Actualizar el método de envío

Confirmación o anulación:

- conn.commit(): Si la operación de actualización se completa correctamente, se confirma la transacción con conn.commit().
- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción con conn.rollback(), deshaciendo cualquier cambio realizado.

5. confirmarRecepcionPedido(...)

Este método permite a un usuario confirmar la recepción de un pedido, actualizando su estado a "entregado".

Validaciones:

- Verifica la conexión
- Verifica la existencia del pedido
- Verifica la pertenencia del pedido al usuario
- Verifica el estado del pedido no sea cancelado

Operaciones:

- Iniciar transacción
- Actualizar el estado del pedido

Confirmación o anulación:

- conn.commit(): Si la operación de actualización se completa correctamente, se confirma la transacción con conn.commit().
- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción con conn.rollback(), deshaciendo cualquier cambio realizado.

6. elegirMetodoPago(...)

Este método permite a un usuario elegir o modificar el método de pago asociado a un pedido específico.

Validaciones:

- Verifica la conexión
- Verifica la pertenencia del pedido al usuario

Operaciones:

- Iniciar transacción
- Actualizar el método de pago:

Confirmación o anulación:

- conn.commit(): Si la operación de actualización se completa correctamente, se confirma la transacción con conn.commit().
- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción con conn.rollback(), deshaciendo cualquier cambio realizado.

1. Agregar Método de Pago (agregarMetodoPago)

Vamos a agregar un método de pago que podemos elegir entre paypal y tarjeta de crédito y ponemos los datos

Validaciones Previas:

- Verifica que la conexión a la base de datos esté activa.
- Comprueba que el usuario exista en la base de datos.
- Si el método de pago es tarjeta:
- Verifica que el número de tarjeta tenga 16 dígitos y sea válido.
- Comprueba que el código CVV sea válido (3 o 4 dígitos).
- Valida que la fecha de expiración esté en el formato correcto (MM/AA).
- Si el método es PayPal:
- Comprueba que el correo electrónico tenga un formato válido.

Operaciones de Inserción:

- Inserta los detalles del método de pago en la tabla PAGO.

Confirmación o Anulación:

- conn.commit(): Si todas las validaciones y la inserción son exitosas.
- conn.rollback(): Si se produce una SQLException durante la ejecución, se revierte la transacción con conn.rollback(), deshaciendo cualquier cambio realizado.

2. Eliminar Método de Pago (eliminarMetodoPago)

Aquí eliminamos el método de pago de un usuario.

Validaciones Previas:

- Verifica que la conexión a la base de datos esté activa.
- Comprueba que el método de pago pertenece al usuario.
- Verifica si el método de pago ya ha sido utilizado en la tabla REALIZA.
- Si ha sido usado, lanza una excepción indicando que no puede ser eliminado.

Operaciones de Eliminación:

- Elimina el registro del método de pago de la tabla PAGO.

Confirmación o Anulación:
- conn.commit(): Si todas las validaciones y la eliminación son exitosas.
- conn.rollback(): Si ocurre cualquier error o excepción.

3. Ver Métodos de Pago Asociados a un Usuario (verMetodosPago)

Aquí podemos ver los métodos de pago guardados asociados a un usuario.

Validaciones Previas:
- Verifica que la conexión a la base de datos esté activa.
- Comprueba que el usuario exista en la base de datos.

Operaciones de Lectura:
- Realiza un SELECT para obtener los métodos de pago asociados al usuario.
- Devuelve información relevante como tipo de método y terminación de tarjeta.

Confirmación o Anulación:
- No requiere conn.commit() ni conn.rollback(), ya que es una operación de lectura.

Errores:
- Si no hay métodos de pago asociados, lanza una excepción indicando esto.

4. Realizar Pago (realizarPago)

Realizamos un pago, deberemos poner el identificador del producto y la cantidad a pagar.

Validaciones Previas:
- Verifica que la conexión a la base de datos esté activa.
- Comprueba que el método de pago pertenece al usuario.
- Valida que el pedido exista y esté pendiente de pago.
- Confirma que la cantidad especificada sea correcta.

Operaciones de Inserción y Actualización:
- Inserta un registro en la tabla REALIZA, asociando el método de pago con el pedido.
- Actualiza el estado del pedido a pagado en la tabla PEDIDO.

Confirmación o Anulación:
- conn.commit(): Si todas las validaciones e inserciones son exitosas.
- conn.rollback(): Si ocurre cualquier error o excepción durante la ejecución.

5. Ver Historial de Transacciones (verHistorialTransacciones)

Aquí podemos ver el historial de transacciones que ha realizado un usuario.

Validaciones Previas:

- Verifica que la conexión a la base de datos esté activa.
- Comprueba que el usuario exista en la base de datos.

Operaciones de Lectura:

- Realiza un SELECT para obtener el historial de transacciones del usuario.
- Devuelve información como el pedido asociado, cantidad pagada y método de pago utilizado.

Confirmación o Anulación:

- No requiere conn.commit() ni conn.rollback(), ya que es una operación de lectura.

Errores:

- Si no hay transacciones registradas, lanza una excepción indicando esto.

3.Disparadores:

```
package practica;

import javax.swing.*;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Disparadores {

   // 1) TRIGGER TRIG_VerificaPedidoYUsuario FLorin
   private static final String TRIG_VERIFICA_PEDIDO_USUARIO =
           "CREATE OR REPLACE TRIGGER TRIG_VerificaPedidoYUsuario
"
                   + "BEFORE INSERT OR UPDATE ON Gestion_Reseña "
                   + "FOR EACH ROW "
                   + "DECLARE "
                   + "   v_contador NUMBER; "
                   + "BEGIN "
                   + "   -- 1) Verificar que el ID_Pedido exista
en la tabla Pedido "
                   + "   SELECT COUNT(*) INTO v_contador "
                   + "     FROM pedido "
                   + "    WHERE ID_Pedido = :NEW.ID_Pedido; "
                   + "   IF v_contador = 0 THEN "
                   + "      RAISE_APPLICATION_ERROR(-20000, "
                   + "          'Error RS5.1.1: El ID_Pedido ' ||
:NEW.ID_Pedido || ' no existe en la tabla Pedido.' "
                   + "      ); "
                   + "   END IF; "
                   + "   -- 2) Verificar que el ID_Usuario
asociado a ese Pedido exista "
                   + "   SELECT COUNT(*) INTO v_contador "
                   + "     FROM pedido p "
```

```
                    + "          JOIN usuario u ON p.ID_Usuario =
u.ID_Usuario "
                    + "    WHERE p.ID_Pedido = :NEW.ID_Pedido; "
                    + "   IF v_contador = 0 THEN "
                    + "      RAISE_APPLICATION_ERROR(-20001, "
                    + "         'Error RS5.1.1: El Pedido ' ||
:NEW.ID_Pedido || ' no tiene un Usuario válido asociado.' "
                    + "      ); "
                    + "   END IF; "
                    + "END;";

   // 2) TRIGGER validar_usuario_en_modificaProducto
   private static final String
TRIG_VALIDAR_USUARIO_MODIFICAPRODUCTO =
           "CREATE OR REPLACE TRIGGER
validar_usuario_en_modificaProducto "
                    + "AFTER INSERT OR UPDATE ON modificaProducto "
                    + "FOR EACH ROW "
                    + "DECLARE "
                    + "   usuario_existe INTEGER; "
                    + "BEGIN "
                    + "   -- Verificar si el usuario asociado al
producto existe "
                    + "   SELECT COUNT(*) INTO usuario_existe "
                    + "     FROM usuario "
                    + "    WHERE ID_Usuario = :NEW.ID_Usuario; "
                    + "   IF usuario_existe = 0 THEN "
                    + "      -- Opcional: Lanzar un error si se
requiere notificar el problema "
                    + "   RAISE_APPLICATION_ERROR(-20002, " //este
error hace un rollback automático
                    + "      'El usuario asociado al producto no
existe.' "
                    + "   ); "
                    + "   END IF; "
                    + "END;";

   // 3) TRIGGER validar_producto_en_modificaProducto
   private static final String
TRIG_VALIDAR_PRODUCTO_MODIFICAPRODUCTO =
           "CREATE OR REPLACE TRIGGER
validar_producto_en_modificaProducto "
                    + "AFTER INSERT OR UPDATE ON modificaProducto "
                    + "FOR EACH ROW "
                    + "DECLARE "
                    + "   producto_existe INTEGER; "
                    + "BEGIN "
```

```
                    + "    -- Verificar si el producto existe en la
tabla producto "
                    + "    SELECT COUNT(*) INTO producto_existe "
                    + "      FROM producto "
                    + "     WHERE ID_Producto = :NEW.ID_Producto; "
                    + "    IF producto_existe = 0 THEN "
                    + "    -- Opcional: Lanzar un error para
notificar el problema "
                    + "    RAISE_APPLICATION_ERROR(-20003, "
                    + "        'El producto asociado no existe s.' "
                    + "    ); "
                    + "    END IF; "
                    + "END;";

    // 4) TRIGGER validar_relacion_producto_modificaProducto
    /*private static final String
TRIG_VALIDAR_RELACION_PRODUCTO_MODIFICAPRODUCTO =
            "CREATE OR REPLACE TRIGGER
validar_relacion_producto_modificaProducto "
                    + "AFTER INSERT OR UPDATE ON producto "
                    + "FOR EACH ROW "
                    + "DECLARE "
                    + "    relacion_valida INTEGER; "
                    + "BEGIN "
                    + "    -- Verificar si el producto tiene una
relación válida en modificaProducto "
                    + "    SELECT COUNT(*) INTO relacion_valida "
                    + "      FROM modificaProducto "
                    + "     WHERE ID_Producto = :NEW.ID_Producto; "
                    + "    -- Opcional: Lanzar un error para
informar sobre la eliminación "
                    + "    RAISE_APPLICATION_ERROR(-20004, "
                    + "        'El producto no tiene una relación
válida en modificaProducto. Producto eliminado.' "
                    + "    ); "
                    + "END;";
*/
    // 5) TRIGGER garantizar que existe el usuario cuando vas a
añadir metodo pago
    private static final String TRIG_VERIFICAR_USUARIO_EXISTE =
            "CREATE OR REPLACE TRIGGER
TRIG_VERIFICAR_USUARIO_EXISTE " +
                    "BEFORE INSERT ON pago " +
                    "FOR EACH ROW " +
                    "DECLARE " +
                    "    usuario_existe NUMBER; " +
                    "BEGIN " +
                    "    SELECT COUNT(*) INTO usuario_existe " +
```

```
                    "    FROM usuario " +
                    "    WHERE ID_Usuario = :NEW.ID_Usuario; " +
                    "    IF usuario_existe = 0 THEN " +
                    "        RAISE_APPLICATION_ERROR(-20002,
'Error: El usuario asociado al método de pago no existe.'); " +
                    "    END IF; " +
                    "END;";

   // 6) TRIGGER garantizar si existe el metodo de pago cuando
vayas a hacer un pago
   private static final String TRIG_VERIFICAR_METODO_PAGO_EXISTE =
           "CREATE OR REPLACE TRIGGER
TRIG_VERIFICAR_METODO_PAGO_EXISTE " +
                    "BEFORE INSERT ON Realiza " +
                    "FOR EACH ROW " +
                    "DECLARE " +
                    "    metodo_pago_existe NUMBER; " +
                    "BEGIN " +
                    "    SELECT COUNT(*) INTO metodo_pago_existe "
+
                    "    FROM pago " +
                    "    WHERE ID_metodoPago = :NEW.ID_metodoPago;
" +
                    "    IF metodo_pago_existe = 0 THEN " +
                    "        RAISE_APPLICATION_ERROR(-20003,
'Error: El método de pago especificado no existe.'); " +
                    "    END IF; " +
                    "END;";

   // TRIGGER para verificar la cantidad de producto antes de
insertar o actualizar en la tabla 'tiene' Gabriel
   private static final String TRIG_VERIFICAR_CANTIDAD_PRODUCTO =
           "CREATE OR REPLACE TRIGGER verificar_cantidad_producto
" +
                    "BEFORE INSERT OR UPDATE ON tiene " +
                    "FOR EACH ROW " +
                    "DECLARE " +
                    "    cantidad_producto INTEGER; " +
                    "    producto_existe INTEGER; " +
                    "BEGIN " +
                    "    SELECT COUNT(*) " +
                    "    INTO producto_existe " +
                    "    FROM producto " +
                    "    WHERE ID_Producto = :NEW.ID_Producto; " +
                    "    IF producto_existe = 0 THEN " +
                    "        RAISE_APPLICATION_ERROR( " +
                    "                -20006, " +
```

```
                    "                'Error: El producto no existe
en la tabla Producto.' " +
                    "        ); " +
                    "    END IF; " +
                    "    SELECT Cantidad " +
                    "    INTO cantidad_producto " +
                    "    FROM producto " +
                    "    WHERE ID_Producto = :NEW.ID_Producto; " +
                    "    IF :NEW.Cantidad > cantidad_producto THEN
" +
                    "        RAISE_APPLICATION_ERROR( " +
                    "                -20005, " +
                    "                'Error: La cantidad a insertar
es mayor que la cantidad disponible del producto.' " +
                    "        ); " +
                    "    END IF; " +
                    "END;";

    // TRIGGER que evita que se inserte una relación duplicada
entre un carrito y un pedido en GESTIONCARRITO. Gabriel
    private static final String TRIG_EVITAR_CARRITO_DUPLICADO =
            "CREATE OR REPLACE TRIGGER evitar_carrito_duplicado " +
                    "BEFORE INSERT ON gestioncarrito " +
                    "FOR EACH ROW " +
                    "DECLARE " +
                    "    carrito_duplicado INTEGER; " +
                    "BEGIN " +
                    "    SELECT COUNT(*) INTO carrito_duplicado " +
                    "    FROM gestioncarrito " +
                    "    WHERE ID_Carrito = :NEW.ID_Carrito AND
ID_Pedido = :NEW.ID_Pedido; " +
                    "    IF carrito_duplicado > 0 THEN " +
                    "        RAISE_APPLICATION_ERROR(-20007,
'Error: Ya existe una relación entre este carrito y pedido.'); " +
                    "    END IF; " +
                    "END;";

    // Trigger para verificar que el ID_Usuario asociado con un
pedido existe en la tabla usuario
    private static final String TRIG_VERIFICAR_USUARIO_EN_PEDIDO =
            "CREATE OR REPLACE TRIGGER verificar_usuario_en_pedido
"
                    + "BEFORE INSERT OR UPDATE ON pedido "
                    + "FOR EACH ROW "
                    + "DECLARE "
                    + "    usuario_existe INTEGER; "
                    + "BEGIN "
                    + "    SELECT COUNT(*) INTO usuario_existe "
```

```
                    + "      FROM usuario "
                    + "     WHERE ID_Usuario = :NEW.ID_Usuario; "
                    + "    IF usuario_existe = 0 THEN "
                    + "       RAISE_APPLICATION_ERROR(-20007, "
                    + "           'Error: El usuario asociado al
pedido no existe.' "
                    + "       ); "
                    + "    END IF; "
                    + "END;";

    /**
     * Método que crea (o reemplaza) todos los disparadores en la BD.
     * Recibe una conexión abierta y ejecuta cada sentencia CREATE
TRIGGER.
     */
    public static void crearDisparadores(Connection conn) throws
SQLException {
        // Usa try-with-resources para asegurar que se cierra el
Statement
        try (Statement st = conn.createStatement()) {
            // Ejecutar cada disparador
            st.execute(TRIG_VERIFICA_PEDIDO_USUARIO);
            st.execute(TRIG_VALIDAR_USUARIO_MODIFICAPRODUCTO);
            st.execute(TRIG_VALIDAR_PRODUCTO_MODIFICAPRODUCTO);
            st.execute(TRIG_VERIFICAR_USUARIO_EXISTE);
            st.execute(TRIG_VERIFICAR_METODO_PAGO_EXISTE);
            st.execute(TRIG_VERIFICAR_CANTIDAD_PRODUCTO);
            st.execute(TRIG_VERIFICAR_USUARIO_EN_PEDIDO );
            st.execute(TRIG_EVITAR_CARRITO_DUPLICADO);

//st.execute(TRIG_VALIDAR_RELACION_PRODUCTO_MODIFICAPRODUCTO);
            System.out.println("Disparadores creados/reemplazados
con éxito.");
        }
    }

    public static void eliminarDisparadores() throws SQLException {
        java.sql.Connection con = practica.Connection.connection;
        Statement stmt = null;
        ResultSet rs = null;

        try {
            stmt = con.createStatement();

            // Obtener todos los nombres de los triggers
            String obtenerTriggersSQL = "SELECT TRIGGER_NAME FROM
USER_TRIGGERS";
            rs = stmt.executeQuery(obtenerTriggersSQL);
```

```
            // Eliminar cada trigger encontrado
            while (rs.next()) {
                String triggerName = rs.getString("TRIGGER_NAME");

                // Usar un nuevo Statement para ejecutar el DROP
TRIGGER
                try (Statement dropStmt = con.createStatement()) {
                    String dropTriggerSQL = "DROP TRIGGER " +
triggerName;
                    System.out.println("Eliminando trigger: " +
triggerName);
                    dropStmt.executeUpdate(dropTriggerSQL);
                } catch (SQLException e) {
                    System.err.println("Error al eliminar trigger "
+ triggerName + ": " + e.getMessage());
                }
            }
            JOptionPane.showMessageDialog(practica.Connection.frame,
"Triggers eliminados correctamente.");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            // Cerrar recursos
            if (rs != null) {
                try {
                    rs.close();
                } catch (SQLException ex) {
                    ex.printStackTrace();
                }
            }
            if (stmt != null) {
                try {
                    stmt.close();
                } catch (SQLException ex) {
                    ex.printStackTrace();
                }
            }
        }
    }


}
```

4. Breve motivación de la elección del software:

Hemos decidido usar Java ya que cuenta con una amplia variedad de bibliotecas para la interacción con bases de datos y para la construcción de interfaces gráficas, lo cual ha facilitado la implementación del programa.

También contábamos con el seminario 1 que estaba implementado en Java y hemos vuelto a reutilizar el SGBD de Oracle.

5. Pasos a seguir para formalizar el registro de una Base de Datos legalmente:

Requisitos y pasos para formalizar el registro de una base de datos en España

En España, el registro de una base de datos de carácter personal está regulado principalmente por el **Reglamento General de Protección de Datos (RGPD)** de la Unión Europea y la **Ley Orgánica 3/2018 de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD)**. A continuación, se detallan los pasos necesarios para formalizar legalmente el registro de una base de datos.

1. Identificar la naturaleza de la base de datos

- Determina si tu base de datos contiene **datos personales** (nombre, DNI, dirección, email, datos financieros, etc.).
- Si no contiene datos personales (por ejemplo, datos estadísticos o empresariales), no será necesario cumplir con las leyes anteriores.

Si la base de datos incluye datos personales:

1. Clasifica los datos según su **nivel de sensibilidad**:
    - **Datos básicos**: nombre, dirección, teléfono, email.
    - **Datos sensibles**: salud, orientación sexual, creencias religiosas, afiliaciones políticas.
2. Define la **finalidad** de la base de datos:
    - ¿Para qué se usarán los datos? (por ejemplo, marketing, gestión de clientes, RRHH).

2. Realizar un análisis de impacto (DPIA)

El **Análisis de Impacto Relativo a la Protección de Datos (DPIA)** es obligatorio si el tratamiento de datos implica riesgos significativos para los derechos y libertades de los usuarios (por ejemplo, sistemas de videovigilancia, uso de IA, tratamientos masivos de datos sensibles).

Este análisis debe incluir:

1. **Descripción de los tratamientos**: ¿Cómo se recogerán, almacenarán y procesarán los datos?
2. **Evaluación de riesgos**: Identificar riesgos para la privacidad y seguridad de los datos.
3. **Medidas correctoras**: Proponer soluciones para minimizar los riesgos detectados.

3. Inscripción del Responsable del Tratamiento

El **responsable del tratamiento** (persona o entidad que gestiona la base de datos, en nuestro caso, sería el grupo completo) debe formalizar su rol y cumplir con las obligaciones legales:

- Identificar la figura del **responsable del tratamiento** en la empresa.
- Formalizar contratos con los posibles **encargados del tratamiento**.

4. Elaborar y registrar la política de privacidad

La política de privacidad debe informar a los usuarios sobre:

1. La identidad del responsable.
2. Los datos que se recopilan y su finalidad.
3. Cómo se garantizan sus derechos: acceso, rectificación, supresión, oposición, limitación y portabilidad de los datos.
4. El plazo de conservación de los datos.

Debe registrarse y hacerse accesible públicamente (normalmente en la página web de la empresa o proyecto).

5. Inscripción en el Registro de Actividades de Tratamiento

El RGPD eliminó la necesidad de inscribir bases de datos en un organismo específico (como ocurría antes con la Agencia Española de Protección de Datos, AEPD). Ahora, es obligatorio llevar un **Registro Interno de Actividades de Tratamiento**, que debe incluir:

1. Identificación del responsable.
2. Finalidades del tratamiento.
3. Categorías de datos personales.
4. Destinatarios de los datos.
5. Plazos de conservación.
6. Transferencias internacionales (si aplican).
7. Medidas de seguridad implementadas.

Este registro debe estar disponible para inspecciones de la AEPD.

6. Implementar medidas de seguridad

Según el nivel de sensibilidad de los datos, se deben implementar las siguientes medidas de seguridad:

1. **Datos básicos**:
    - Cifrado de las bases de datos.
    - Gestión de permisos y accesos.
2. **Datos sensibles**:
    - Cifrado más robusto.
    - Pseudonimización o anonimización.
    - Sistemas de auditoría.

Estas medidas deben documentarse en la política de seguridad de la organización.

7. Redactar contratos de tratamiento de datos

Si trabajas con terceros que procesan los datos (por ejemplo, empresas de marketing o hosting), debes redactar un contrato que:

- Detalle las obligaciones del encargado del tratamiento.
- Prohíbe que los datos se utilicen para finalidades distintas a las establecidas.

8. Notificar brechas de seguridad

Cualquier brecha de seguridad que afecte a la base de datos debe notificarse:

- A la Agencia Española de Protección de Datos (AEPD) **en un plazo máximo de 72 horas**.
- A los usuarios afectados (si la brecha implica un riesgo para sus derechos).

10. Formalizar auditorías periódicas

La AEPD exige que se realicen auditorías periódicas para verificar el cumplimiento del RGPD y la LOPDGDD. Estas auditorías deben:

1. Evaluar la eficacia de las medidas de seguridad.
2. Revisar el registro de actividades de tratamiento.
3. Comprobar el cumplimiento de derechos de los usuarios.