

Facultad de Ingeniería

Ingeniería de Software I

**PROYECTO DE INGENIERÍA DE SOFTWARE: PRIMER INCREMENTO DEL SISTEMA DE TIENDA VIRTUAL PARA LA IMPRENTA Y LIBRERÍA CATÓLICA, LOS COBRES LIMITADO**

Autores:

Martín Alejandro Ibaceta Henríquez

Alejandro Iván Marco Hannus

Luis Alberto Núñez Acuña

Luis Nicolás Núñez Castro

Felipe Ignacio Rodríguez Proboste

Ignacio Alejandro Zañartu Höppner

Profesor:

Paulo Luis Francisco Quinsacara Jofré

Santiago, Chile

2025

Índice de Contenidos

1. Introducción 5

2. Propuesta de Solución 6

2.1 Porcentaje Parcial y Acumulado 6

3. Casos de Uso 7

4. Product Backlog List 19

5. Sprint 20

6. Vista Lógica 21

6.1 Diagrama de Clases 21

6.2 Modelo de Datos 22

7. Vista de Proceso 32

8. Vista de Desarrollo 33

9. Vista Física 34

10. Vista Externa 35

11. Casos de Prueba 36

12. Retrospectiva 37

12.1 Gráfico Burn-Up 37

12.2 Gráfico Burn-Down 37

**Índice de Figuras**

**Anexo A: Recursos 17**

Figura A.1: Organización del Trabajo, Primer Incremento 17

**Índice de Tablas**

**2. Propuesta de Solución 5**

[Tabla 2.1 Porcentaje Parcial y Acumulado, Primer Incremento Scrum++ 6](#_Toc200754439)

[Tabla 3.1 CU N°07, Incorporando Nuevo Cliente mediante Registro 7](#_Toc200754440)

# Introducción

El presente documento expone la implementación de la metodología ágil “Scrum++” como marco de trabajo para el desarrollo del sistema de ventas en línea de la tienda “Los Cobres Limitado”, empresa dedicada a la comercialización de artículos religiosos y actualmente en proceso de transformación digital. Este proyecto responde a la necesidad de modernizar sus canales de atención y transacción, con el fin de alinearse a las exigencias del mercado digital contemporáneo y ampliar su alcance hacia nuevos segmentos de clientes.

Como parte del primer incremento de desarrollo, se han incorporado prácticas de documentación y trazabilidad propias de enfoques ágiles, permitiendo capturar la evolución iterativa del producto, la definición de requisitos priorizados y la gestión colaborativa del equipo de trabajo.

# Propuesta de Solución

El desarrollo del proyecto se encuentra estructurado bajo la metodología ágil “Scrum++”, una extensión del marco Scrum que incorpora las prácticas complementarias orientadas al diseño arquitectónico, trazabilidad técnica y la validación progresiva del sistema.

Scrum++ promueve una metodología de entrega continua y orientada al valor, basándose en ciclos iterativos de corta duración que permiten integrar retroalimentación temprana, adaptación del alcance y mejora continua.

El **presente documento corresponde al primer incremento** del sistema, el cual contempla la implementación de una parte representativa del alcance total previsto. Este primer conjunto de entregables incluye funcionalidades esenciales, estructuradas según la arquitectura definida.

## Porcentaje Parcial y Acumulado

En función de la planificación establecida, se incorpora un porcentaje de avance parcial y acumulado correspondiente al primer incremento del proyecto, desarrollado conforme a los principios de la metodología Scrum++.

En la ***Tabla 2.1*** se señala la distribución de los casos de uso implementados durante la fase inicial, detallando su cantidad, porcentaje de representación con respecto al total de funcionalidades definidas y el nivel de avance acumulado. Esta información brinda una perspectiva precisa del estado actual del desarrollo, facilitando la evaluación del cumplimiento de los objetivos planteados para la primera iteración del proyecto.

Tabla 2.1: “Porcentaje Parcial y Acumulado, Primer Incremento Scrum++”.

|  |  |  |  |
| --- | --- | --- | --- |
| Incremento | Casos de Uso | Porcentaje % | Porcentaje Acumulado % |
| 1 | 10 | 20 | 20 |

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Casos de Uso

El apartado siguiente aborda los casos de uso (CU) correspondientes al 20 % de los requerimientos funcionales definidos para el primer incremento. La selección de estos casos responde a criterios de entrega temprana de valor y validación progresiva de los componentes esenciales del sistema.

A continuación, en la ***Tabla 3.1***, se describe el flujo de eventos y las condiciones asociadas al proceso de registro de un nuevo cliente en el sistema.

Tabla 3.1: “CU N°07, Incorporando Nuevo Cliente mediante Registro”.

|  |  |
| --- | --- |
| **Caso de Uso N°7** | Incorporando nuevo cliente mediante registro |
| Actores | Cliente |
| Propósito | Permitir a nuevos actores registrarse en el sistema para acceder a sus preferencias e información vinculada dentro del sistema. |
| Precondiciones | El actor no debe encontrarse previamente registrado en el sistema. |
| Resumen | El actor proporciona la información requerida para crear su cuenta. El sistema valida los datos e inicia la sesión automáticamente si el registro es exitoso. |
| Postcondiciones | El actor queda registrado y con una sesión activa en el sistema. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. El actor selecciona la opción  “Registrarse”. | 2. El sistema despliega un formulario de registro  solicitando: RUT, nombre, apellido, correo electrónico,  contraseña y confirmación de la contraseña. |
| 3. El actor completa todos los  campos requeridos y  selecciona la opción “Crear”. | 4. El sistema valida los datos ingresados y en caso de  ser correctos, almacena la información del actor  registrado y lo redirige al inicio con la sesión iniciada.  Por el contrario, si existen campos vacíos, datos  inválidos o la existencia del correo ingresado en el  sistema, se notifica al actor indicando los errores y  solicitando su corrección. |

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Tabla 3.2*** se detalla la secuencia de eventos asociada al inicio de sesión en el sistema, a través de la validación de las credenciales proporcionadas por el cliente.

Tabla 3.2: “CU N°08, Iniciando Sesión”.

|  |  |
| --- | --- |
| **Caso de Uso N°8** | Iniciando Sesión |
| Actores | Cliente |
| Propósito | Permitir a los actores registrados acceder a sus cuentas personales dentro del sistema. |
| Precondiciones | El actor debe encontrarse previamente registrado en el sistema. |
| Resumen | El actor proporciona sus credenciales de acceso. El sistema valida dicha información e inicia la sesión si los datos son correctos. |
| Postcondiciones | El actor accede al sistema con su sesión activa. |
| Tipos | Principal |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. El actor selecciona la opción  “Iniciar Sesión”. | 2. El sistema despliega un formulario solicitando el  correo electrónico y la contraseña. |
| 3. El actor ingresa ambos datos  y nuevamente selecciona la  opción “Iniciar Sesión”. | 4. El sistema valida las credenciales y en caso de  ser correctos, establece la sesión y redirige al actor  al panel de inicio. Por el contrario, si existen campos  vacíos o credenciales incorrectas, el sistema informa  los errores correspondientes y solicita su corrección  antes de proceder. |

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Tabla 3.3*** presenta el flujo de eventos y las condiciones vinculadas al proceso de actualización de las unidades del producto en inventario después de efectuarse una compra.

Tabla 3.3: “CU N°21, Actualizando unidades de producto en inventario posterior a la compra”.

A close-up of a document

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Tabla 3.4*** se describe detalladamente la secuencia de eventos correspondiente al proceso mediante el cual un actor registrado añade productos a su lista personalizada denominada “Favoritos”.

Tabla 3.4: “CU N°23, Incorporando artículos a lista de favoritos”.

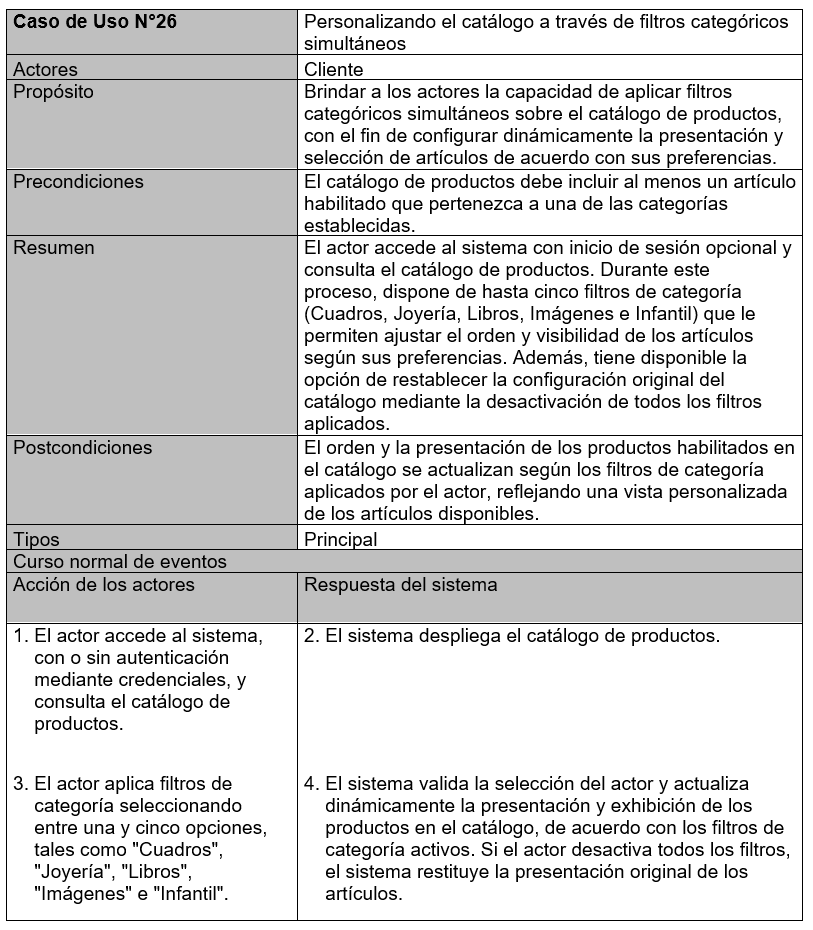
A screenshot of a document

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Tabla 3.5*** expone el flujo de eventos y las condiciones relacionadas con el proceso de configuración del catálogo de productos, a través de la funcionalidad que permite aplicar múltiples filtros por categoría.

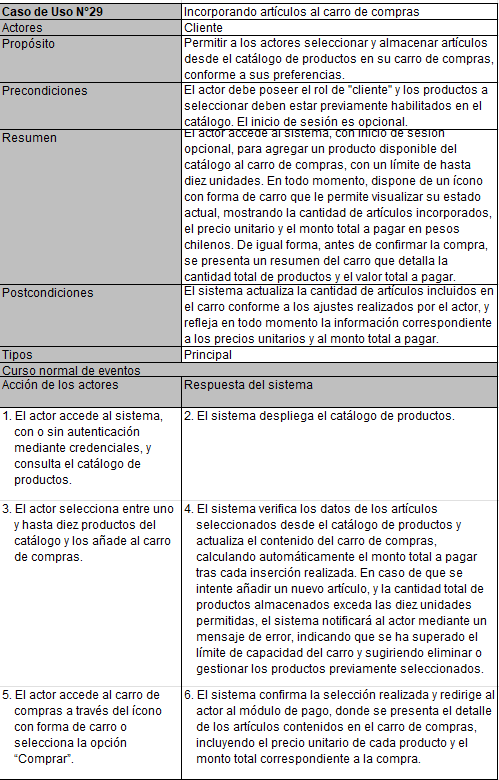
Tabla 3.5: “CU N°26, Personalizando el catálogo a través de filtros categóricos simultáneos”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Tabla 3.6*** se presenta la secuencia de eventos del proceso por el cual un actor añade artículos habilitados a su carro de compras desde el catálogo de productos.

Tabla 3.6: “CU N°29, Incorporando Artículos al Carro de Compras”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Tabla 3.7*** describe la secuencia de eventos y las condiciones asociadas al proceso de verificación de pedidos mediante el uso de un código de retiro, incluyendo las acciones del sistema, las validaciones correspondientes y los posibles resultados de la operación.

Tabla 3.7: “CU N°40, Verificando Pedidos Mediante Código de Retiro”.

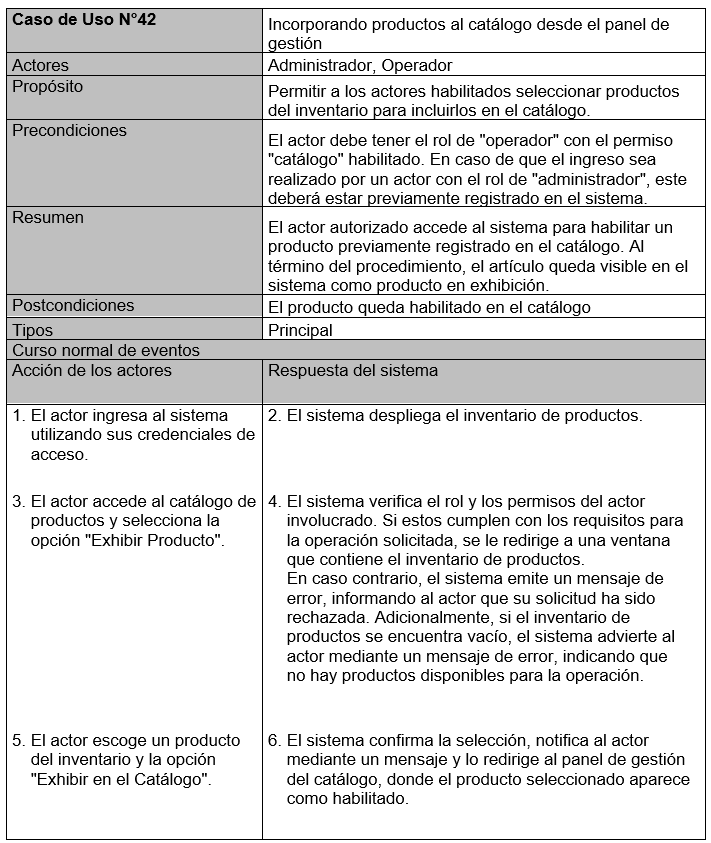
A close-up of a document

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Tabla 3.8*** presenta el flujo de eventos y las condiciones vinculadas al proceso de incorporación de artículos al catálogo desde una interfaz de gestión de productos.

Tabla 3.8: “CU N°42, Incorporando Productos al Catálogo desde el Panel de Gestión”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Tabla 3.9*** se presenta la secuencia de eventos del proceso por el cual se gestionan los permisos de usuario.

Tabla 3.9: “CU N°45, Gestionando Permisos de Usuario”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Tabla 3.10*** se detalla la secuencia de eventos asociada a la eliminación de productos del catálogo con confirmación.

Tabla 3.10: CU N°43, Eliminando productos del catálogo con confirmación”.

A close-up of a document

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Product Backlog List

La planificación de actividades y tareas ha sido llevada a cabo por el grupo de trabajo identificado como Equipo 23*,* cuya misión es diseñar, construir y validar una plataforma funcional que satisfaga los requerimientos operativos, técnicos y estratégicos de “Los Cobres Limitado”, siguiendo un enfoque iterativo e incremental, centrado en el valor entregado al cliente.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white and black text

AI-generated content may be incorrect.

# Sprint

Para efectos del desarrollo del proyecto en su fase inicial de desarrollo y en consideración de la metodología “Scrum++” implementado, se adopta el modelo arquitectónico de vistas “4+1”, a través del cual se describe la solución propuesta desde distintas perspectivas: vista lógica, vista de procesos, vista de desarrollo y vista física. Además, se incorporan los de casos de uso definidos previamente y seleccionados para el primer incremento, asegurando una comprensión integral y coherente de la arquitectura del sistema.

# Vista Lógica

En la presente sección se describen los diagramas de clases correspondiente tanto al esquema lógico como al modelo de carácter funcional, los cuales representan las especificaciones y el diseño estructural de la base de datos seleccionada. A continuación, se detallan las entidades involucradas, su estructura interna (atributos y propiedades) y las relaciones que mantienen entre sí, con el fin de comprender su comportamiento e interacción dentro del sistema.

## Diagrama de Clases

El diagrama de clases representa la arquitectura lógica de una tienda en línea, modelando las entidades principales involucradas en la administración de usuarios, productos y el flujo completo de compras y pedidos. Además de atributos y relaciones, incorpora métodos específicos para reflejar acciones clave dentro del sistema.

**Usuario:**

Entidad base que representa a cualquier persona que interactúe con el sistema. Contiene información personal y de acceso como:

* Nombre, Apellido, Email, Password\_Hash, RUT, Teléfono.
* Fechas de registro y última conexión (Fecha\_Registro, Última\_Conexión).
* Atributo booleano Activo para gestionar la vigencia de la cuenta.

No contiene métodos, ya que actúa como entidad base para Cliente y Operador.

**Cliente:**

Representa a los usuarios que compran productos en la tienda, heredados de la entidad “Usuario”.

**Métodos:**

* registrarCuenta(): Permite a un usuario crear una cuenta como cliente.
* registrarCompra(): Registra una nueva compra, asociando un pedido al cliente.
* añadirCarro(): Crea o modifica el carrito de compras del cliente.

**Gestión operativa**, **Operador:**

Subtipo de Usuario que tiene permisos especiales dentro del sistema, encargándose de la administración de pedidos.

**Atributos:** Cargo, ya sea el rol asignado al usuario o nivel del operador.

**Métodos:**

* registrarPedido(): Ingresa un nuevo pedido al sistema.
* eliminarPedido(): Elimina un pedido existente.
* modificarPedido(): Permite editar la información de un pedido.
* verificarPedido(): Marca un pedido como revisado y conforme.
* entregarPedido(): Asocia la entrega del pedido al operador correspondiente.

Este modelo permite trazar quién intervino en cada etapa del pedido, con referencias a “Id\_Operador”, “Id\_Op\_Registro”, “Id\_Op\_Entrega”.

**Gestión de productos**, **Producto:**

Entidad que almacena los productos ofrecidos por la tienda.

**Atributos:** “Nombre\_Producto”, “Descripción”, “Precio\_Unitario”, “Stock\_Actual”, “SKU” y “Activo”.

No presenta métodos declarados, lo cual sugiere que su gestión es responsabilidad de la capa de datos del sistema o usuario con rol “operador” a través de interfaces externas.

**Imagen\_Producto:**

Almacena múltiples imágenes asociadas a un producto.

**Atributos:** “URL\_Imagen”, Orden.

**Categoría:**

Permite organizar los productos jerárquicamente mediante el atributo “Id\_Padre”.

**Atributos:** “Nombre\_Categoría”, “Descripción\_Categoría”.

**Gestión de carritos y Favoritos**, **Carrito\_Compra:**

Asocia productos seleccionados de forma temporal a un cliente.

**Atributos:** “Fecha\_Creación”, “Fecha\_Ultima\_Actualización”.

**Item\_Carrito:**

Relaciona un producto con un carrito específico.

**Atributos:** “Cantidad”, “Precio\_Unitario\_Momento”.

**Lista\_Favoritos:**

Permite a los clientes guardar productos de interés.

**Atributos:** “Nombre\_Lista”, “Fecha\_Creación”, “Fecha\_Ultima\_Actualización”.

**Item\_Favorito:** Elemento individual de una lista de favoritos.

**Atributos:** “Fecha\_Agregado”, “Comentarios”.

Estas entidades no poseen métodos, ya que se asume que sus operaciones (agregar/quitar elementos) se gestionan desde la lógica de aplicación.

**Gestión de Pedidos**, **Pedido:** Entidad principal del flujo de compra.

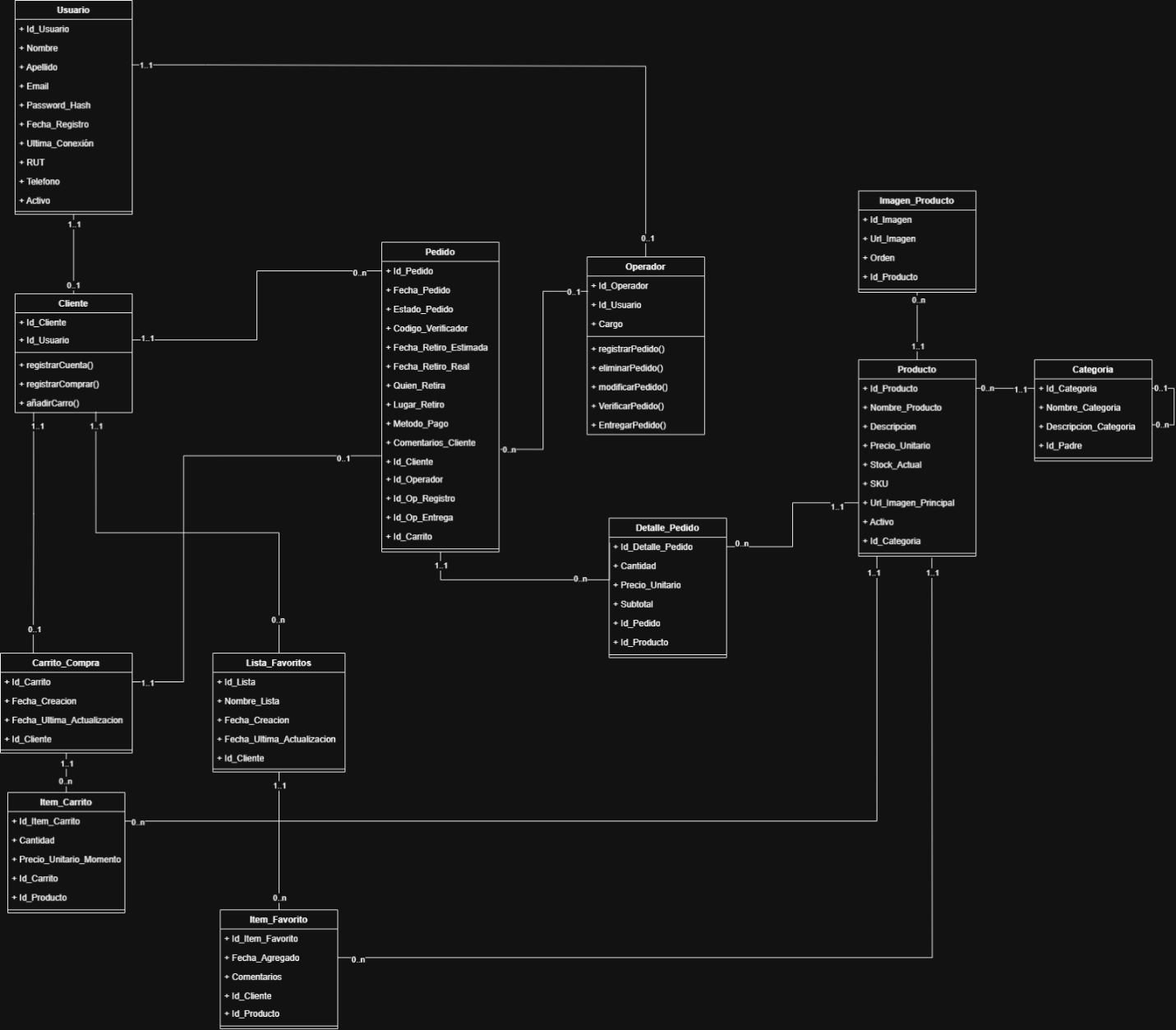
**Atributos:** Fechas estimadas y reales de retiro, lugar de retiro, método de pago, código verificador, estado, comentarios del cliente.

No tiene métodos directamente definidos, pero su manipulación está delegada a los operadores.

**Detalle\_Pedido:** Relaciona un producto con un pedido.

**Atributos:** Cantidad, Precio\_Unitario, Subtotal.

Figura 6.1: “Diagrama de clases”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

## Modelo de Datos

Este modelo de base de datos está diseñado para gestionar de manera eficiente las operaciones de un sistema de comercio electrónico, incluyendo la gestión de usuarios, productos, pedidos, carritos de compra y listas de favoritos.

**Componentes Clave del Modelo:**

**Gestión de Usuarios (Superclase y Subclases):**

**USUARIO (superclase\*)**: Es la tabla central que almacena la información básica de todos los usuarios del sistema (nombre, email, contraseña, etc.).

**OPERADOR (subclase\*\*)**: Representa a los empleados o administradores del sistema. Son un tipo especial de USUARIO y pueden gestionar pedidos.

**CLIENTE (subclase)**: Representa a los usuarios que realizan compras. Son también un tipo especial de USUARIO y están directamente relacionados con los carritos de compra, las listas de favoritos y los pedidos. Esta estructura permite una gestión unificada y flexible de distintos tipos de usuarios.

\*Superclase: contiene toda la información que define a los usuarios

\*\*subclase: heredan la información de una superclase y aparte cuantan con sus propias características.

**Catálogo de Productos:**

**CATEGORIA**: Organiza los productos en diferentes grupos (por ejemplo, "Electrónica", "Ropa", "Libros"), permitiendo una navegación y búsqueda eficientes. Permite subcategorías para una clasificación más detallada.

**PRODUCTO**: Almacena los detalles de cada artículo que se vende (nombre, descripción, precio, stock, etc.). Cada producto pertenece a una categoría.

**IMAGEN\_PRODUCTO**: Permite asociar múltiples imágenes a un producto, mejorando su presentación visual en la tienda.

**Proceso de Compra:**

**CARRITO\_COMPRA**: Registra los productos que un CLIENTE ha seleccionado antes de finalizar una compra. Es un paso intermedio en el proceso de pedido.

**ITEM\_CARRITO**: Detalla cada producto dentro de un carrito de compra, incluyendo la cantidad y el precio en el momento de la adición.

**PEDIDO**: Representa una compra finalizada por un CLIENTE. Un pedido puede originarse a partir de un CARRITO\_COMPRA y es gestionado por uno o más OPERADORes (para registro y entrega).

**DETALLE\_PEDIDO**: Contiene la información de los productos individuales que forman parte de un PEDIDO, incluyendo la cantidad y el precio final de cada artículo.

**Funcionalidades Adicionales:**

**LISTA\_FAVORITOS**: Permite a los Clientes guardar productos de su interés para futuras referencias o compras.

**ITEM\_FAVORITO**: Relaciona productos específicos con las listas de favoritos de los clientes.

A continuación, se describen las principales conexiones en nuestro modelo:

**Usuarios y sus Roles:**

**Un Usuario puede ser un Operador o un Cliente:** Partimos de una tabla general de USUARIO (donde guardamos su email, contraseña, etc.). Luego, se "extiende" a roles específicos: un OPERADOR (quien gestiona el sistema o pedidos) y un CLIENTE (quien realiza compras). Esto significa que un operador o un cliente SIEMPRE será primero un usuario.

**Organización de Productos:**

**Categorías organizan Productos:** Cada PRODUCTO (como un televisor o una camisa) siempre pertenece a una CATEGORIA (por ejemplo, "Electrónica" o "Ropa"). Una categoría, a su vez, puede tener "subcategorías" para una mejor organización (ej: "Televisores" dentro de "Electrónica").

**Un Producto tiene Imágenes:** Un PRODUCTO puede mostrar varias IMAGEN\_PRODUCTO para que el cliente lo vea desde diferentes ángulos.

**El Proceso de Compra (Carrito y Pedido):**

**Un Cliente tiene un Carrito de Compra:** Un CLIENTE puede tener uno o varios CARRITO\_COMPRA, donde va añadiendo los artículos que le interesan.

**Un Carrito contiene Ítems:** Cada CARRITO\_COMPRA está compuesto por varios ITEM\_CARRITO, que son los productos específicos (con sus cantidades) que el cliente desea comprar. Cada ITEM\_CARRITO se refiere a un PRODUCTO en particular.

**Un Cliente realiza Pedidos:** Cuando un CLIENTE decide comprar, crea un PEDIDO.

**Un Pedido viene de un Carrito:** Un PEDIDO generalmente se genera a partir de un CARRITO\_COMPRA que el cliente ha decidido finalizar.

**Un Pedido contiene Detalles:** Cada PEDIDO se desglosa en DETALLE\_PEDIDO, que lista todos los productos comprados, sus cantidades y el precio final al momento de la compra. Cada DETALLE\_PEDIDO se refiere a un PRODUCTO específico.

**Operadores gestionan Pedidos:** Los Operadores son los encargados de registrar y, en algunos casos, entregar los Pedidos, lo que significa que un pedido está vinculado a los operadores que lo manejaron.

**Listas de Favoritos:**

**Un Cliente tiene Listas de Favoritos:** Los Clientes pueden crear LISTA\_FAVORITOS para guardar productos que les interesan para el futuro.

**Las Listas de Favoritos contienen Ítems:** Cada LISTA\_FAVORITOS incluye varios ITEM\_FAVORITO, que son los Productos específicos que el cliente ha marcado como favoritos.

Vista de Procesos

El siguiente apartado describe los diagramas de proceso, en los cuales se detallan las funciones del comportamiento dinámico del sistema, permitiendo visualizar la interacción entre actores y componentes durante la ejecución de los procesos funcionales dentro del sistema a modo de sustento para la validación de los requisitos definidos.

En la ***Figura 7.1***, se puede visualizar el proceso de registro de un nuevo cliente en el sistema, el cual refiere al usuario al panel de inicio mediante la confirmación de la creación de su cuenta de uso personal, distribuyendo su secuencia de operación entre los diferentes controladores y tablas vinculadas al modelo de base de datos incorporando, conforme a las especificaciones de los modelos pertenecientes a la vista lógica. En el proceso se puede apreciar una dependencia hacia los controladores de “Cliente” y “PostgreSQL”, los cuales operan de manera individual y representan un diseño de arquitectura modular. Particularmente, el controlador de clientes valida los datos de ingreso y administra a los clientes como entidades del negocio, no necesariamente como usuarios autenticados. Por otro lado, el controlador de base de datos se encarga de abstraer y centralizar todas las operaciones de lectura/escritura, tales como: consulta, inserción, actualización y eliminación, validando la lógica transaccional. Finalmente, se encuentra el controlador de vistas, el cual se encarga de gestionar el flujo entre las ventanas del sistema a modo de dirigir acceso a clientes respecto a la solicitud de diferentes funciones y para efectos del proceso de creación de cuenta, redirigiendo acceso hacia el panel de inicio ante la confirmación y validación del proceso de registro.

Figura 6.2: “Modelo de Base de Datos”.

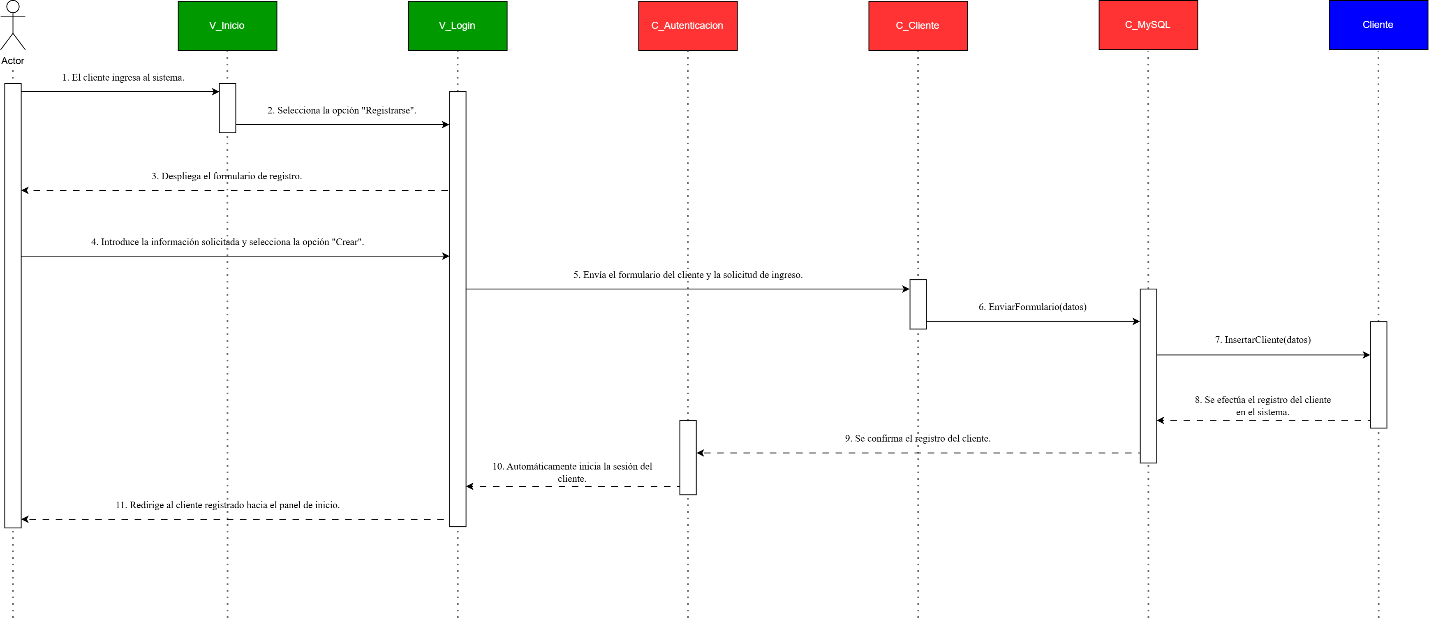
  
Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Vista de Proceso

Se describen a continuación los procesos funcionales del sistema derivados del total de casos de uso (CU) definidos en el primer incremento. Los diagramas incluidos ilustran la secuencia de operaciones entre los componentes clave del sistema, conforme a la arquitectura establecida. Se consideran las ventanas de interfaz de usuario, visualizadas en color verde, que reflejan las vistas accesibles durante la interacción; los controladores funcionales, en color rojo, diseñados bajo un enfoque modular que asigna responsabilidades específicas a cada segmento del sistema; y las entidades o tablas de base de datos, en color azul, seleccionadas en coherencia con un modelo transaccional que respalda la ejecución operativa y permite el almacenamiento de productos, usuarios y otros elementos relevantes para la gestión y administración del sistema.

La ***Figura 7.1*** ilustra el proceso de registro de un nuevo cliente en el sistema, el cual concluye con la derivación del actor hacia el panel de inicio una vez confirmada la creación de su cuenta de uso personal. La secuencia de operaciones se distribuye entre distintos controladores y entidades asociadas al modelo de base de datos, de acuerdo con las especificaciones definidas en los modelos correspondientes a la vista lógica. Durante este proceso, el controlador de cliente se encarga de validar los datos ingresados y gestionar a los clientes como entidades del negocio, sin requerir necesariamente su autenticación como usuarios registrados. En cuanto al controlador de base de datos, este centraliza todas las operaciones de lectura y escritura, como consultas, inserciones, actualizaciones y eliminaciones, asegurando el cumplimiento de la lógica transaccional definida para el sistema; en particular, dicho controlador se encarga de insertar en la base de datos la información del cliente, incorporando los datos como nuevos registros.

Figura 7.1: “CU N°07, Incorporando Nuevo Cliente mediante Registro”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Figura 7.2***, se observa la situación en la cual se despliega un mensaje de error al ingresar datos incorrectos para efectos del proceso de ingreso de un nuevo cliente.

Figura 7.2: “Error en la Incorporación del Cliente por Validación Fallida de Datos”.

A diagram of a diagram

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

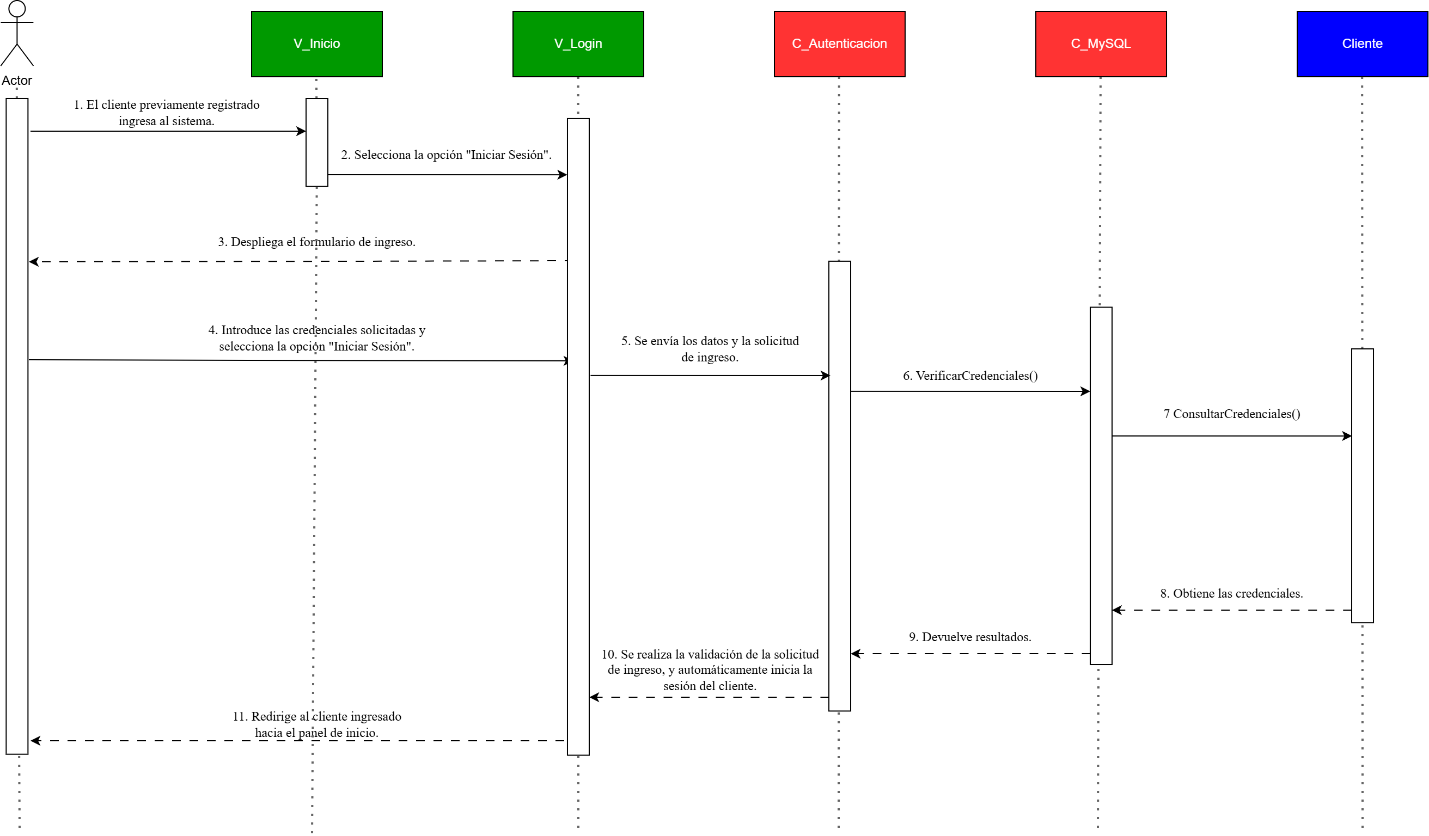
Figura 7.3: “Rechazo de Incorporación del Cliente por Correo Electrónico Previamente Registrado”.

A diagram of a company

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

Figura 7.4: “CU N°08, Iniciando Sesión”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

Figura 7.5: “Ingresando Credenciales de Acceso Invalidas”.

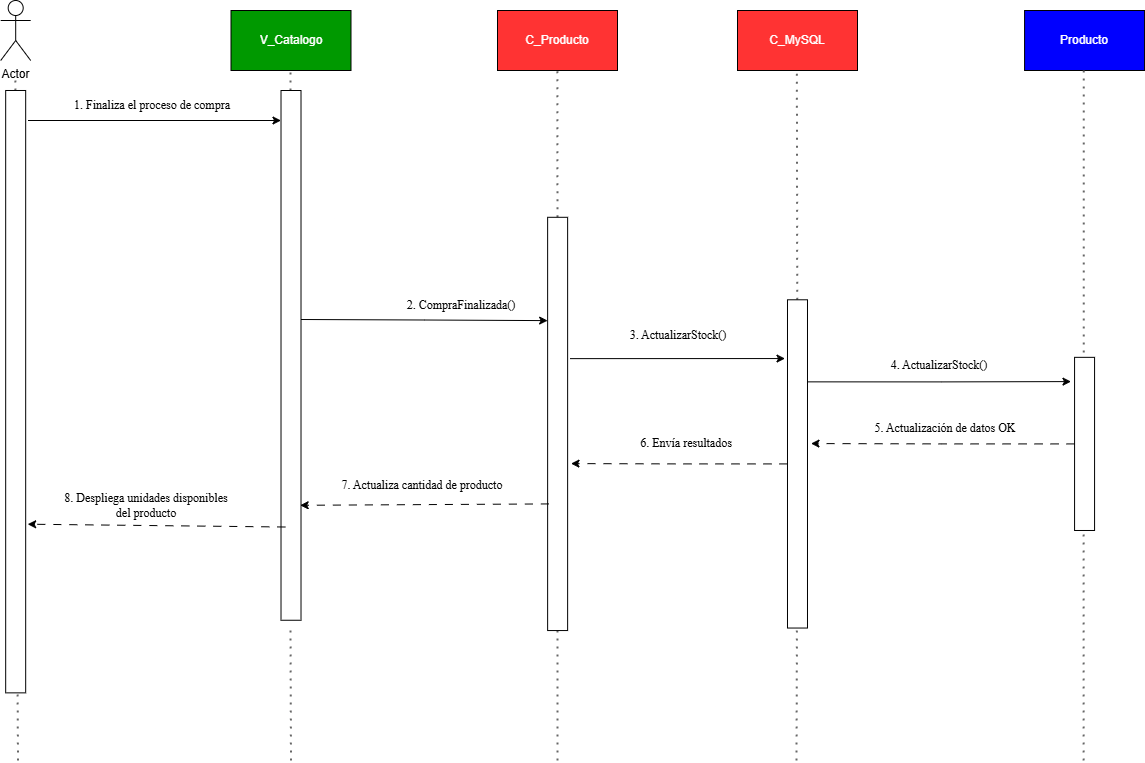
A diagram of a project

AI-generated content may be incorrect.

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.6*** ilustra el proceso que ejecuta el sistema al finalizar una compra, específicamente cuando se descuenta correctamente la cantidad de producto del stock y aun permaneciendo unidades disponibles. Una vez completada la transacción, el sistema invoca al controlador de producto y al controlador de base de datos para actualizar el valor del campo “Stock\_actual” en la tabla “producto”. Tras esta operación, los nuevos datos de la cantidad son devueltos, reorganizados por el controlador de producto y posteriormente reflejados en el catálogo actualizado.

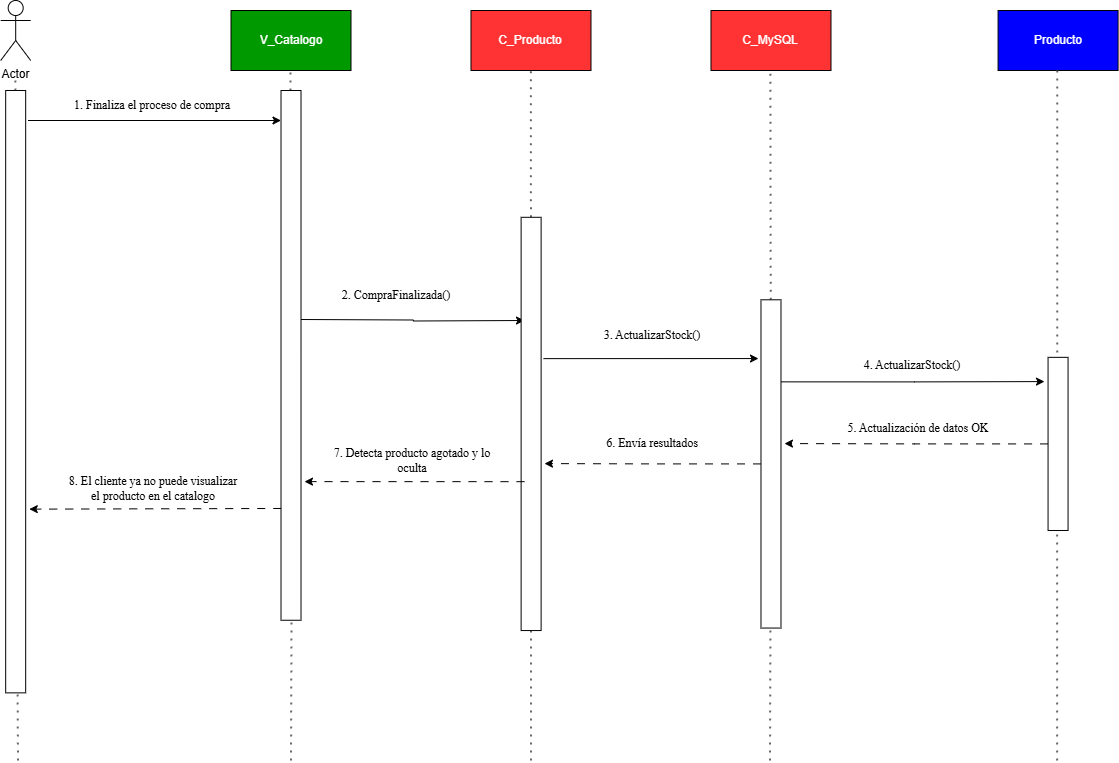
Figura 7.6: “CU N°21, Actualizando Cantidad de Producto tras Compra”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.7*** muestra el proceso de actualización de la cantidad de un producto tras completarse una compra. En este caso, la unidad vendida corresponde a la última disponible, por lo que el producto se agota. El sistema actualiza el valor correspondiente en la base de datos mediante el controlador de producto y el controlador de base de datos. Una vez realizada esta actualización, si el controlador de producto detecta que la cantidad disponible es igual a cero, el producto es ocultado automáticamente del catálogo, evitando que se muestre como disponible en futuras consultas.

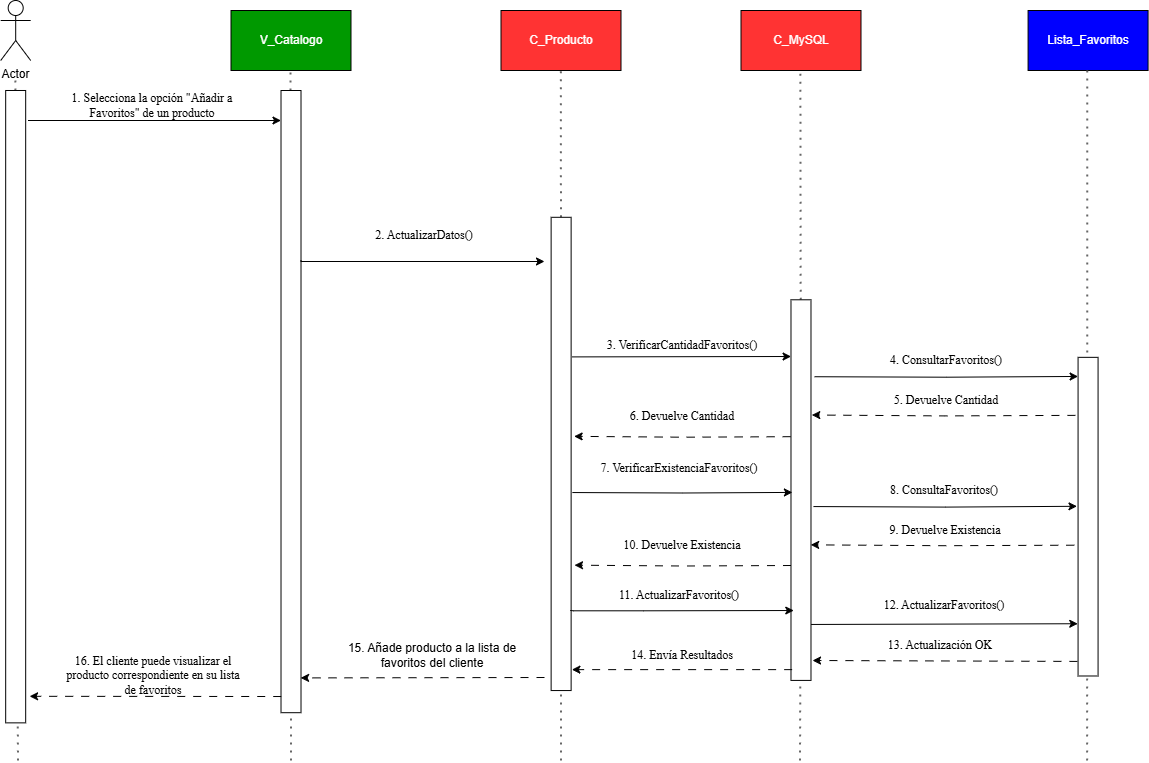
Figura 7.7: “CU N°21, Actualizando Cantidad de Producto tras Compra y Producto Agotado”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Figura 7.8*** se observa que, al seleccionar la opción “Añadir a favoritos” en un producto, el sistema ejecuta el método correspondiente para actualizar los datos. Antes de incorporar el artículo a la lista de favoritos, se verifica que la cantidad total de productos almacenados por el usuario no supere el máximo permitido de treinta y que el producto seleccionado no haya sido añadido previamente.

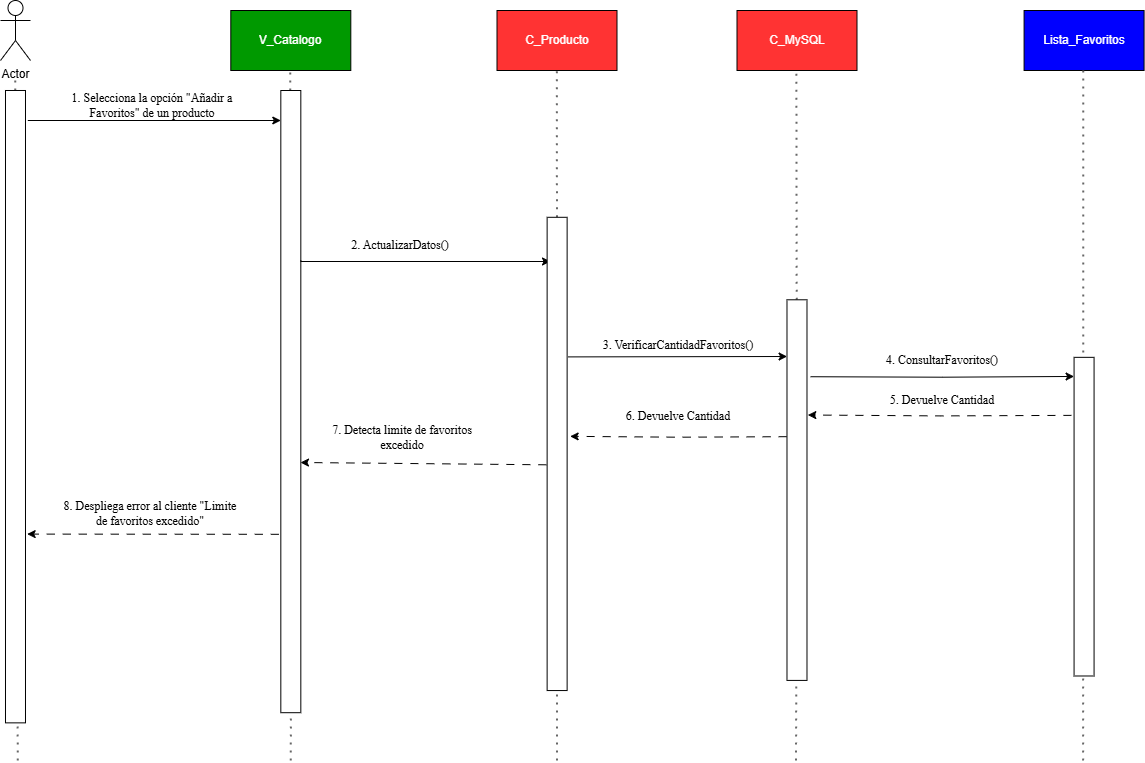
Figura 7.8: CU N°23, Agregando artículos a la lista de favoritos”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

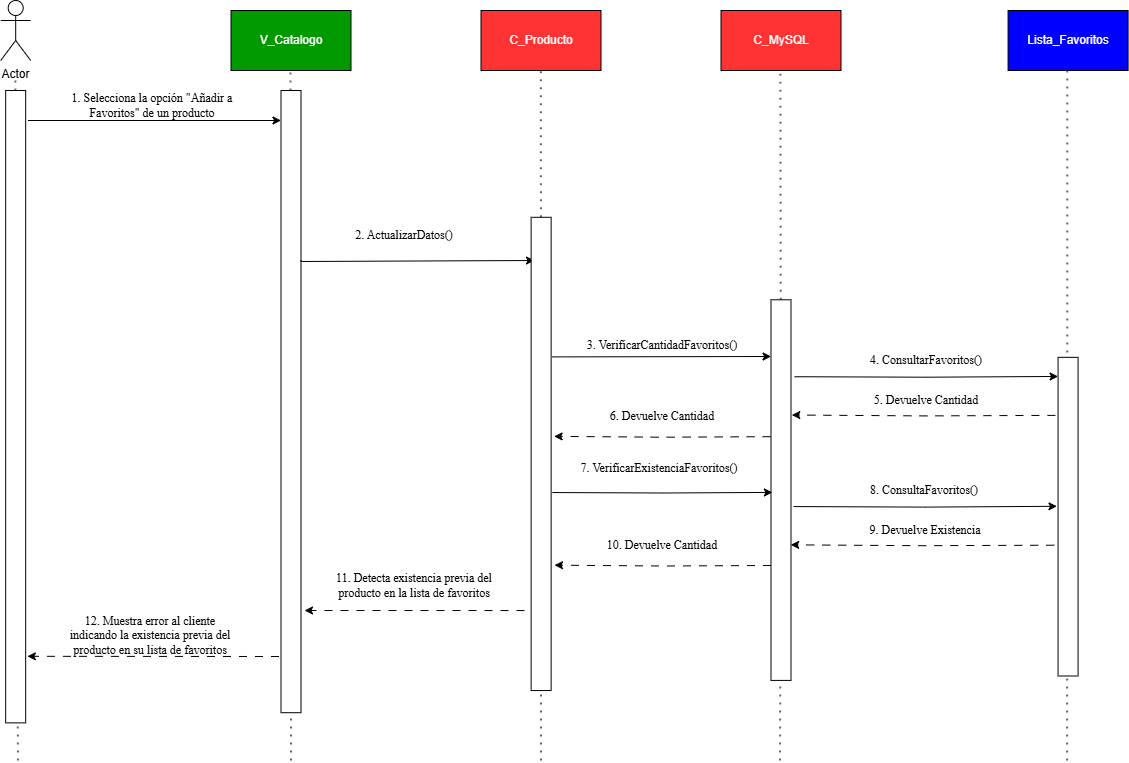
La ***Figura 7.9*** representa un escenario alternativo, en el cual el usuario ha excedido el límite permitido de productos en su lista de favoritos. Al detectar esta condición a través del controlador de producto, el sistema instruye a la vista del catálogo para que despliegue el mensaje de error correspondiente, informando al usuario de dicha restricción.

Figura 7.9: “Agregando artículos a la lista de favoritos, límite de favoritos excedido”.

  
Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.10*** muestra un escenario adicional relacionado con la incorporación de productos a la lista de favoritos de un cliente. En este caso, al verificarse que el producto ya se encuentra previamente registrado en dicha lista, el sistema detecta la duplicación y despliega el mensaje de error correspondiente para informar al cliente.

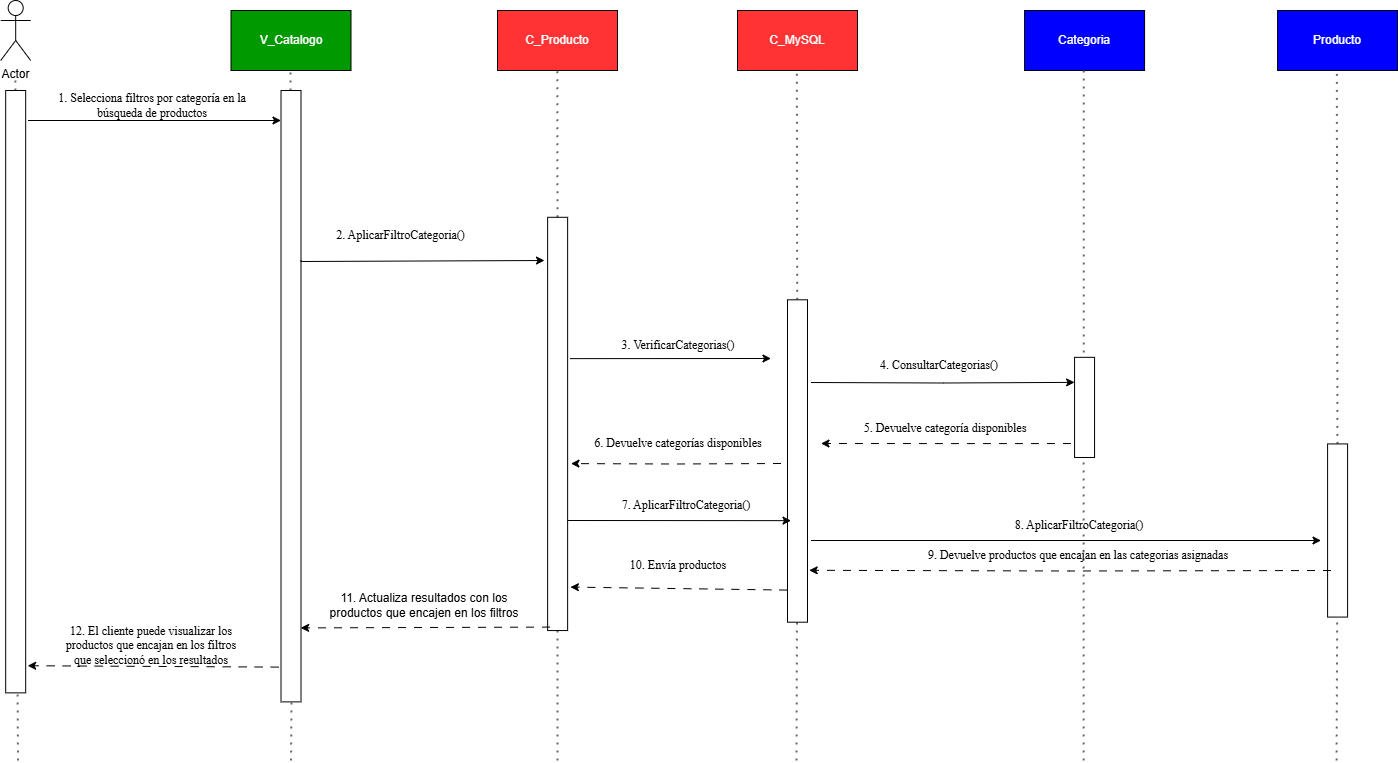
Figura 7.10: “Agregando artículos a la lista de favoritos, producto duplicado”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.11*** representa el proceso de filtrado de productos por múltiples categorías durante una búsqueda. Al iniciarse esta acción, el sistema invoca el método correspondiente en el controlador de producto, el cual verifica las categorías disponibles para aplicar como filtros. Esta información se obtiene mediante una consulta al controlador de base de datos, que se comunica con la tabla “Categoría”. Una vez confirmada la validez de la cantidad de filtros seleccionados, el sistema aplica el filtrado a través del controlador de base de datos, recuperando los productos que cumplen con los criterios definidos, los cuales son posteriormente desplegados en los resultados de búsqueda.

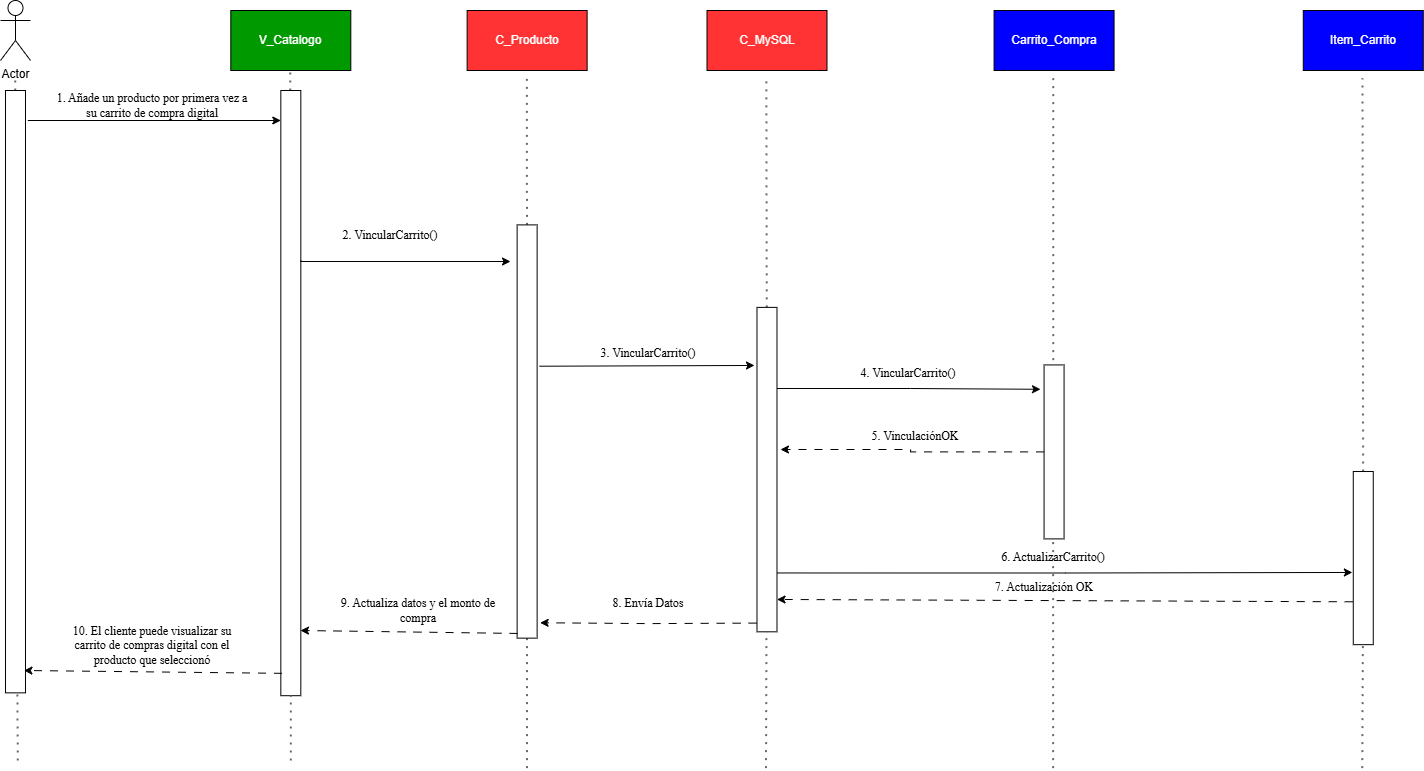
Figura 7.11: “CU N°26, Filtrando catálogo por categorías simultaneas”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La Figura ***7.12*** muestra el escenario en que un usuario añade un producto por primera vez a su carrito de compras digital. En este contexto, el sistema debe asociar el carrito al usuario, lo cual se realiza mediante una consulta a la tabla “Carrito\_compra” de la base de datos. Esta vinculación implica registrar la fecha de creación del carrito junto con su ID y la del cliente. Dado que se trata de la primera vez que se utiliza, el campo “fecha\_ultima\_actualizacion” conserva la misma fecha. A continuación, el sistema solicita la actualización del carrito a través del controlador de producto, que verifica que no se haya superado el límite de diez productos antes de enviar la información al controlador de la base de datos. Este, a su vez, actualiza la tabla “Item\_Carrito” y retorna los datos al controlador de producto. Finalmente, se recalcula el monto total de la compra, el cual se envía a la vista de catálogo para que el usuario pueda visualizar su carrito actualizado con el nuevo valor.

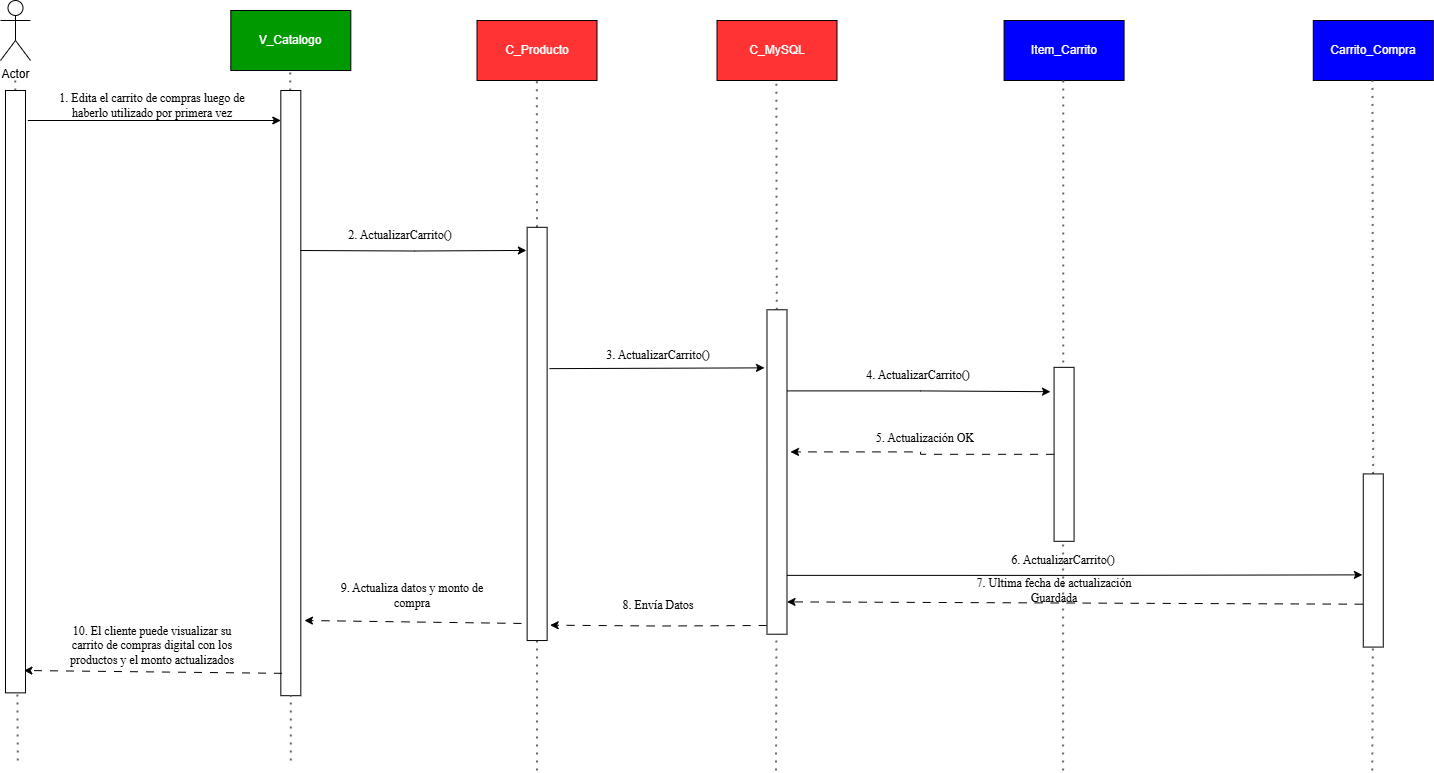
Figura 7.12: “CU N°29, Utilizando el carro de compras digital”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.13*** muestra el escenario en que el cliente vuelve a utilizar su carrito de compras digital tras haberlo usado previamente. En este caso, el sistema actualiza el contenido del carrito con los datos del nuevo producto, previa verificación de que no se ha superado el límite máximo de diez artículos. A continuación, se modifica el campo “fecha\_ultima\_actualizacion” con la fecha actual. Finalmente, el sistema recalcula el monto total de la compra y lo presenta al usuario en la vista correspondiente.

Figura 7.13: “Utilizando el carro de compras digital luego de haberlo utilizado por primera vez”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.14*** muestra el escenario en que un cliente intenta añadir un producto adicional a su carrito de compras digital, habiendo superado previamente el límite máximo de diez artículos. Durante el proceso, el controlador de producto realiza la verificación correspondiente y, al detectar que el límite ha sido excedido, el sistema genera un mensaje informativo destinado al cliente, indicándole que ha alcanzado el número máximo de productos permitidos y bloqueando la posibilidad de continuar añadiendo nuevos elementos al carrito.

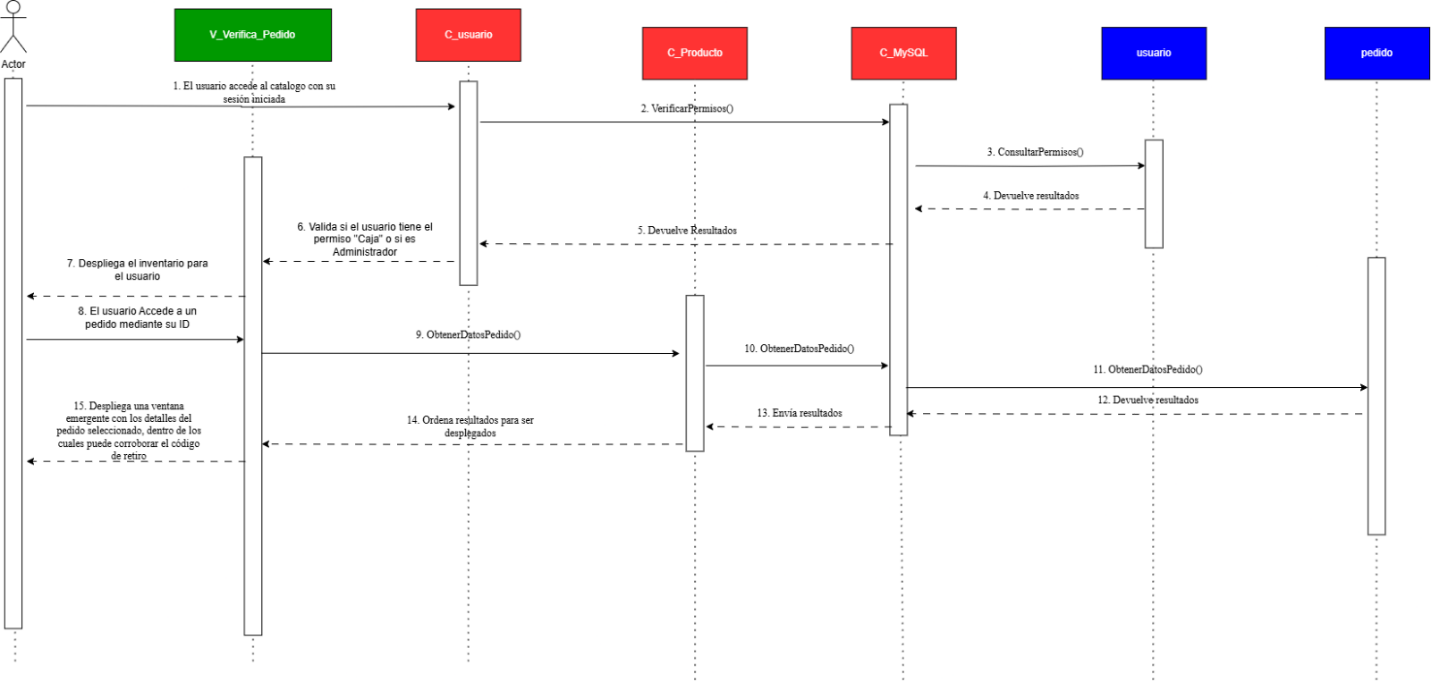
Figura 7.14: “Utilizando el carro de compras digital luego de haberlo utilizado por primera vez, límite de productos excedido”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.15*** muestra el proceso mediante el cual un usuario accede al inventario de productos. Al hacerlo, el sistema verifica que cuente con los permisos necesarios, ya sea el permiso de “Caja” o el rol de “Administrador”, lo cual se valida consultando la tabla “usuario” de la base de datos. Si se cumple alguna de estas condiciones, el sistema despliega la vista “Verificar pedidos” para dicho usuario. A partir de esta interfaz, el usuario puede acceder a un pedido ingresando su ID, lo que activa una consulta a la tabla “pedido” de la base de datos. Los datos obtenidos son enviados al controlador de producto, el cual los organiza y presenta en una ventana emergente. En esta vista, el usuario puede visualizar los detalles del pedido y confirmar su entrega mediante la verificación del código de retiro asociado.

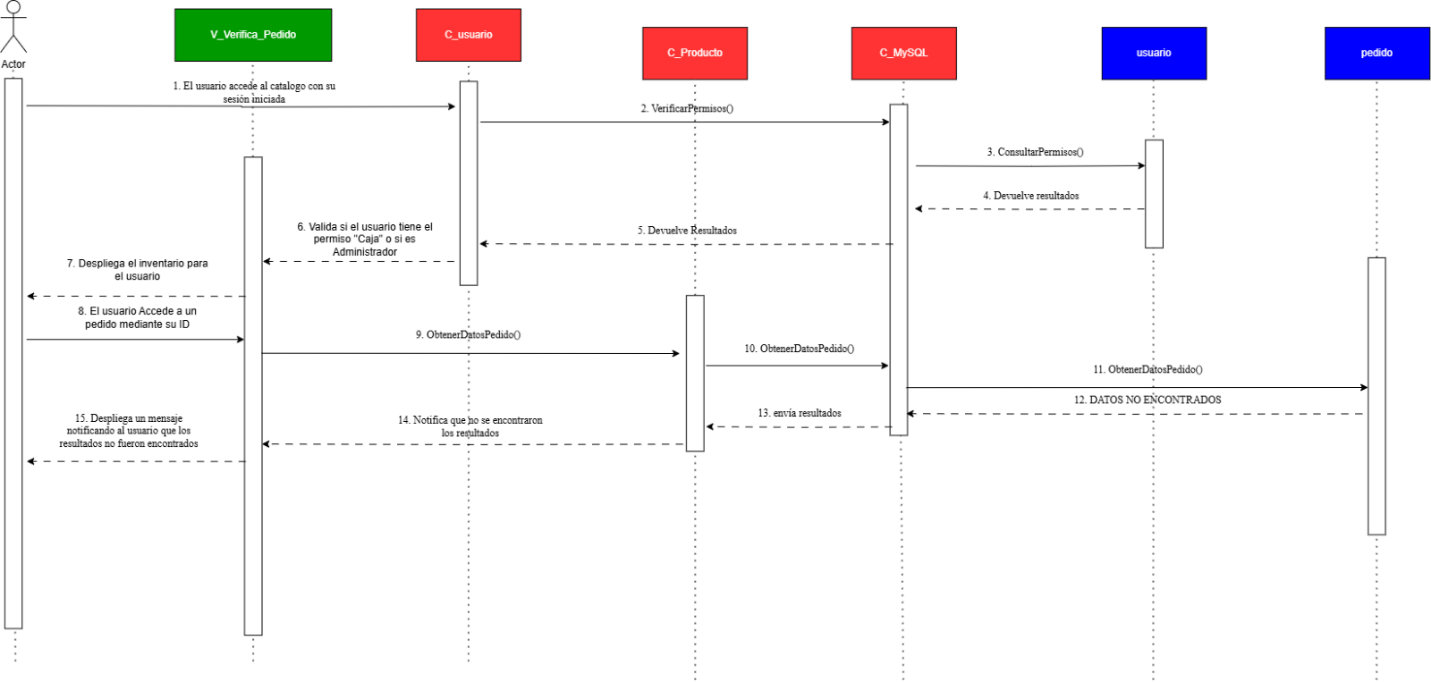
Figura 7.15: “CU N°40, Verificando pedidos mediante código de retiro”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.16*** representa el escenario en que, tras superar la verificación de permisos e ingresar un ID de pedido, el sistema no logra encontrar la información correspondiente. Como resultado, se notifica al usuario mediante un mensaje indicando que no se han encontrado datos asociados al identificador ingresado.

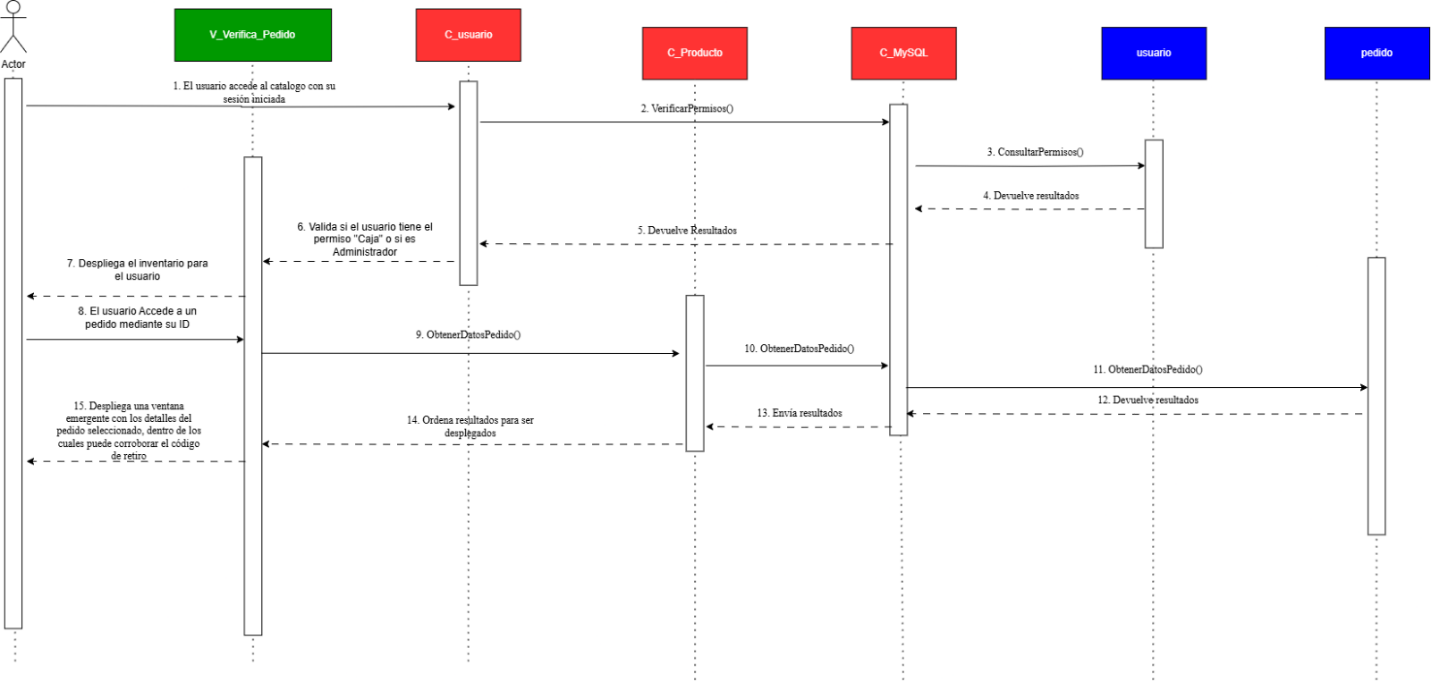
Figura 7.16: “Verificando pedidos mediante código de retiro, datos no encontrados”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.17*** muestra el escenario en que un usuario intenta acceder a la vista “Verificar pedidos” sin contar con los permisos o roles requeridos. Ante esta situación, el sistema identifica la falta de autorización y despliega el mensaje de error correspondiente.

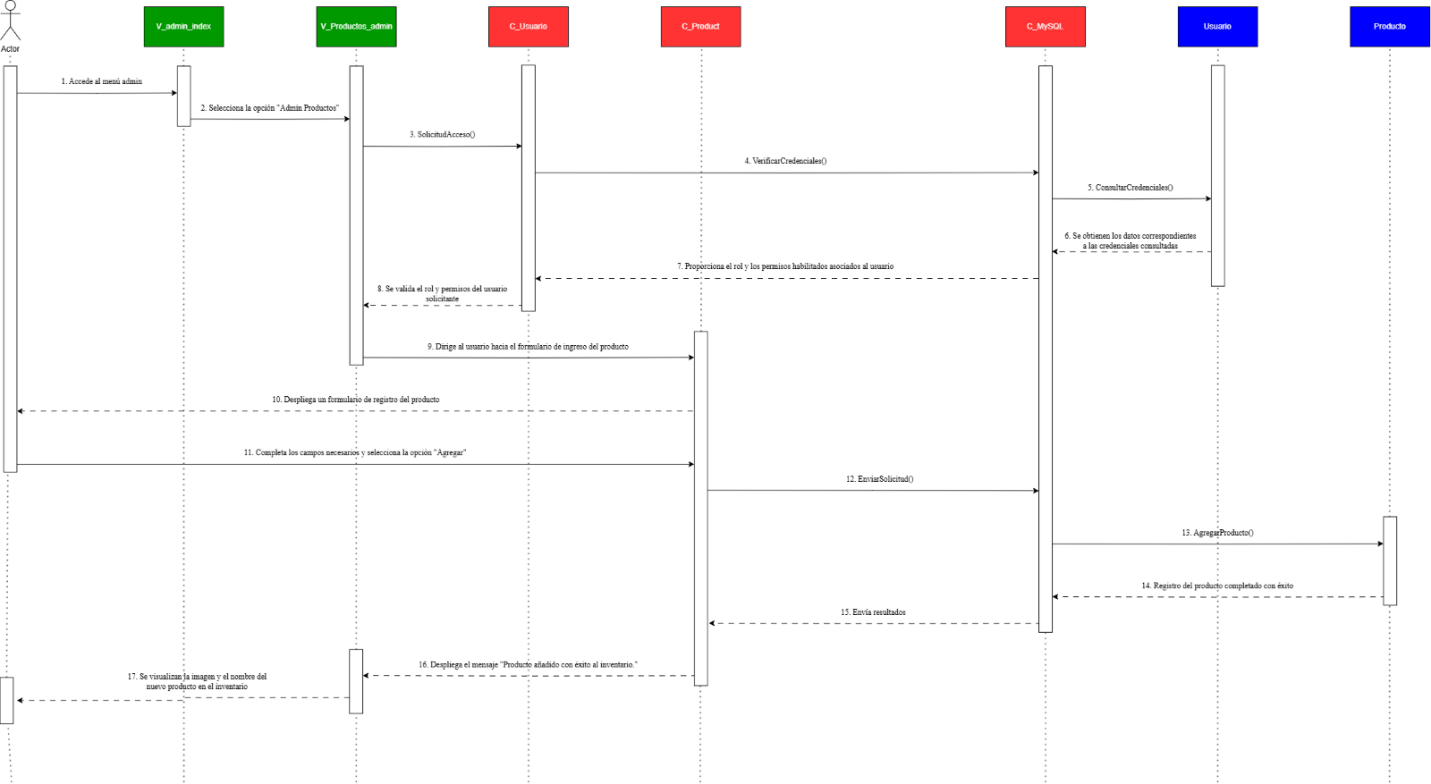
Figura 7.17: “Verificando pedidos mediante código de retiro, error de permisos”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.18*** muestra el proceso mediante el cual un usuario agrega un nuevo producto al catálogo. Para ello, debe haber iniciado sesión con un perfil autorizado, ya sea como administrador o como operador con permisos de gestión de catálogo. Una vez verificada esta condición, el usuario accede al panel “Productos\_admin”, donde puede ingresar las características del nuevo producto completando el formulario correspondiente. Al presionar el botón “Agregar”, el sistema valida la información ingresada y, si no se detectan errores, confirma al usuario que el producto ha sido cargado exitosamente.

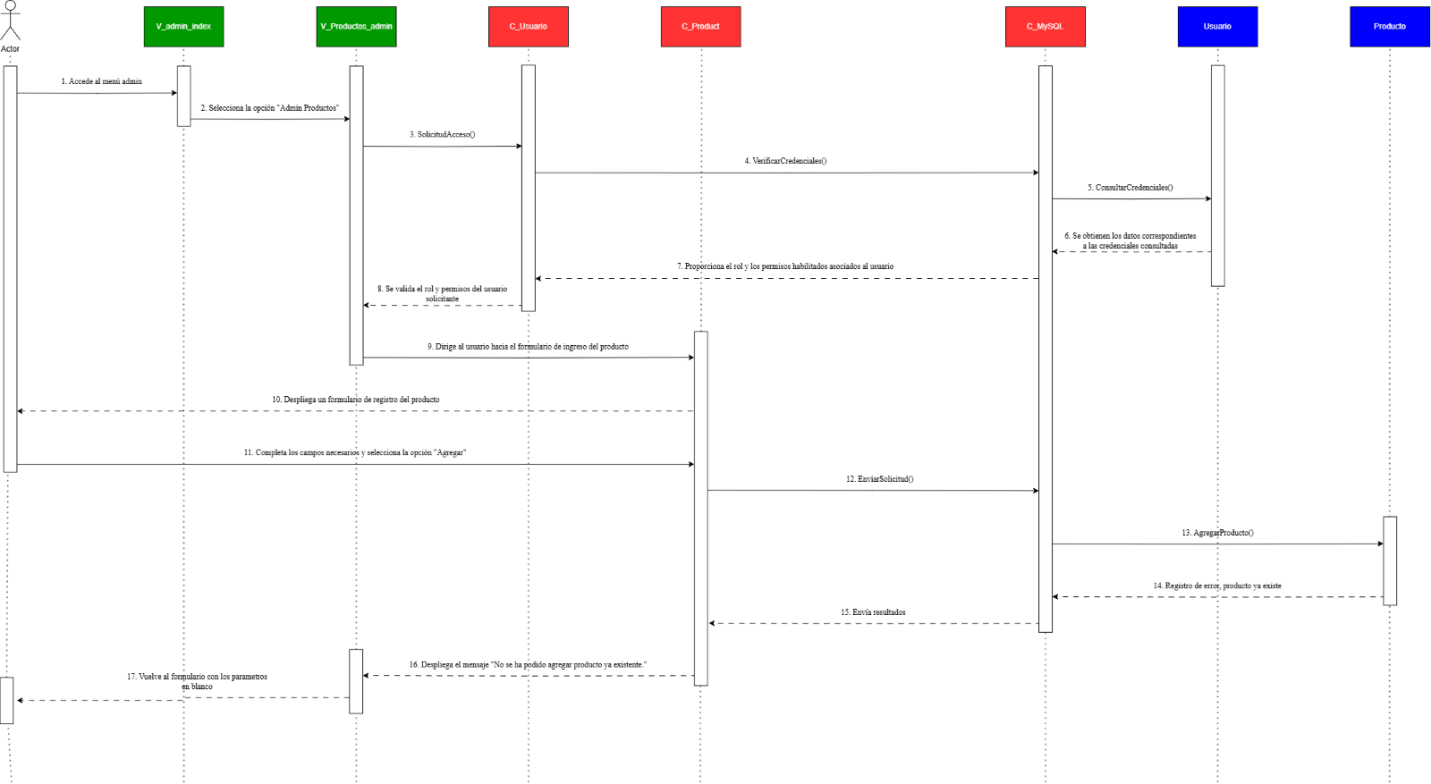
Figura 7.18: “CU N°42, Añadiendo productos al catálogo desde interfaz”



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La Figura ***7.19*** muestra el proceso mediante el cual un usuario intenta agregar un producto al catálogo. Para ello, debe haber iniciado sesión con un perfil autorizado, ya sea como administrador o como operador con permisos de gestión de catálogo. Una vez verificada esta condición, el usuario accede al panel “Productos\_admin”, donde completa el formulario con las características del producto y lo envía al sistema mediante el botón “Agregar”. El sistema valida la información ingresada y, al detectar que el producto ya existe en el catálogo, notifica al usuario que no ha sido posible completar la carga.

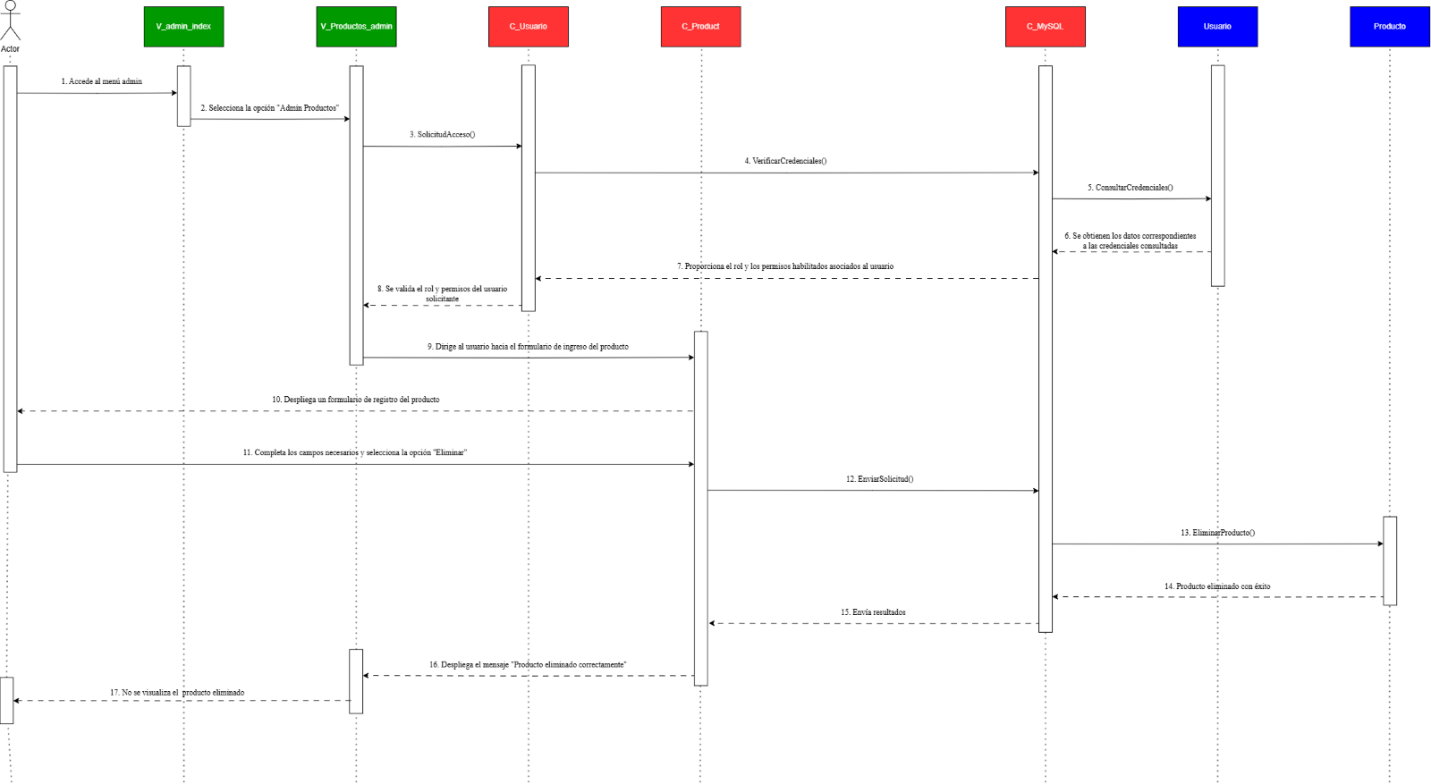
Figura 7.19: “Añadiendo productos al catálogo desde interfaz, Error de carga producto ya existe”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

La ***Figura 7.20*** muestra el proceso mediante el cual un usuario elimina un producto del catálogo. Para realizar esta acción, el usuario debe haber iniciado sesión con credenciales autorizadas, ya sea como administrador o como operador con permisos de gestión de catálogo. Una vez validado el acceso, puede ingresar al panel “Productos\_admin”, donde dispone del botón “Eliminar” para llevar a cabo la operación. Al activarlo, el sistema solicita confirmación de la decisión y, tras recibirla, valida la información correspondiente. Finalmente, se confirma al usuario que el producto ha sido eliminado exitosamente del sistema.

Figura 7.20: “CU N°43, Eliminando productos al catálogo desde interfaz”.

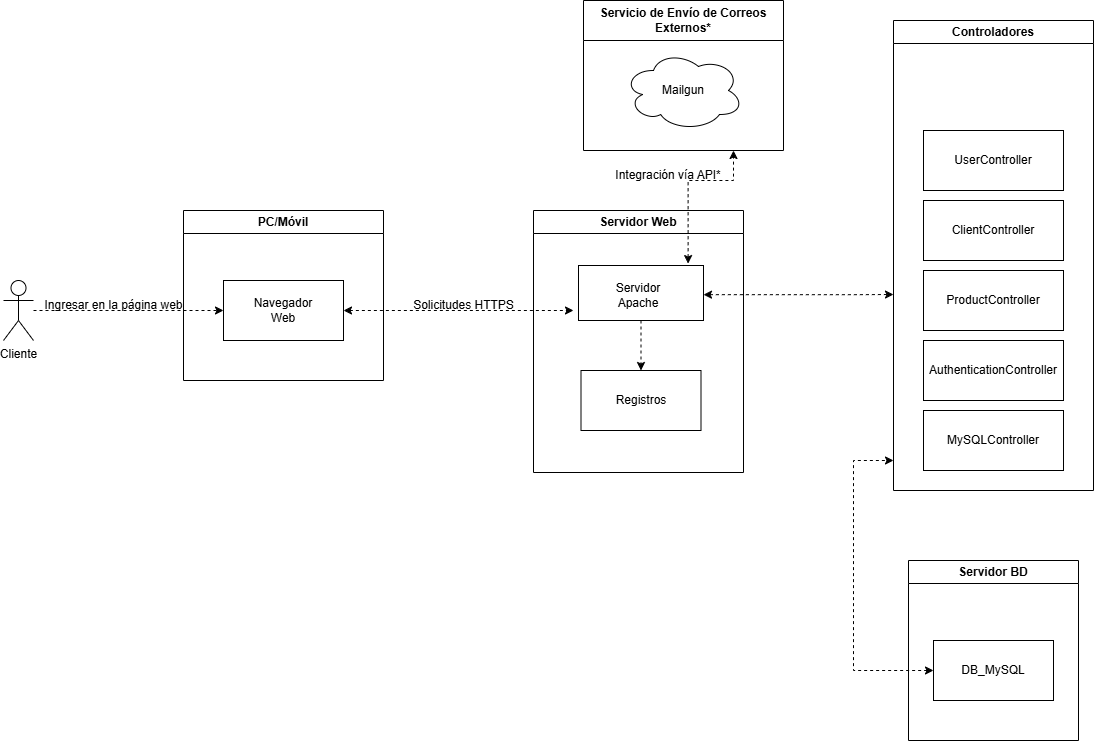


Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Vista de Desarrollo

A continuación, se presenta el diagrama de componentes:

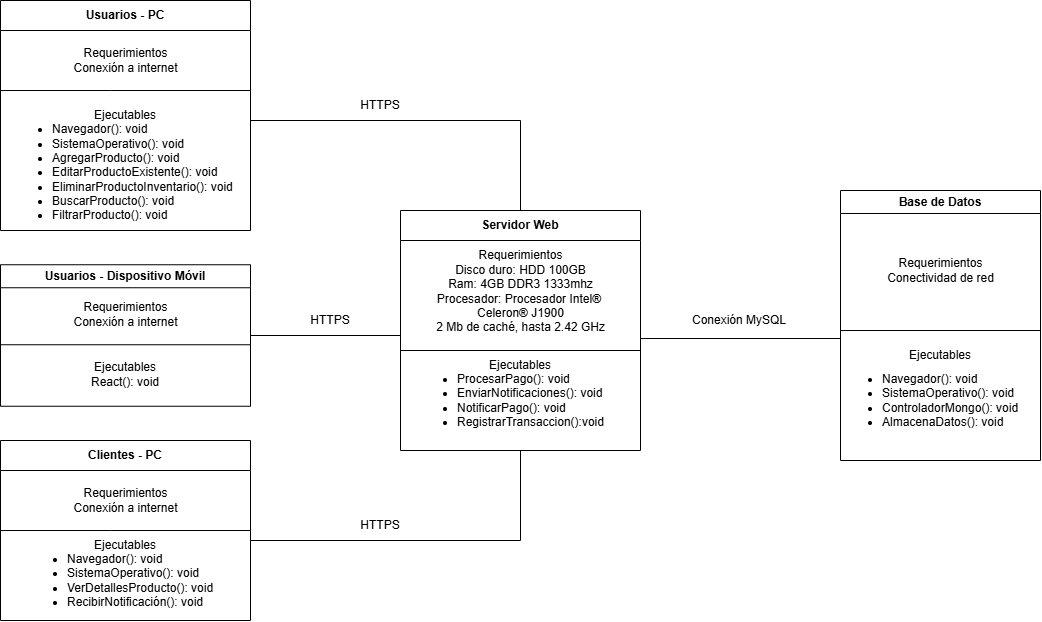
*Figura 8.1: “Vista Desarrollo: Diagrama de Componentes”.*

Fuente: Realizado por el estudiante según el proyecto

# Vista Física

A continuación, se presenta el diagrama de despliegue de la aplicación web “los cobres”.

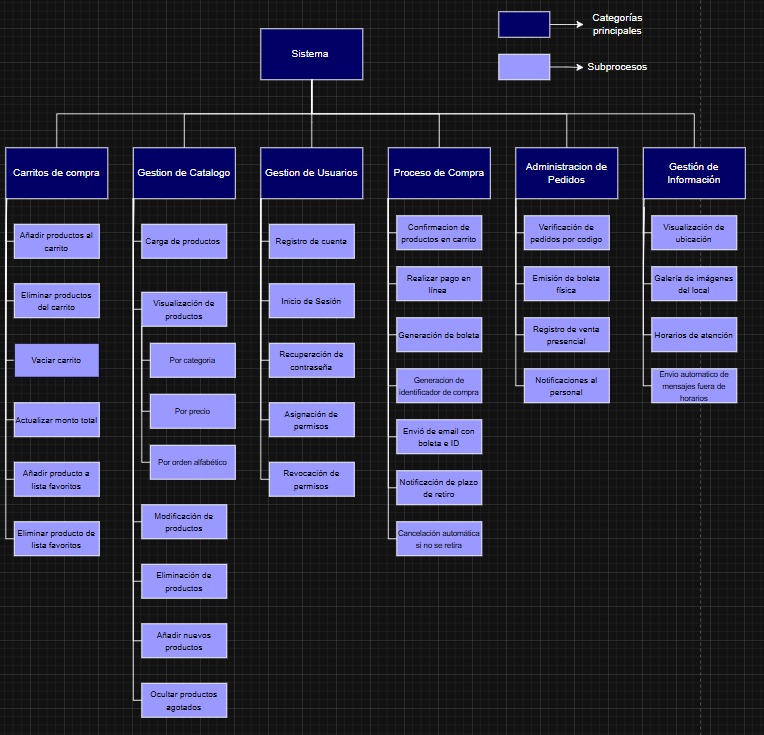
Figura 9.1: “Vista Física: Diagrama de Despliegue”.



Fuente: Realizado por el estudiante según el proyecto.

# Vista Externa

Figura 7.1: “Árbol de navegación”.



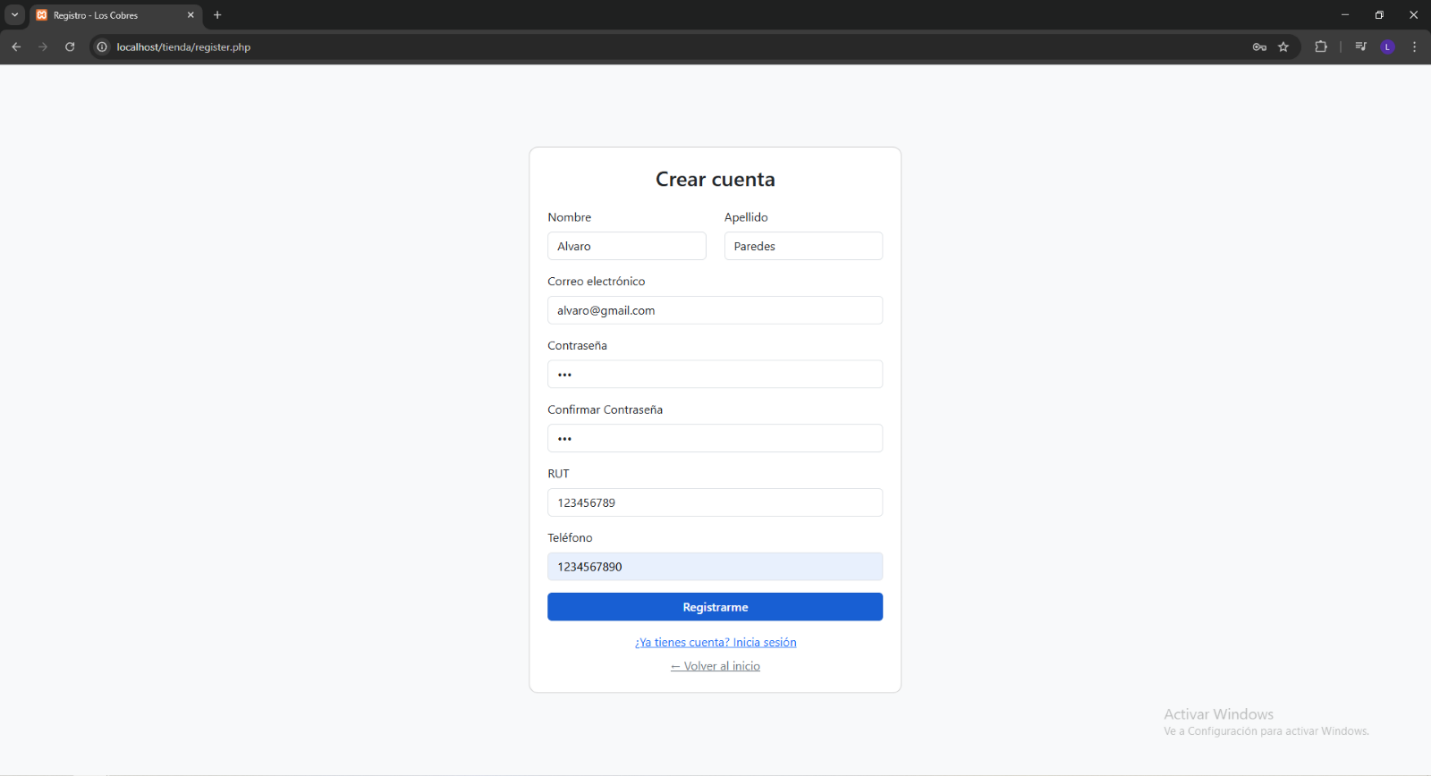
Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Casos de Prueba

A continuación, las siguientes figuras demostraran los casos de prueba.

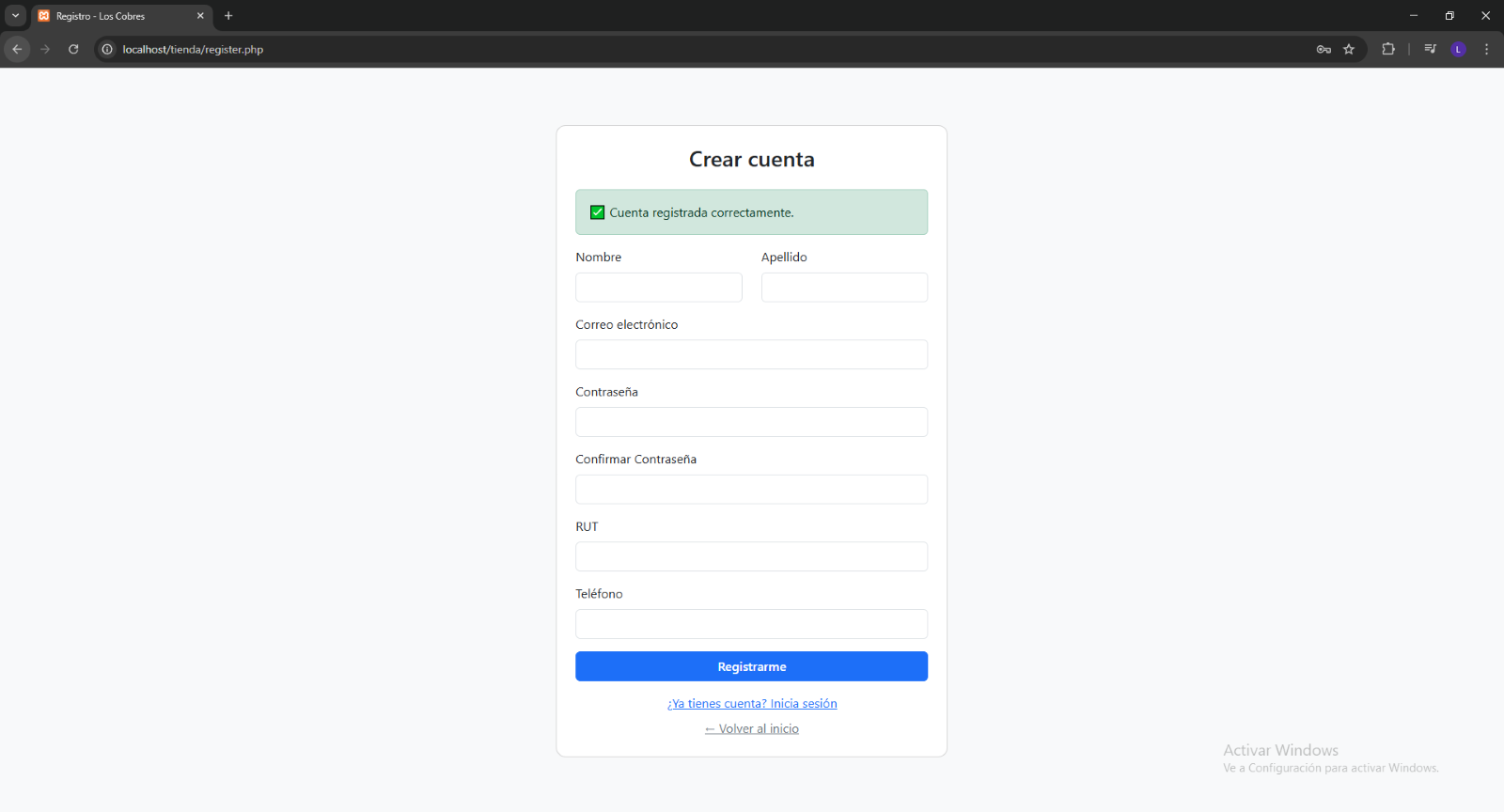
En la ***Figura 11.1*** se visualiza la prueba de registro de nuevo usuario con los campos llenos con información de prueba.

Figura 11.1: ““CU N°7, Prueba de registro de nuevo usuario”.

  
Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la ***Figura 11.2*** se tiene el registro exitoso de un nuevo usuario al sistema.

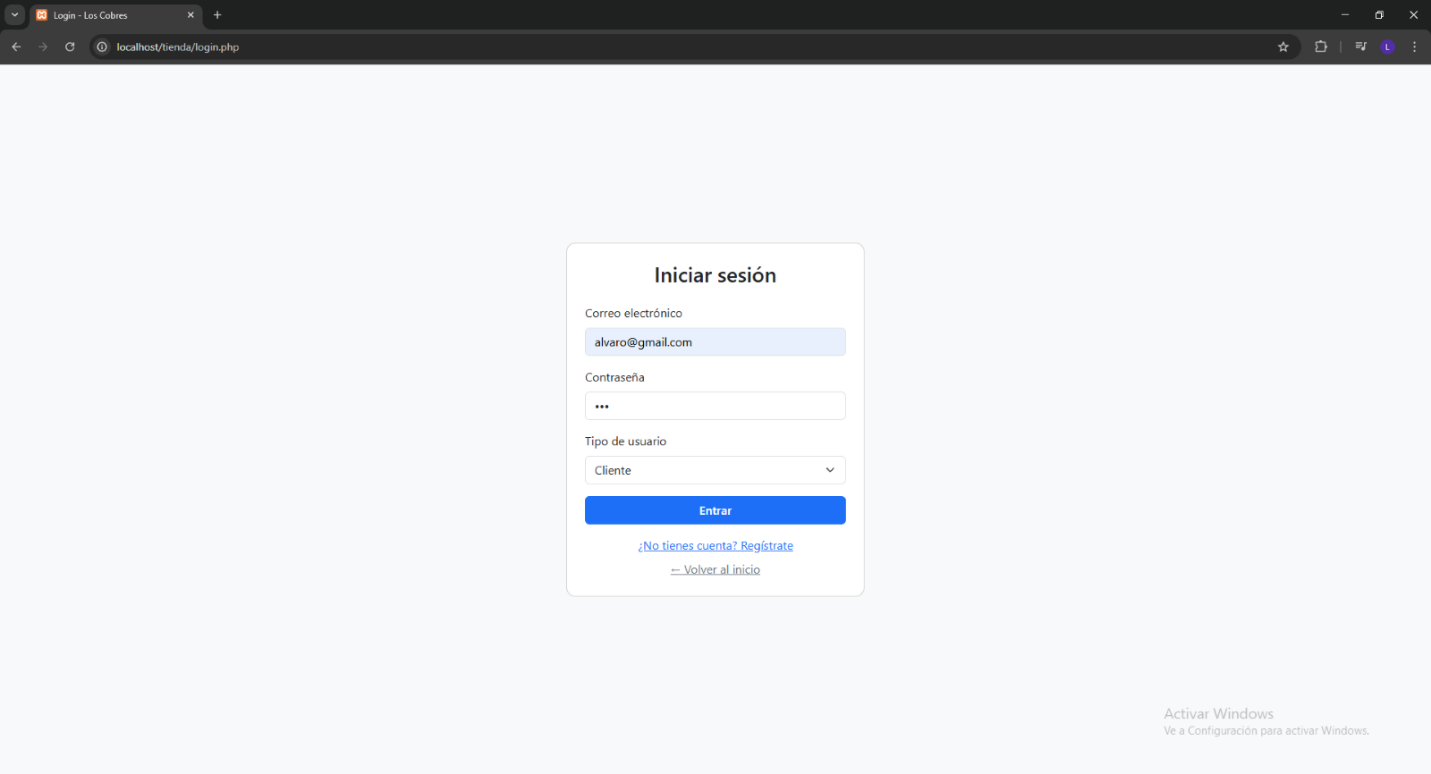
Figura 11.2: ““CU N°7, Prueba de registro de nuevo usuario, Registro exitoso”



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

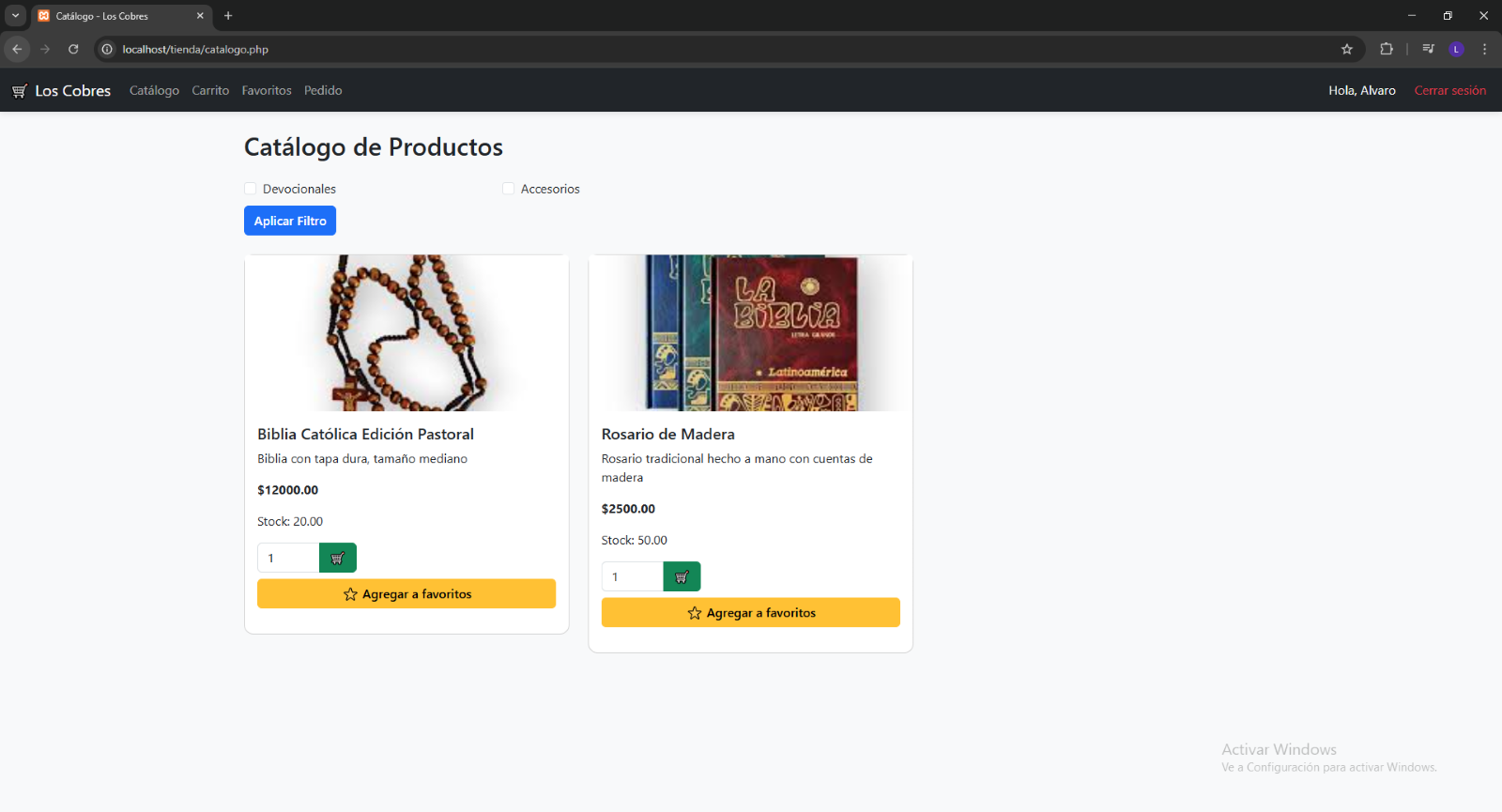
En la figura x.x se demuestra la autenticación de sesión con el usuario creado en la figura anterior

*Figura x.x: “CU N°8, Prueba de autenticación de sesión”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

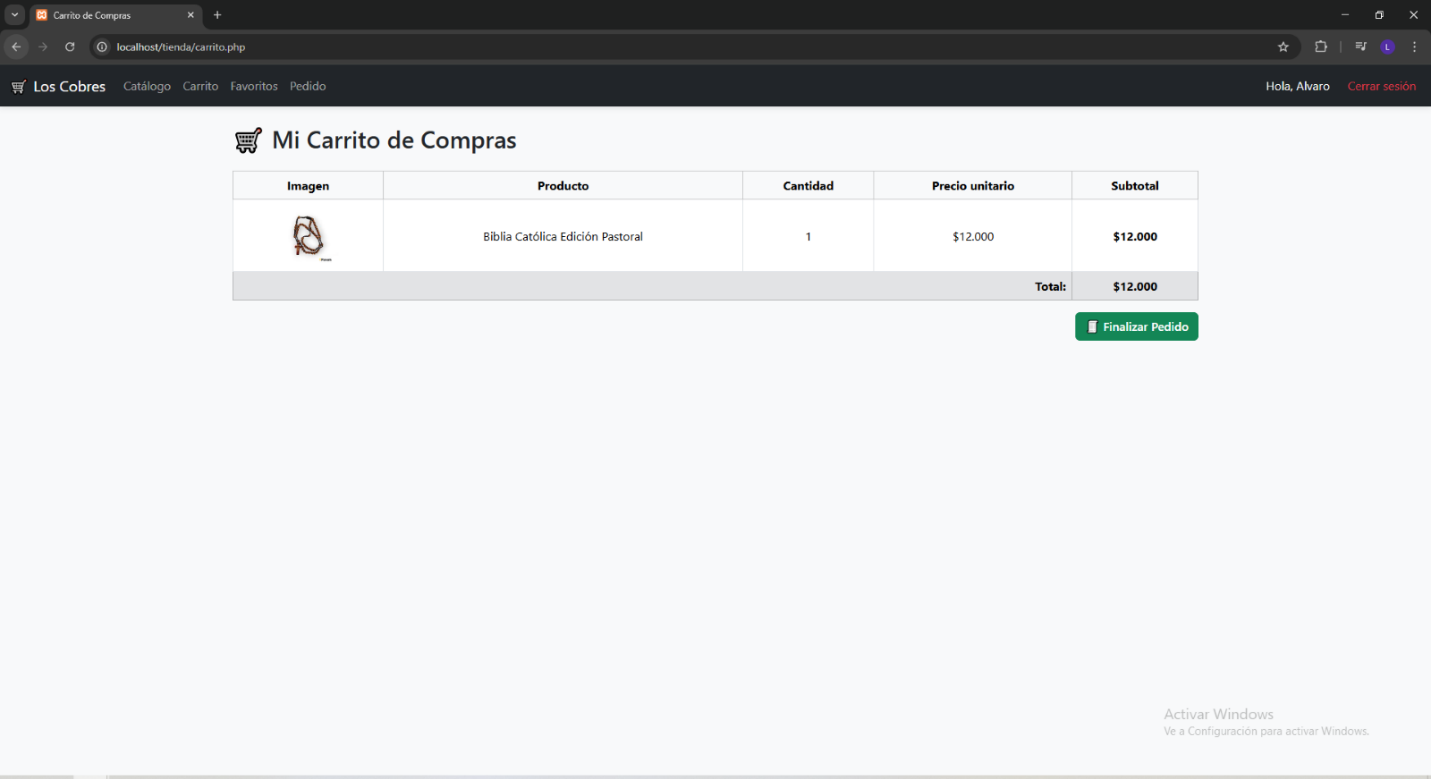
En la figura x.x se demuestra la autenticación de sesión con exito, el usuario entra correctamente al menú principal.

*Figura x.x: “CU N°8, Prueba de autenticación de sesión, inicio exitoso”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la figura x.x se aprecia como en el carrito de compras hay un ítem.

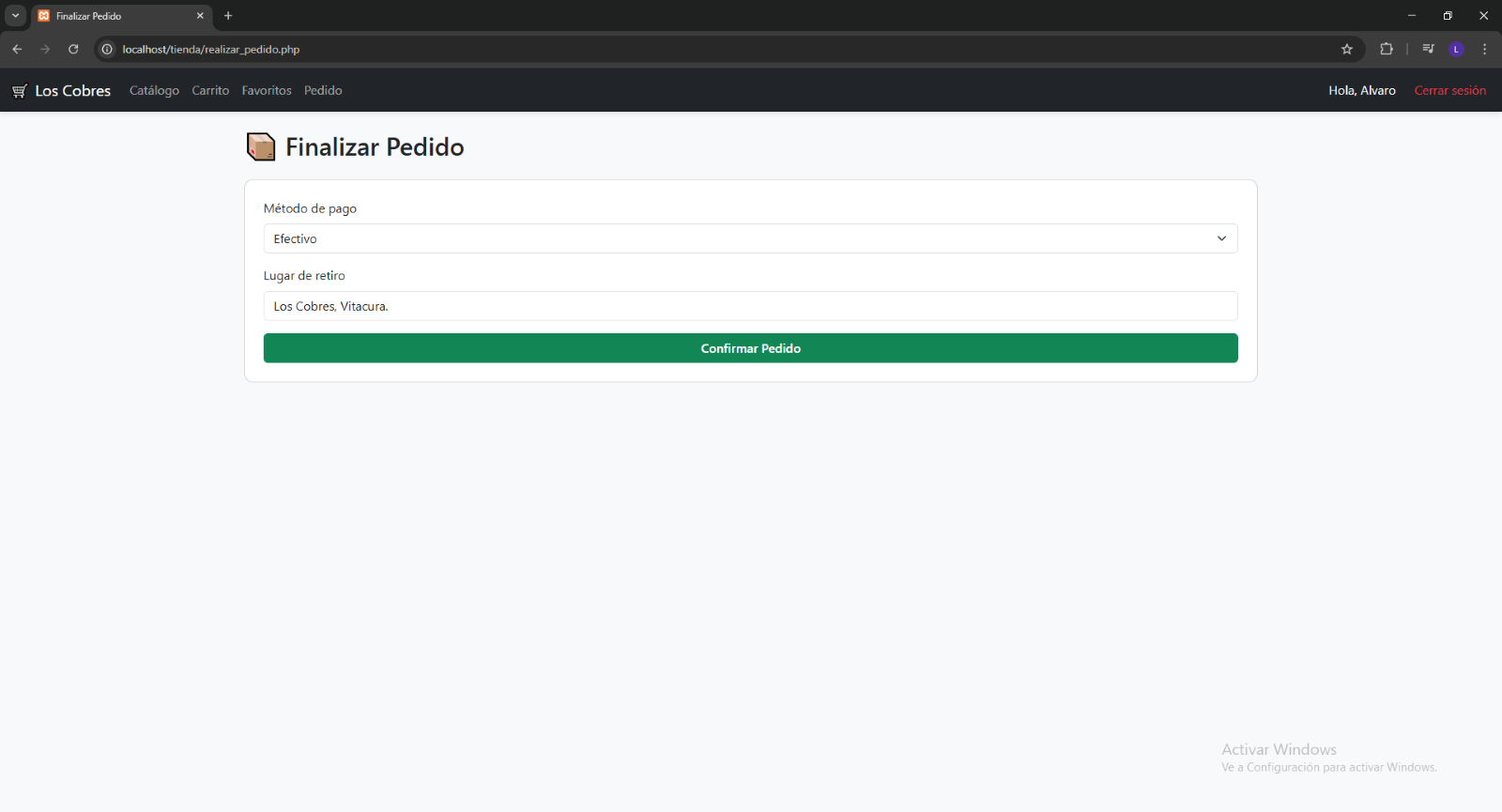
*Figura x.x:” CU N°21, Prueba del carrito cargado con un ítem”*

**

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

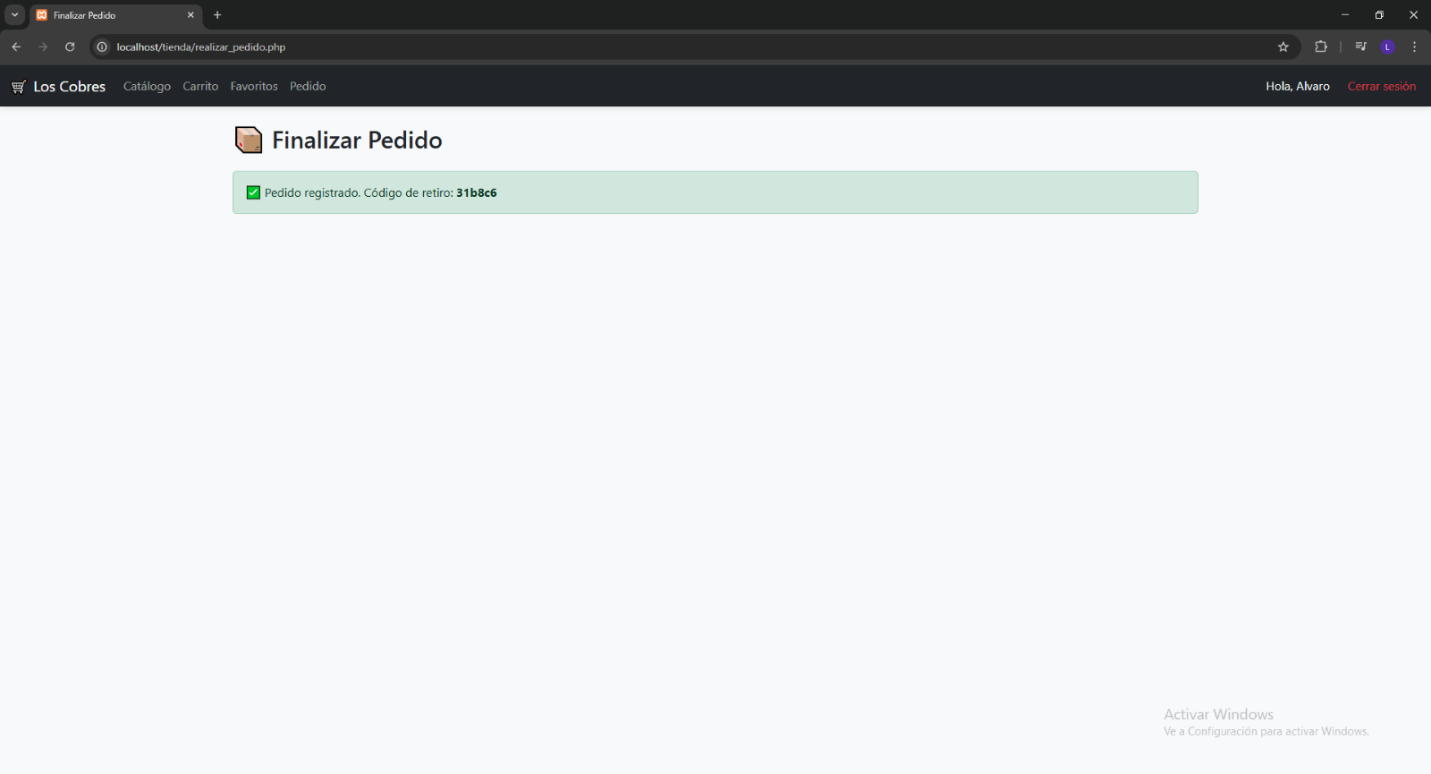
En la figura x.x se aprecia como se finaliza la compra del carrito de compras.

*Figura x.x: “CU N°21, Prueba del pedido siendo finalizado”*

**Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

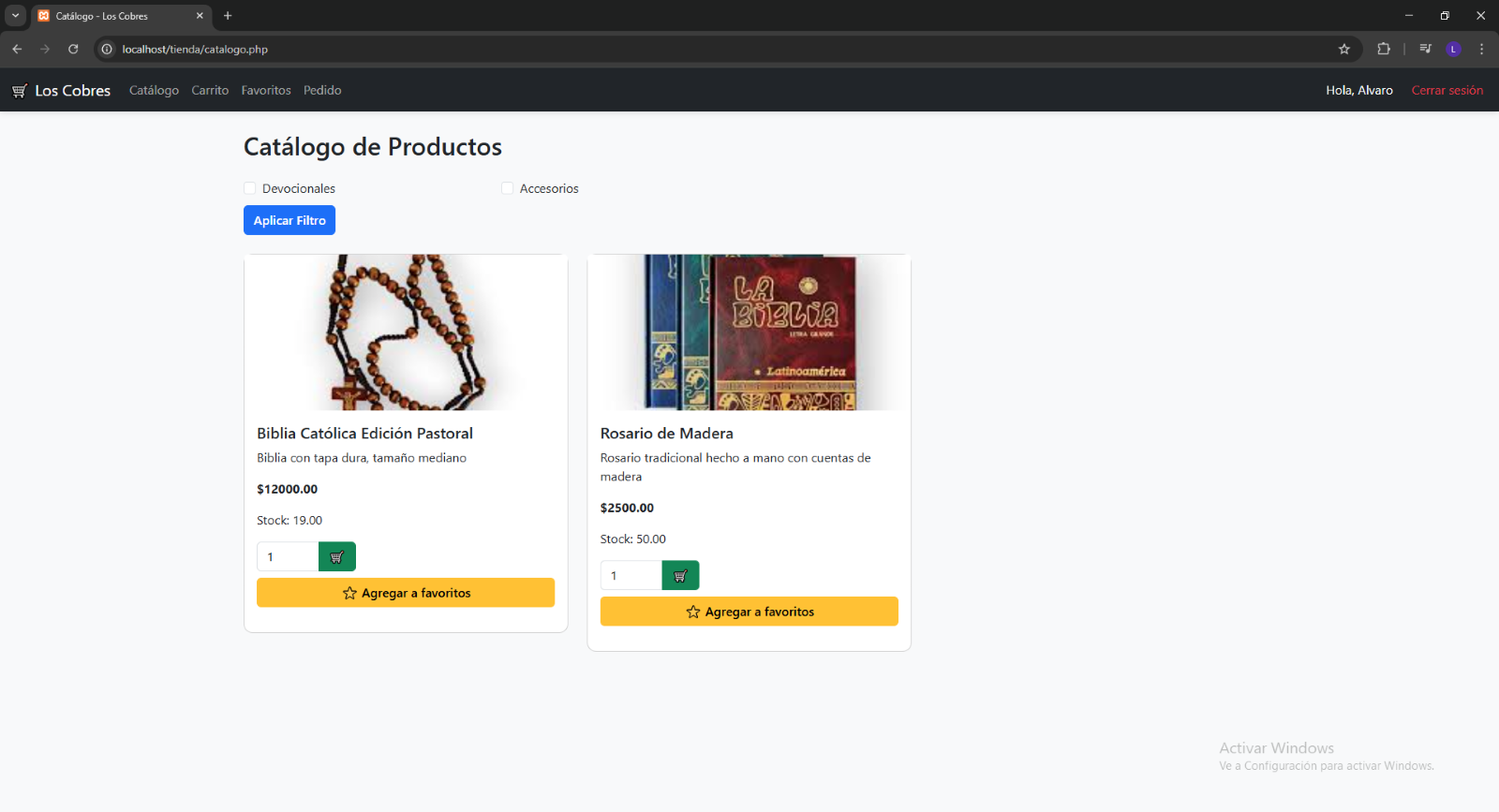
En la figura x.x se aprecia como el pedido ya fue finalizado y entrego el respectivo código de retiro.

*Figura x.x: “CU N°21, Prueba del pedido finalizado”*

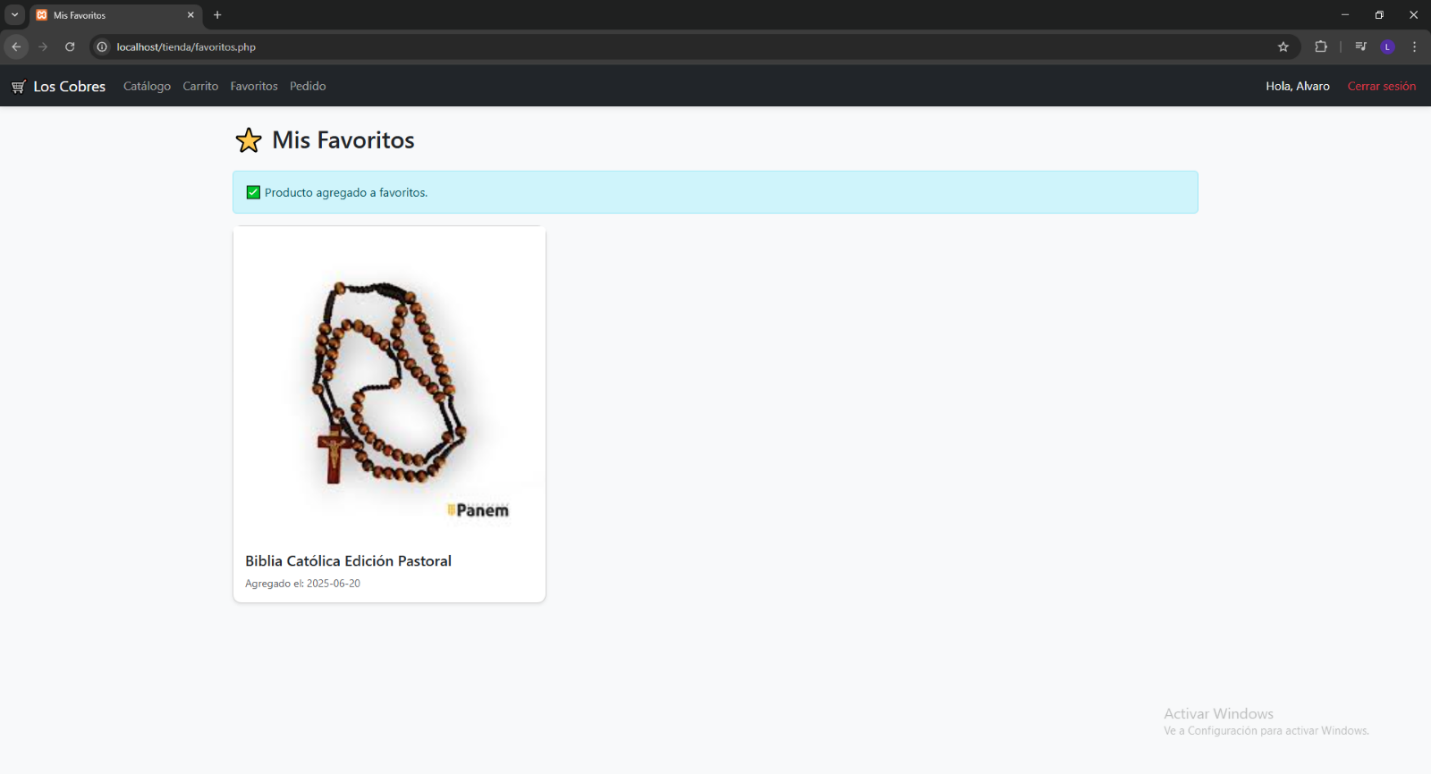
**Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la figura x.x se demuestra el descuento en la cantidad del ítem tras la compra.

*Figura x.x: “CU N°21, Prueba del producto con menos stock“*

**Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

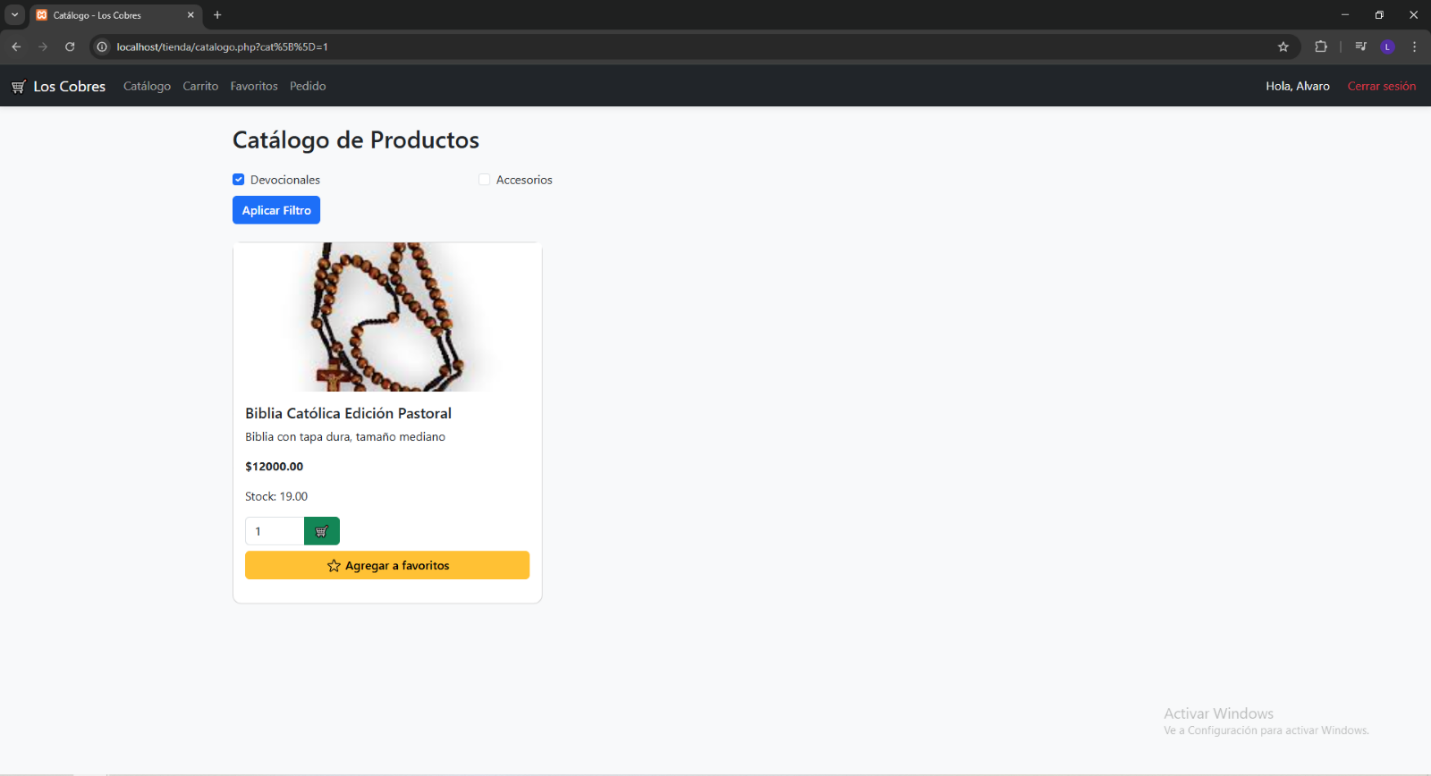
En la figura x.x se muestra cómo se guarda un ítem en favoritos.

*Figura x.x: “CU N°23, Prueba de un ítem agregado a favorito“**,”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

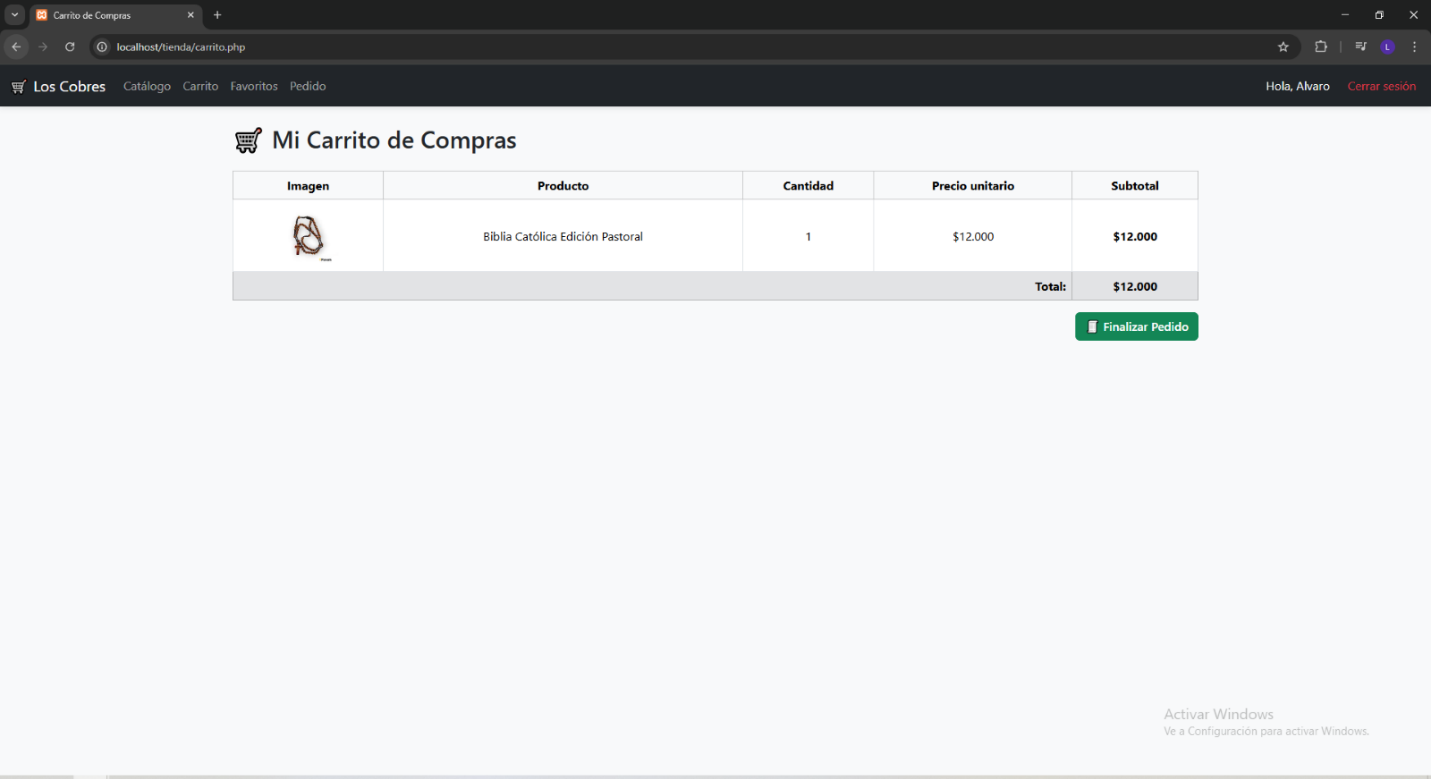
En la figura x.x se muestra cómo se filtran objetos en el catálogo.

*Figura x.x: “CU N°26, Prueba del catálogo filtrado”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

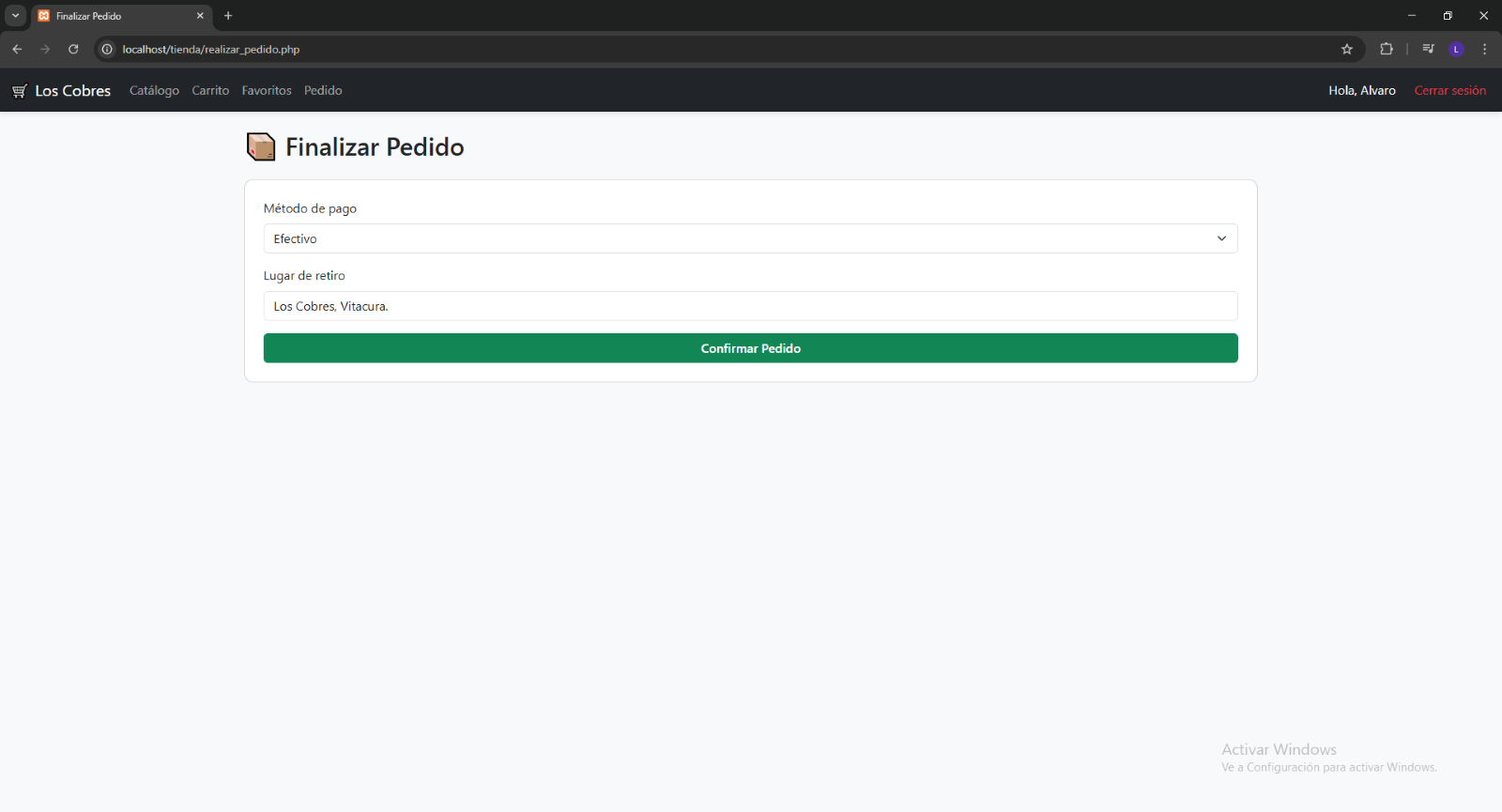
En la figura x.x se muestra el funcionamiento del carrito de compras.

*Figura x.x: “CU N°29, Prueba del carrito de compras”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la figura x.x se muestra cómo se finaliza el pedido, con sus respectivas opciones.

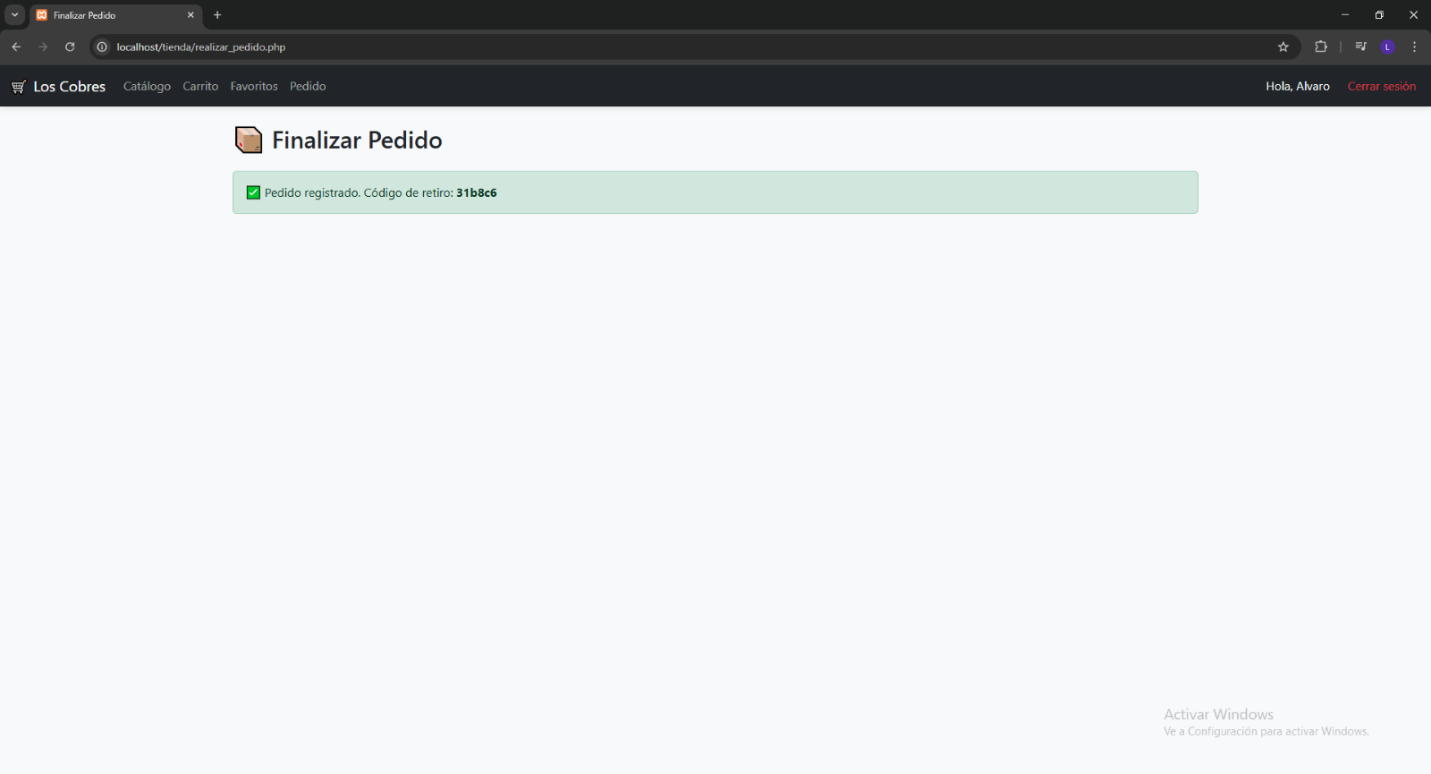
*Figura x.x: “CU N°29, Pruebas de la pestaña de finalizar pedido”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

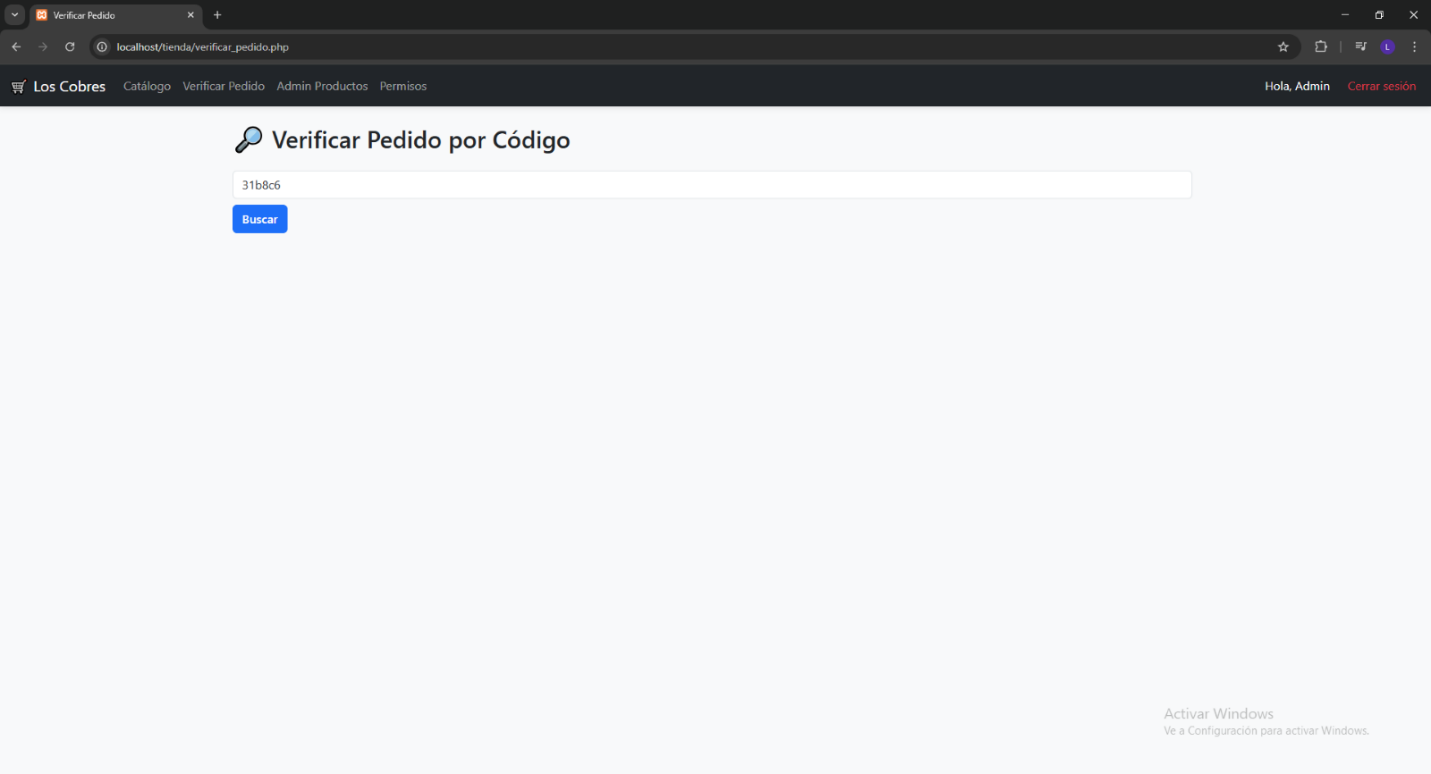
En la figura x.x se muestra la pestaña de finalizar pedido con su respectivo código de retiro.

*Figura x.x: “CU N°29, Prueba del código de retiro, tras finalizar la compra”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

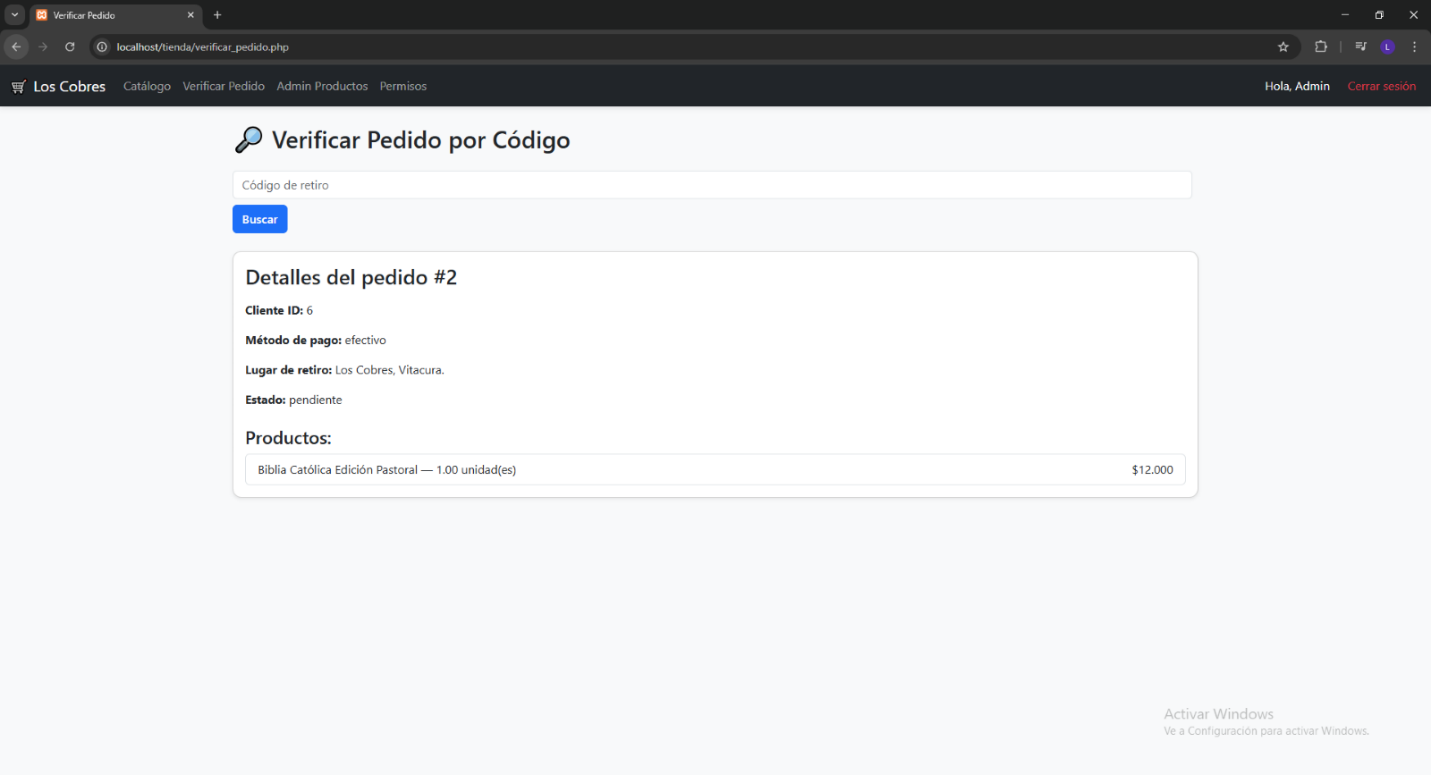
En la figura x.x se .

*Figura x.x: “CU N°40,”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

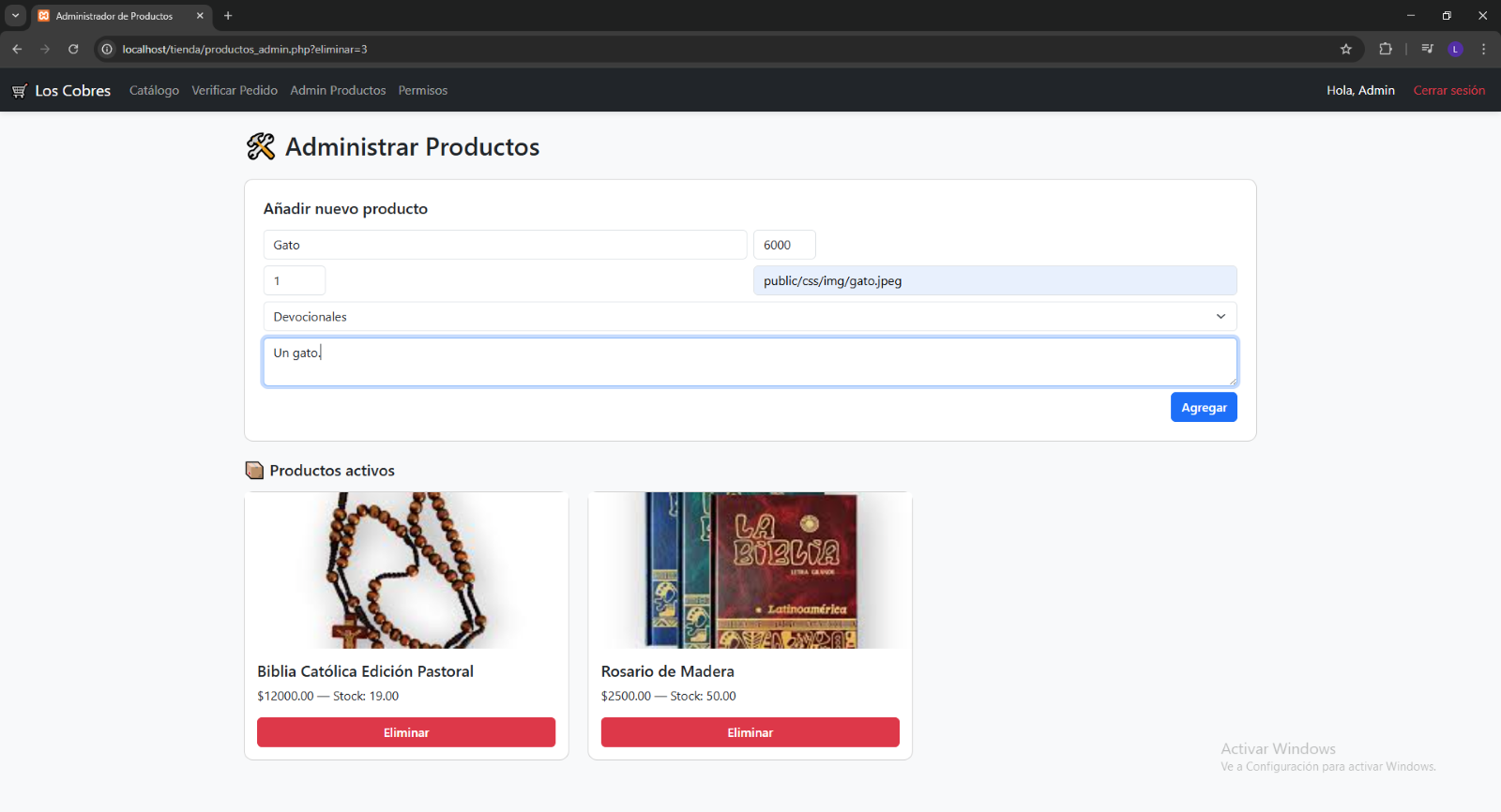
En la figura x.x se .

*Figura x.x: “CU N°40, Verificando pedido por Código”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

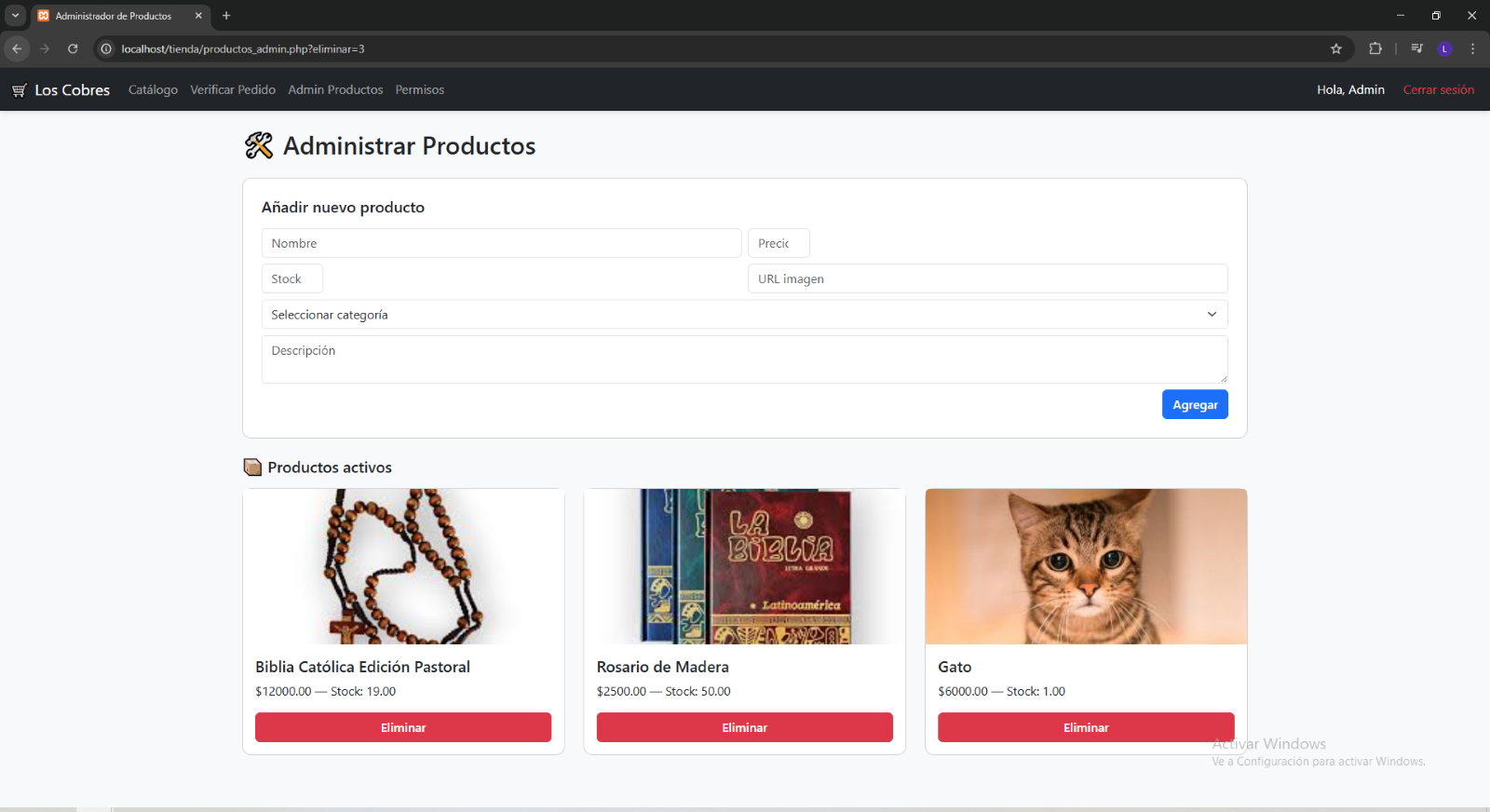
En la figura x.x se muestra como el usuario administrador/operador(catalogo) pueden agregar un producto.

*Figura x.x: “CU N°42, Añadiendo producto al catálogo desde interfaz”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

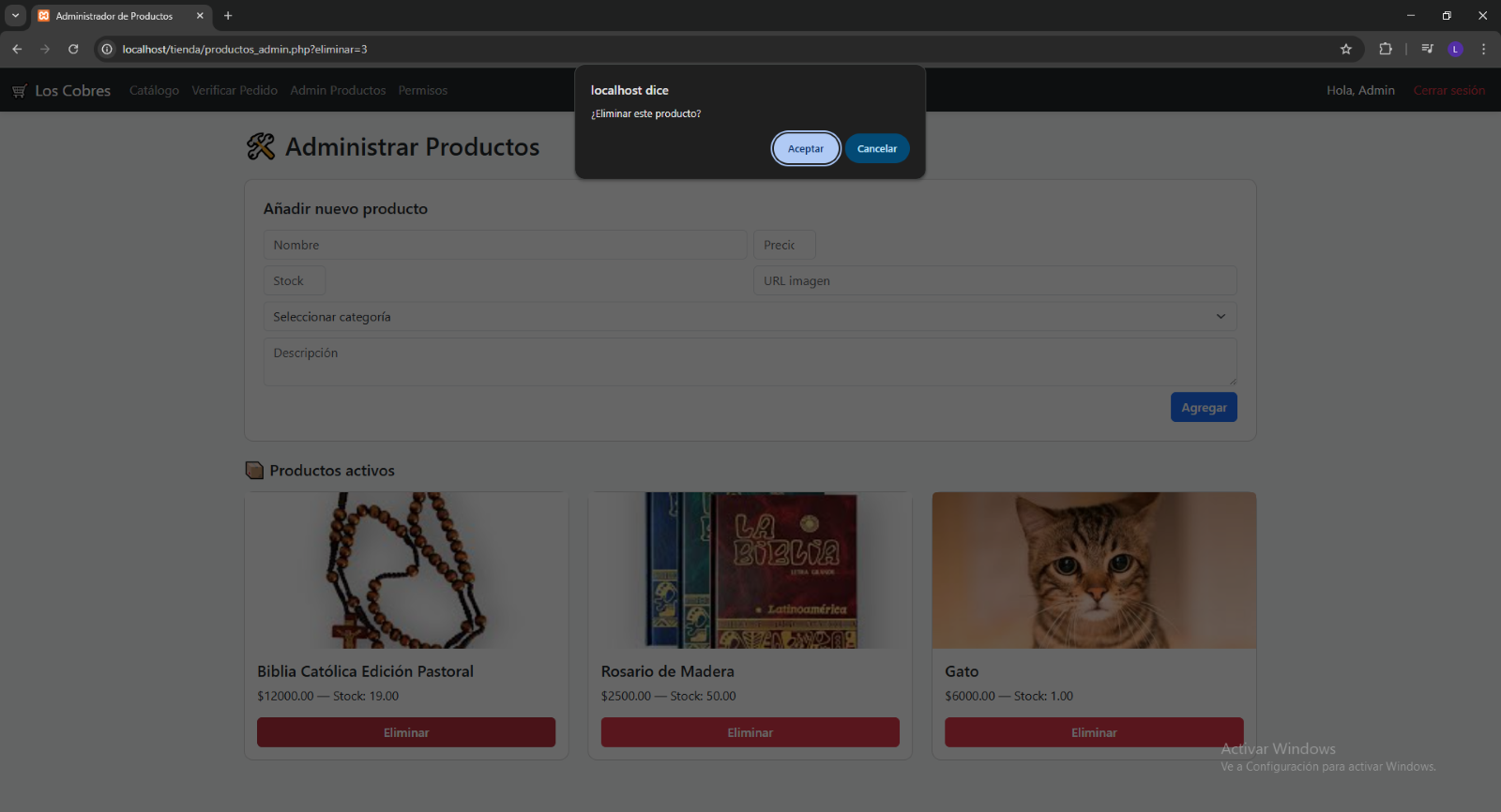
En la figura x.x se visualiza como se ha agregado el nuevo producto.

*Figura x.x: “CU N°42, Añadiendo producto al catálogo desde interfaz, producto agregado”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

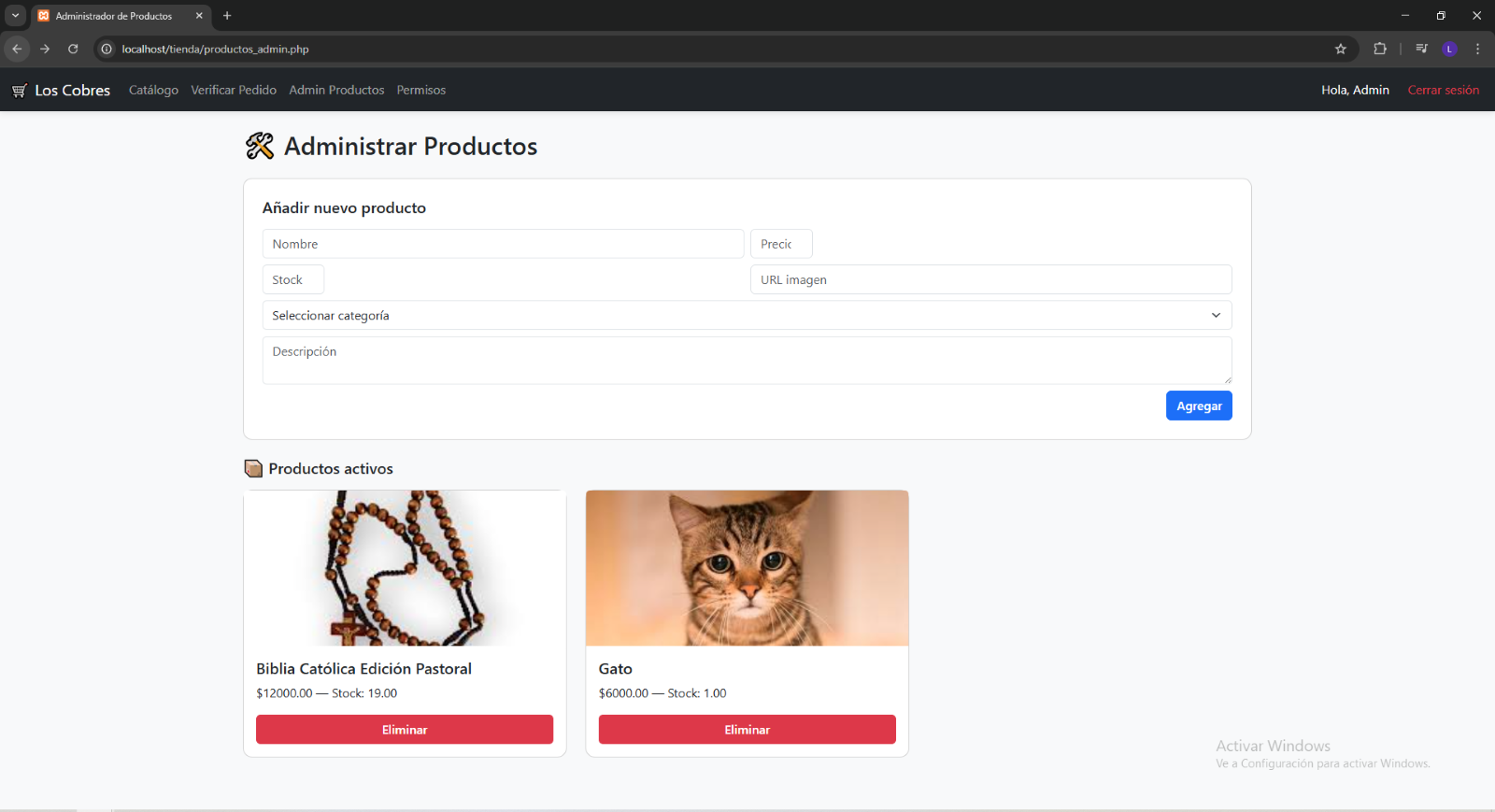
En la figura x.x se visualiza como el sistema pide confirmación cuando el administrador/operador(catalogo) al intentar eliminar un producto.

*Figura x.x: “CU N°43, Eliminando producto del catálogo desde interfaz”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la figura x.x se visualiza el producto eliminado.

*Figura x.x: “CU N°43, Eliminando producto del catálogo desde interfaz”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

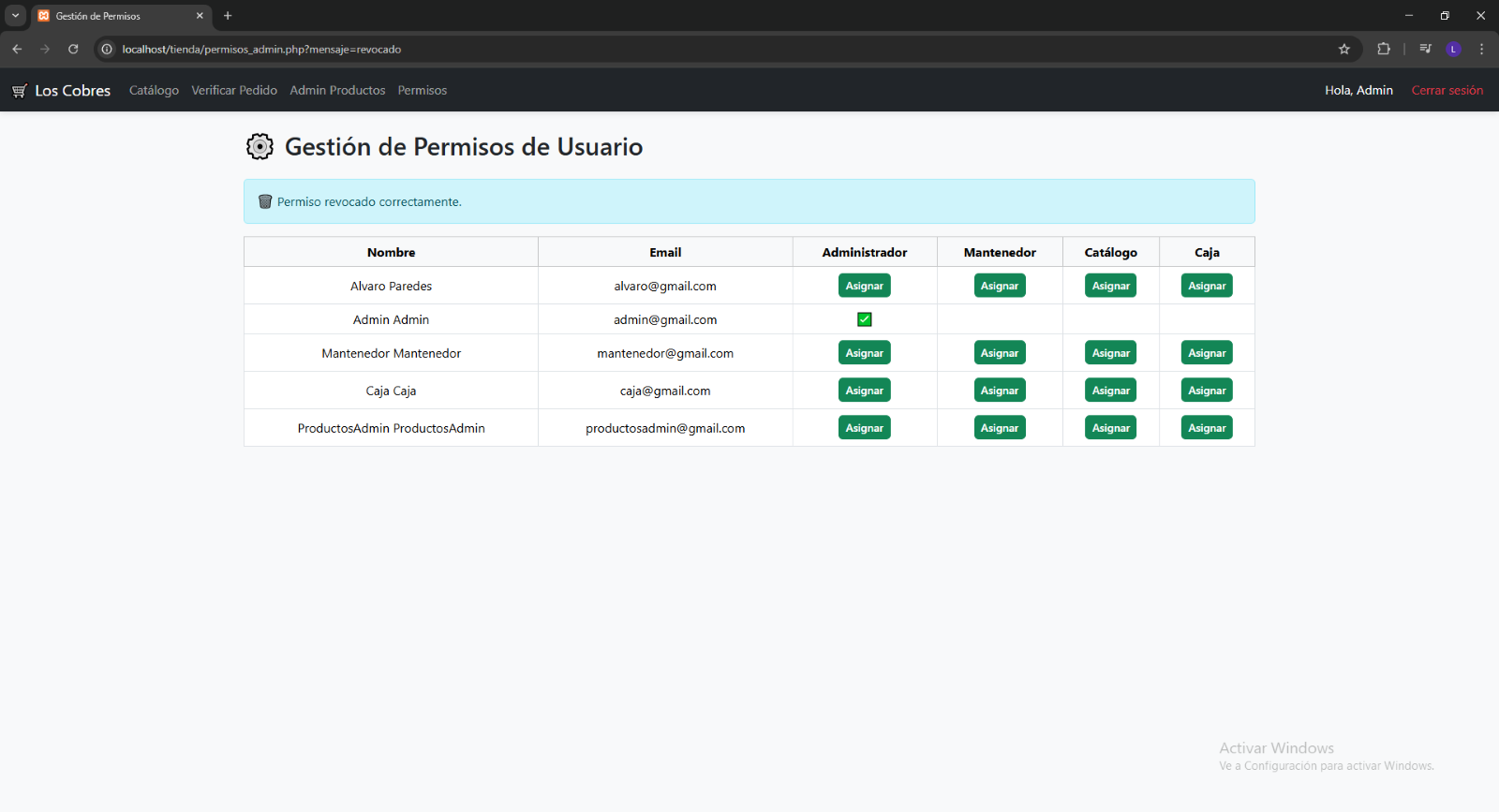
En la figura x.x se visualiza que no se puede eliminar producto existente en pedidos en curso.

*Figura x.x: “CU N°43, Eliminando producto del catálogo desde interfaz, advertencia intento eliminar producto en pedido”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

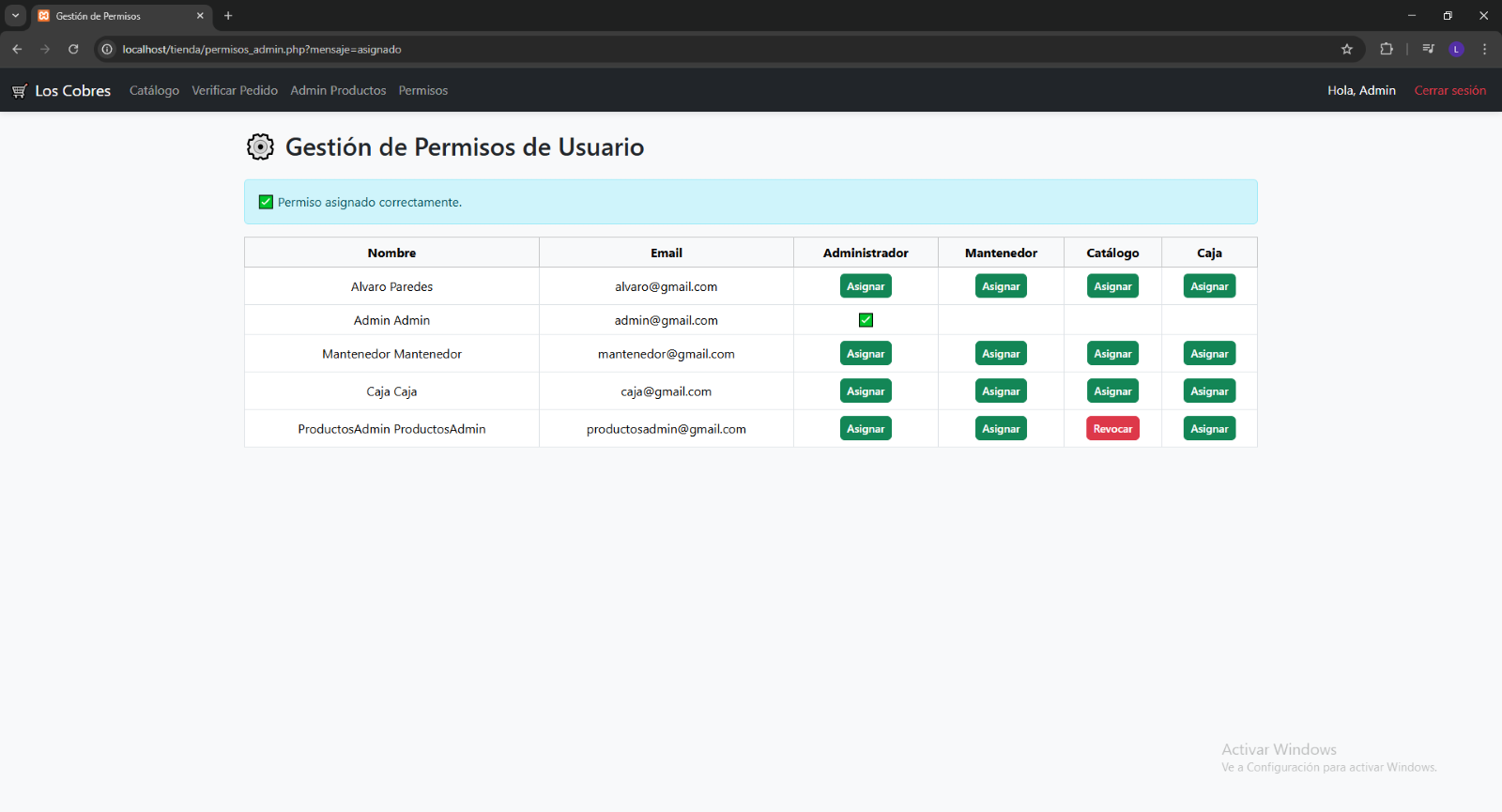
En la figura x.x se visualiza que el administrador a revocando un permiso exitosamente.

*Figura x.x: “CU N°45,”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

En la figura x.x se visualiza que el administrador a otorgado un permiso exitosamente.

*Figura x.x: “CU N°45, Gestionando permisos usuarios, otorgando permiso catalogo”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto.

# Retrospectiva

Descripción, Sprint review.

## Gráfico Burn-Up

Descripción, Figura. Adjuntar una vez efectuadas todas las horas para cada tarea asignada en el Product Backlog List.

## Gráfico Burn-Down

Descripción breve, Figura. “ ”.

**ANEXO A: RECURSOS**

En la ***Figura A.1*** se presenta el registro de eventos y tareas del grupo de integrantes del proyecto para efectos de planificación interna.

Figura A.1: “Organización del Trabajo, Primer Incremento”.



Fuente: Elaborado por el estudiante de acuerdo con el proyecto.