

Universidad de Alicante

Máster en Desarrollo de Aplicaciones y Servicios Web

Programación Avanzada en Entornos de Escritorio

Proyecto Final

TPV de Comercio – Tienda Musical

Maitán Jiménez, José Daniel

jmdj3@alu.ua.es

Alicante, Mayo 2025

ÍNDICE DE CONTENIDOS

1. Introducción	4
2. Tecnologías utilizadas	4
3. Estructura del proyecto	5
3.1. Modelos.....	5
3.2. Vistas	6
3.3. ViewModels.....	7
3.4. Conexión a base de datos	7
3.5. Servicios.....	8
4. Funcionalidades principales.....	9
4.1. Autenticación	9
4.2. Catálogo de productos	10
4.3. Carrito de compras.....	11
4.4. Proceso de pago	12
4.5. Historial de ventas.....	14
4.6. CRUD de productos	17
4.7. Estadísticas y Reportes.....	21
4.8. Internacionalización	22
4.9. Persistencia de Interfaz	24
5. Conclusiones	25

LISTA DE ILUSTRACIONES

Ilustración 1. Modelos utilizados	6
Ilustración 2. Vistas utilizadas	6
Ilustración 3. ViewModels utilizados	7
Ilustración 4. Clases para manejo de base de datos.....	8
Ilustración 5. Vista de autenticación.....	9
Ilustración 6. Vista de catálogo	10
Ilustración 7. Ejemplo de búsqueda y filtrado	11
Ilustración 8. Vista de carrito	12
Ilustración 9. Vista de checkout.....	14
Ilustración 10. Compra exitosa	14
Ilustración 11. Vista de historial de ventas	16
Ilustración 12. Detalle de venta pasada.....	16
Ilustración 13. Ejemplo de exportación a PDF	17
Ilustración 14. Vista de administración de productos	19
Ilustración 15. Vista de detalle de producto.....	19
Ilustración 16. Vista de creación/edición de productos	20
Ilustración 17. Ejemplo de creación de producto	20
Ilustración 18. Botones para operaciones CUD	21
Ilustración 19. Vista de estadísticas y reportes	22
Ilustración 20. Ejemplo de cambio de idioma en UI	24

1. Introducción

Las aplicaciones de escritorio continúan desempeñando un papel fundamental en el panorama actual del software, ofreciendo robustez, rendimiento y una interacción directa con el hardware que las aplicaciones web a menudo no pueden igualar. Su utilidad se evidencia particularmente en entornos comerciales, donde la eficiencia y la fiabilidad son cruciales. En este contexto, el desarrollo de un sistema de Terminal de Punto de Venta (TPV) se presenta como una herramienta esencial para la modernización y optimización de la gestión operativa de los negocios. Este proyecto se centra en el diseño e implementación de un sistema TPV específico para una tienda de artículos musicales, buscando facilitar la administración de productos, la realización de ventas y el seguimiento de clientes. Para alcanzar los objetivos planteados, se ha empleado la plataforma Windows Presentation Foundation (WPF), la cual proporciona un marco de trabajo sólido y flexible para la creación de interfaces de usuario ricas y modernas. El sistema desarrollado abarca las funcionalidades esenciales para la operación del TPV, incluyendo la autenticación de usuarios, el catálogo de productos, el carrito de compras, el proceso de pago, las listas de ventas, el CRUD de productos, las estadísticas y reportes, la generación de facturas en PDF, y la internacionalización y persistencia de la interfaz.

2. Tecnologías utilizadas

El desarrollo de este sistema TPV se ha llevado a cabo empleando un conjunto de tecnologías modernas y robustas, seleccionadas para garantizar el rendimiento, la escalabilidad y la mantenibilidad de la aplicación. La base del proyecto se construye sobre la plataforma **.NET 8**, que proporciona un entorno de ejecución eficiente y un amplio conjunto de bibliotecas para el desarrollo de aplicaciones de escritorio. Para la gestión de la interacción con la base de datos, se ha utilizado **Entity Framework Core (EF Core)**, un ORM (Object-Relational Mapper) que facilita el acceso y la manipulación de los datos, permitiendo una mayor abstracción y reduciendo la complejidad del código de acceso a datos. La base de datos elegida para el almacenamiento persistente de la información es **SQL Server**, un sistema de gestión de bases de datos relacional que destaca por su fiabilidad y rendimiento. La interfaz de usuario de la aplicación se ha desarrollado con **Windows Presentation Foundation (WPF)**,

un marco de trabajo de .NET que permite la creación de interfaces gráficas ricas y atractivas, adaptadas a las necesidades de un sistema TPV. Para la generación de los documentos de factura en formato PDF, se ha integrado la librería **iTextSharp**, que ofrece las funcionalidades necesarias para la creación y manipulación de archivos PDF de manera programática. Finalmente, para la gestión del registro de eventos y el seguimiento del comportamiento de la aplicación, se ha implementado **Serilog**, una librería de logging flexible y potente que permite la captura y el almacenamiento eficiente de los registros de la aplicación.

3. Estructura del proyecto

La arquitectura de este sistema TPV se basa en el patrón de diseño **Modelo-Vista-VistaModelo (MVVM)**, el cual ha sido fundamental para lograr una separación clara de responsabilidades entre la lógica de negocio, la interfaz de usuario y la presentación de los datos. Esta separación facilita el desarrollo, las pruebas y el mantenimiento de la aplicación, promoviendo un código más limpio y organizado. A continuación, se detallan los componentes principales de la estructura del proyecto:

3.1. Modelos

La capa de Modelos representa la estructura de los datos de la aplicación. Las clases de modelo definen las entidades del dominio, como "Producto", "Venta", "Cliente" y "Usuario". Estas clases son simples objetos de datos (POCOs) que contienen las propiedades necesarias para representar la información y no incluyen lógica de negocio. Su principal función es la de transportar datos entre las diferentes capas de la aplicación.

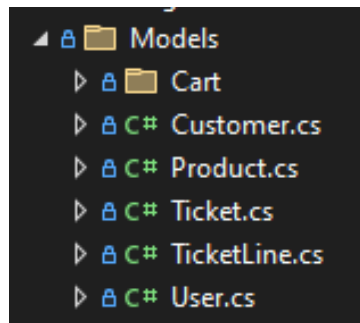


Ilustración 1. Modelos utilizados

3.2. Vistas

La capa de Vistas se encarga de la presentación visual de la aplicación. Las Vistas están implementadas utilizando XAML y definen la interfaz de usuario, incluyendo la disposición de los controles, la presentación de los datos y la interacción con el usuario. Las Vistas son pasivas, es decir, su lógica se limita a la presentación y no contienen lógica de negocio. Se vinculan a los ViewModels a través del Data Binding de WPF para mostrar los datos y responder a las acciones del usuario.

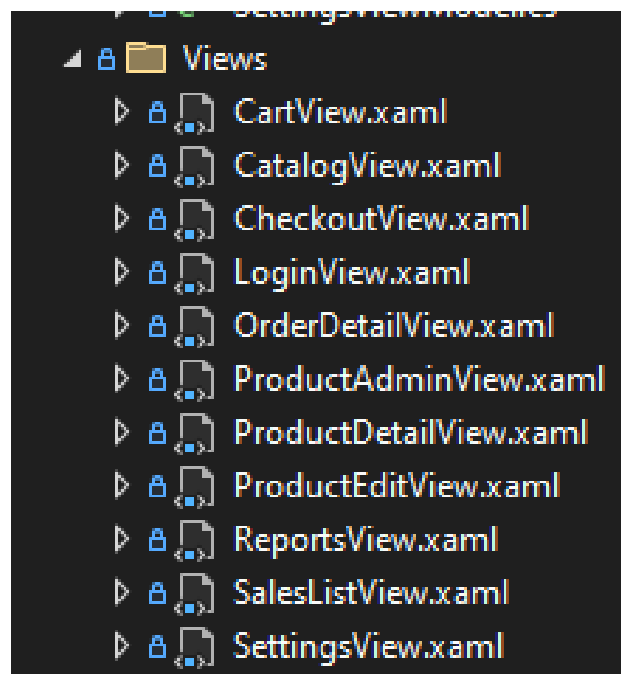


Ilustración 2. Vistas utilizadas

3.3. ViewModels

La capa de ViewModels actúa como intermediario entre las Vistas y los Modelos. Los ViewModels exponen los datos de los Modelos en un formato adecuado para su presentación en las Vistas y contienen la lógica de presentación, como la validación de datos, la transformación de datos y la gestión de las acciones del usuario. Los ViewModels no tienen conocimiento directo de las Vistas, lo que permite que una misma ViewModel pueda ser utilizada por diferentes Vistas.

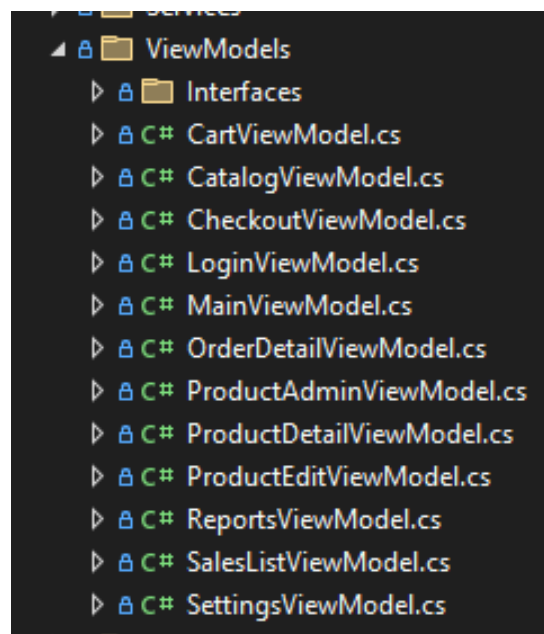


Ilustración 3. ViewModels utilizados

3.4. Conexión a base de datos

La conexión a la base de datos se gestiona a través de Entity Framework Core. Esta tecnología proporciona una abstracción de la base de datos, permitiendo que la aplicación interactúe con los datos utilizando objetos .NET en lugar de código SQL directamente. El contexto de la base de datos se encarga de establecer la conexión, realizar las consultas y guardar los cambios en la base de datos SQL Server.

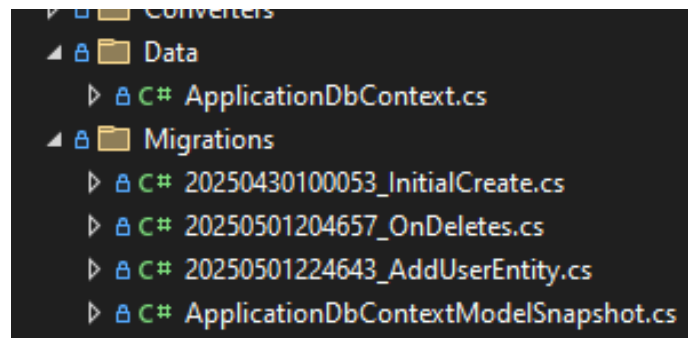


Ilustración 4. Clases para manejo de base de datos

3.5. Servicios

La capa de Servicios encapsula la lógica reutilizable y las operaciones que no están directamente relacionadas con la interfaz de usuario ni con la gestión de la base de datos. En este proyecto, se han implementado los siguientes servicios:

- **LocalizationService:** Este servicio se encarga de la gestión de la internacionalización y localización de la aplicación. Proporciona la funcionalidad para acceder a los recursos de texto específicos del idioma, permitiendo que la interfaz de usuario se adapte a diferentes idiomas y culturas.
- **LoggingService:** Este servicio implementa la funcionalidad de registro de eventos y seguimiento del comportamiento de la aplicación. Permite la captura y el almacenamiento eficiente de los registros de la aplicación, facilitando la depuración, el seguimiento de errores y el análisis del uso de la aplicación.

Estos servicios son utilizados por los ViewModels para acceder a la lógica específica que proporcionan, promoviendo la reutilización de código y mejorando la mantenibilidad de la aplicación.

4. Funcionalidades principales

4.1. Autenticación

La funcionalidad de autenticación es el punto de entrada al sistema TPV, garantizando que solo los usuarios autorizados puedan acceder a sus diversas funciones. El proceso de autenticación permite determinar el rol del usuario que inicia sesión, diferenciando entre administradores y dependientes.

- **Inicio de sesión:** La aplicación presenta una interfaz de inicio de sesión donde el usuario debe ingresar sus credenciales (nombre de usuario y contraseña).
- **Roles de usuario:** El sistema reconoce diferentes roles de usuario con permisos específicos. Los administradores tienen acceso completo a todas las funcionalidades, incluyendo la gestión de productos, usuarios y reportes. Los dependientes, por otro lado, pueden tener un acceso más limitado, enfocado en la realización de ventas y otras operaciones del día a día.
- **Seguridad:** Se implementan medidas de seguridad para proteger las credenciales de los usuarios, como el almacenamiento seguro de contraseñas (por ejemplo, mediante hashing) y la prevención de accesos no autorizados.

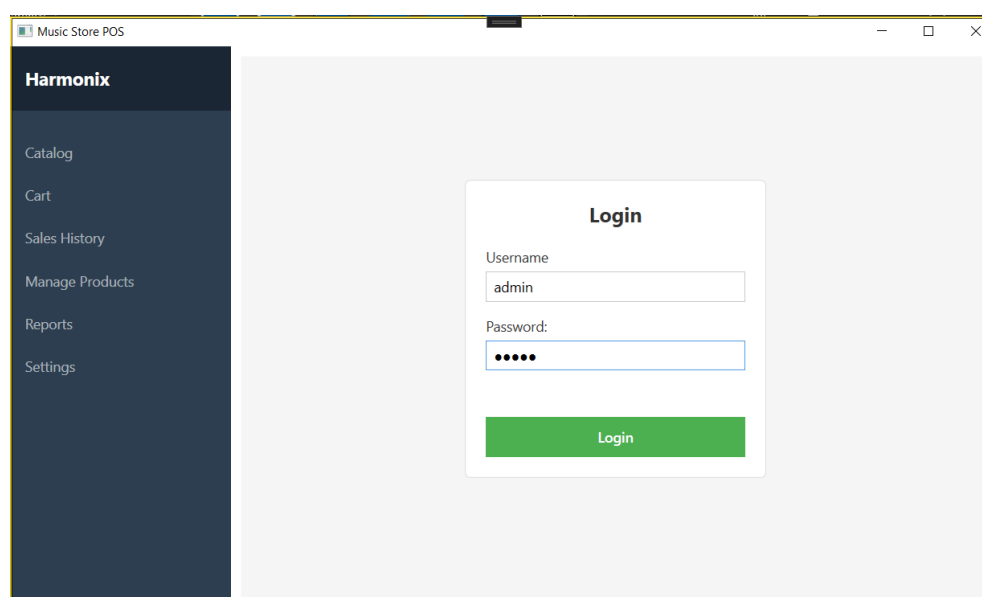


Ilustración 5. Vista de autenticación

4.2. Catálogo de productos

El catálogo de productos es una funcionalidad central del sistema TPV, que permite a los usuarios visualizar los artículos disponibles para la venta y añadirlos al carrito de compra.

- **Visualización de productos:** Los productos se presentan en un formato de cartas, similar al estilo de Bootstrap, mostrando información relevante como el nombre y el precio. No se incluyen imágenes en la visualización del catálogo.
- **Detalle y adición al carrito:** Cada producto en el catálogo ofrece opciones para ver los detalles del artículo y para agregarlo al carrito de compra.
- **Búsqueda en tiempo real:** El catálogo incluye una barra de búsqueda que permite a los usuarios filtrar los productos en tiempo real por su nombre. A medida que el usuario escribe en la barra de búsqueda, la lista de productos se actualiza automáticamente para mostrar solo los que coinciden con el texto ingresado.

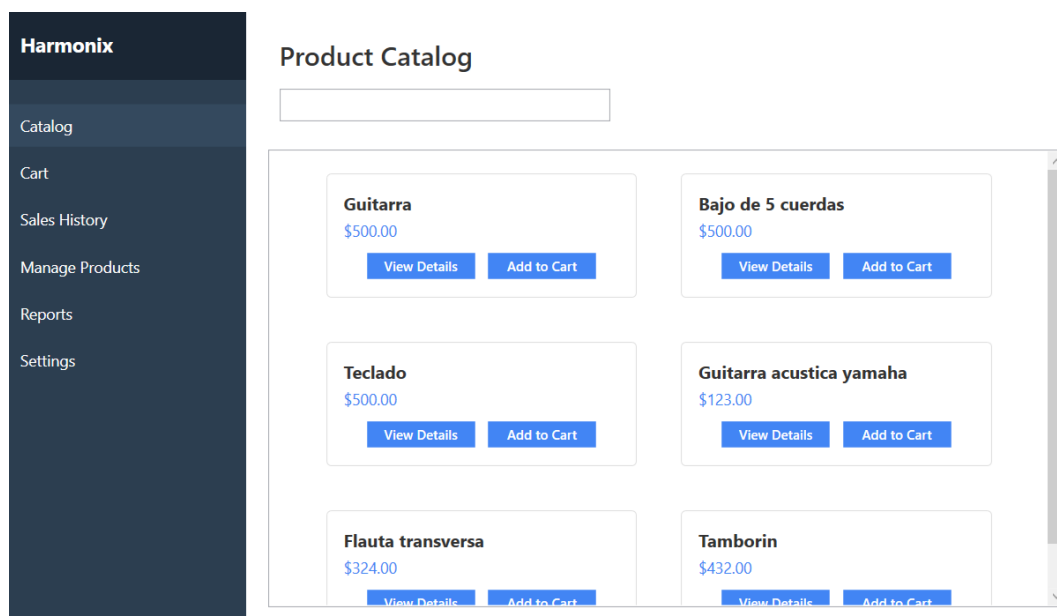


Ilustración 6. Vista de catálogo

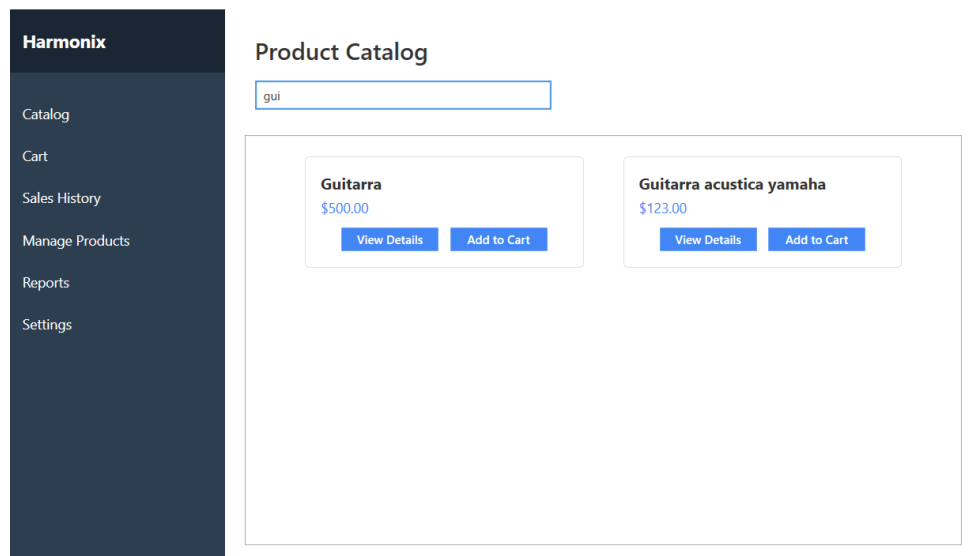


Ilustración 7. Ejemplo de búsqueda y filtrado

4.3. Carrito de compras

El carrito de compras permite a los usuarios gestionar los productos seleccionados antes de proceder al pago. Esta funcionalidad proporciona una visión clara de los artículos que se van a adquirir y permite realizar ajustes en la cantidad o eliminar productos.

- **Visualización de los artículos seleccionados:** El carrito muestra una lista de los productos que el usuario ha añadido, incluyendo información relevante como el nombre, el precio unitario y la cantidad seleccionada.
- **Modificación de la cantidad:** Los usuarios pueden modificar la cantidad de cada producto en el carrito, lo que actualiza automáticamente el subtotal del artículo y el total general de la compra.
- **Eliminación de productos:** Los usuarios pueden eliminar productos del carrito si cambian de opinión o desean ajustar su compra.
- **Cálculo del total:** El carrito calcula automáticamente el subtotal de cada artículo (precio unitario x cantidad) y el total general de la compra.
- **Modelo de datos transitorio:** Para implementar la funcionalidad del carrito, se utiliza un modelo de datos que no se guarda directamente en la base de datos. Este modelo transitorio permite gestionar la información del carrito de compra de forma eficiente

durante la sesión del usuario, sin necesidad de persistir los datos hasta el momento de la confirmación de la venta.

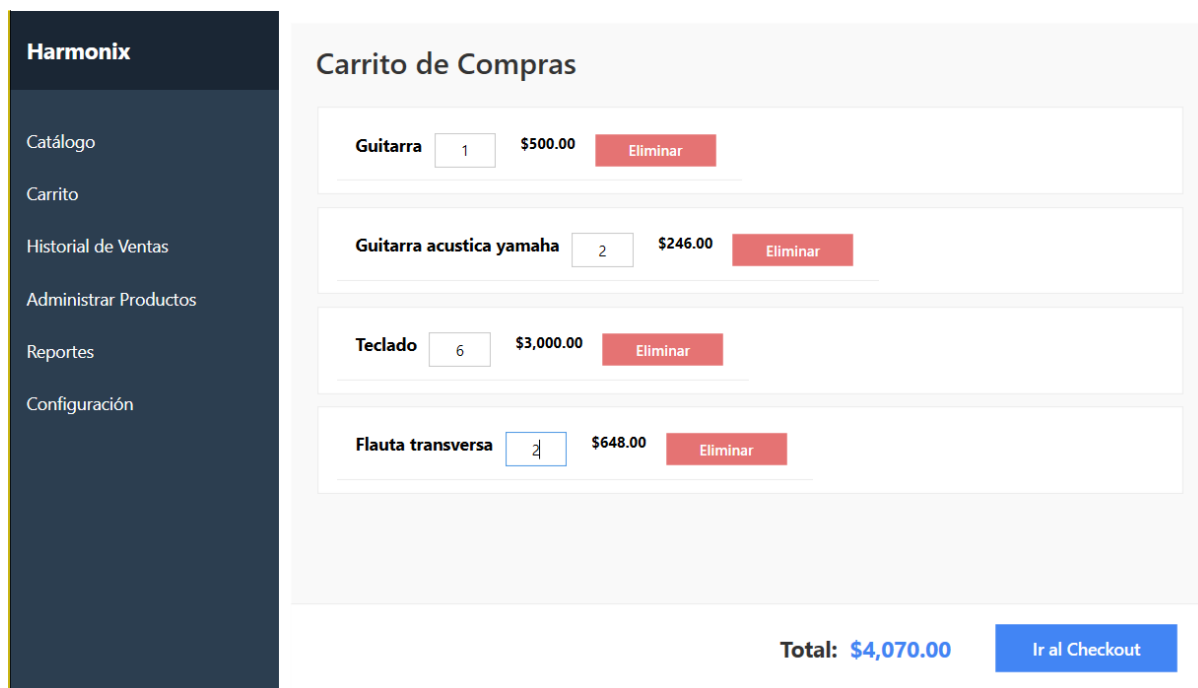


Ilustración 8. Vista de carrito

4.4. Proceso de pago

El proceso de pago se inicia una vez que el usuario ha revisado los artículos en el carrito de compras y está listo para finalizar la transacción. Esta funcionalidad guía al usuario a través de los pasos necesarios para completar la compra, incluyendo la recopilación de información del cliente y la selección del método de pago.

- **Información del cliente:** Antes de seleccionar el método de pago, el sistema solicita al usuario que ingrese información básica del cliente, como nombre, apellido y dirección. Estos datos son necesarios para registrar la venta y, potencialmente, para futuras comunicaciones o seguimiento.
- **Selección del método de pago:** El sistema ofrece una variedad de métodos de pago para adaptarse a las preferencias del cliente. En la implementación actual, se incluyen

opciones como "Efectivo", "Tarjeta de Crédito" y "Tarjeta de Débito". El usuario debe seleccionar el método de pago deseado para continuar con el proceso.

- **Finalización de la compra:** Una vez que se han proporcionado los datos del cliente y se ha seleccionado el método de pago, el usuario puede confirmar la compra. Al hacerlo, el sistema realiza las siguientes acciones:
 - **Crea un nuevo registro de cliente:** Se guarda la información del cliente en la base de datos.
 - **Crea un nuevo ticket de venta:** Se registra la venta en la base de datos, incluyendo los artículos comprados, las cantidades, los precios y el total. Se asocia este ticket al cliente registrado.
 - **Vacía el carrito de compras:** Los artículos del carrito se eliminan, preparándolo para la siguiente venta.
 - **Resetea el total:** El total del carrito se restablece a cero.
 - **Muestra un mensaje de confirmación:** Se informa al usuario que la compra se ha realizado con éxito.
 - **Navega de vuelta al catálogo:** Después de la compra, el sistema redirige al usuario a la vista del catálogo de productos.
- **Registro de la transacción:** Se utiliza el `LoggingService` para registrar información relevante sobre el proceso de pago, incluyendo el intento de finalización de la orden, el método de pago seleccionado y el resultado de la transacción (éxito o error).
- **Manejo de errores:** El sistema incluye un mecanismo de manejo de errores para informar al usuario si falta información importante (como los datos del cliente) o si ocurre algún problema durante el proceso de finalización de la compra.

Checkout

resumen del pedido

Guitarra x 1 - €500.00
Guitarra acustica yamaha x 2 - €246.00
Teclado x 6 - €3,000.00
Flauta transversa x 2 - €648.00

Datos del Cliente

Nombre:

Jose

Apellido:

Maitan

Dirección:

Alicante

Método de Pago:

Finalizar Compra

Ilustración 9. Vista de checkout

Datos del Cliente

Nombre:

Jose

Apellido:

Maitan


Dirección:

Alicante

Método de Pago:

Tarjeta de Débito

Confirmación

 ¡Compra realizada con éxito!

OK

Ilustración 10. Compra exitosa

4.5. Historial de ventas

La funcionalidad de historial de ventas permite a los usuarios (administradores o personal autorizado) consultar y revisar el historial de transacciones realizadas en el sistema TPV.

- **Visualización de la lista de ventas:** La aplicación muestra una lista de todos los tickets de venta registrados en la base de datos. Cada entrada en la lista representa una transacción individual y contiene información resumida como el ID del ticket, la fecha y hora de la venta, el cliente asociado y el total de la venta.
- **Carga de datos desde la base de datos:** La lista de ventas se carga dinámicamente desde la base de datos utilizando Entity Framework Core. Se recuperan los tickets de venta, incluyendo la información del cliente relacionado, para proporcionar un contexto completo de cada transacción.
- **Selección y detalle de venta:** Los usuarios pueden seleccionar una venta específica de la lista para ver los detalles completos de la transacción. Al seleccionar una venta, se muestra una vista detallada que incluye información sobre los artículos vendidos, las cantidades, los precios, el cliente y el método de pago.
- **Navegación a los detalles de la orden:** La funcionalidad permite navegar a la vista de detalles de una orden específica al seleccionar un ticket de venta en la lista. Esto facilita la revisión exhaustiva de cada transacción y la generación de facturas.
- **Exportación de facturas:** Desde la vista de detalles de la orden, los usuarios pueden exportar la factura correspondiente a un archivo PDF. El sistema genera un documento PDF con la información completa de la venta, incluyendo los datos del cliente, los artículos comprados, los totales y la información de la tienda.
- **Registro de eventos:** Se utiliza el `LoggingService` para registrar información relevante sobre la interacción con la lista de ventas, incluyendo la carga de la lista, la selección de ventas y la generación de facturas. Esto facilita el seguimiento del uso de la funcionalidad y la resolución de problemas.

Historial de Ventas				
1	4/30/2025 8:31 PM	Pedro	Perez	\$5,000.00
2	4/30/2025 8:56 PM	Jose	Jimenez	\$2,000.00
3	4/30/2025 8:59 PM	Luis	Maitan	\$1,000.00
4	4/30/2025 8:59 PM	Test	Tester	\$1,500.00
5	4/30/2025 9:22 PM	Test	Test	\$1,000.00
6	4/30/2025 9:58 PM	test	test	\$1,000.00
7	4/30/2025 11:05 PM	Vero	Rojas	\$5,500.00
8	5/1/2025 1:13 PM	Jose	Herrera	\$623.00
9	5/1/2025 9:06 PM	Humberto	Ramirez	\$1,000.00
10	5/1/2025 9:25 PM	John	Doe	\$1,447.00

Ilustración 11. Vista de historial de ventas

Detalles del Pedido

Información del Cliente

Nombre:: Jose
 Apellido:: Maitan
 Dirección:: Alicante
 Método de Pago:: Tarjeta de Débito

Guitarra x 1 **\$500.00**

Guitarra acustica yamaha x 2 **\$246.00**

Teclado x 6 **\$3,000.00**

Flauta transversa x 2 **\$648.00**

Total del Pedido: \$4,394.00

Exportar PDF

Ilustración 12. Detalle de venta pasada

Harmonix

ID de Ticket: 46
Fecha: 02/05/2025
Cliente: Jose Maitan
Dirección: Alicante
Método de pago: Tarjeta de Débito

Producto	Cantidad	Precio Unitario	Subtotal
Guitarra	1	€500.00	€500.00
Guitarra acustica yamaha	2	€123.00	€246.00
Teclado	6	€500.00	€3,000.00
Flauta transversa	2	€324.00	€648.00

Total: €4,394.00

¡Gracias por su compra!

Ilustración 13. Ejemplo de exportación a PDF

4.6. CRUD de productos

La funcionalidad de CRUD (Crear, Leer, Actualizar y Eliminar) de productos permite a los usuarios autorizados (administradores) gestionar el catálogo de productos de la tienda. Esta funcionalidad es esencial para mantener el inventario actualizado, agregar nuevos artículos, modificar la información de los existentes y eliminar los que ya no se ofrecen.

- **Visualización de la lista de productos:** La aplicación muestra una lista de todos los productos registrados en la base de datos. Cada entrada en la lista representa un producto individual y contiene información relevante como el nombre, precio y categoría.
- **Carga de datos desde la base de datos:** La lista de productos se carga dinámicamente desde la base de datos utilizando Entity Framework Core. Se recuperan los productos para su visualización y gestión.
- **Agregar un nuevo producto:** La funcionalidad permite a los usuarios agregar nuevos productos al catálogo. Al iniciar el proceso de creación de un nuevo producto, se presenta una vista o formulario que solicita la información necesaria del producto, como nombre, descripción, precio, categoría, etc.

- **Editar un producto existente:** La funcionalidad permite a los usuarios modificar la información de los productos existentes. Al seleccionar un producto de la lista, se puede acceder a una vista o formulario de edición que muestra los detalles actuales del producto y permite realizar cambios.
- **Eliminar un producto:** La funcionalidad permite a los usuarios eliminar productos del catálogo. Al seleccionar un producto de la lista, se puede iniciar el proceso de eliminación, que generalmente incluye una confirmación para evitar eliminaciones accidentales.
- **Validación de la selección:** La interfaz de usuario refleja visualmente si un producto está seleccionado, habilitando o deshabilitando los comandos de "Editar" y "Eliminar" según corresponda. Esto mejora la usabilidad y previene operaciones no válidas.
- **Comunicación entre vistas:** Se utilizan eventos (`RequestNewProductView`, `RequestEditProductView`, `RequestCloseView`) para coordinar la navegación y la comunicación entre la vista principal de administración de productos y las vistas de creación/edición de productos. Esto permite una separación clara de responsabilidades y un flujo de trabajo bien definido.
- **Registro de eventos:** Se utiliza el `LoggingService` para registrar información relevante sobre las operaciones de CRUD, incluyendo la carga de la lista de productos, la creación, edición y eliminación de productos, y cualquier error que pueda ocurrir. Esto facilita el seguimiento de los cambios en el catálogo y la resolución de problemas.

Administración de Productos

ID	Nombre	Descripción	Precio	Categoría
1	Guitarra	Guitarra Electrica Ibanez	\$500.00	Instrumentos
2	Bajo de 5 cuerdas	Bajo Electrico Fender	\$500.00	Instrumentos
3	Teclado	Teclado Roland	\$500.00	Instrumentos
5	Guitarra acustica yamaha	Guitarra de 6 cuerdas con cuerdas de nylon	\$123.00	Instrumentos
8	Flauta transversa	Flauta orquestal	\$324.00	Instrumentos
9	Tamborin	Tambor urbano tribal	\$432.00	Percusion

[Nuevo Producto](#)[Editar Producto](#)[Eliminar Producto](#)

Ilustración 14. Vista de administración de productos

Guitarra

Precio: **\$500.00**

Categoría: Instrumentos

Descripción:

Guitarra Electrica Ibanez

[Añadir al carrito](#)

Ilustración 15. Vista de detalle de producto

Detalles del Producto

Nombre:	<input type="text" value="Pedalera Ibanez TS-808"/>
Descripción:	<input type="text" value="Pedalera para distorsión de guitarra eléctrica"/>
Precio:	<input type="text" value="80.00"/>
Categoría:	<input type="text" value="Accesorios"/>

Guardar

Cancelar

Ilustración 16. Vista de creación/edición de productos

Administración de Productos

ID	Nombre	Descripción	Precio	Categoría
1	Guitarra	Guitarra Electrica Ibanez	\$500.00	Instrumentos
2	Bajo de 5 cuerdas	Bajo Electrico Fender	\$500.00	Instrumentos
3	Teclado	Teclado Roland	\$500.00	Instrumentos
5	Guitarra acustica yamaha	Guitarra de 6 cuerdas con cuerdas de nylon	\$123.00	Instrumentos
8	Flauta transversa	Flauta orquestal	\$324.00	Instrumentos
9	Tamborin	Tambor urbano tribal	\$432.00	Percusion
10	Pedalera Ibanez TS-808	Pedalera para distorsión de guitarra eléctrica	\$80.00	Accesorios

Ilustración 17. Ejemplo de creación de producto

ID	Nombre	Descripción	Precio	Categoría
1	Guitarra	Guitarra Electrica Ibanez	\$500.00	Instrumentos
2	Bajo de 5 cuerdas	Bajo Electrico Fender	\$500.00	Instrumentos
3	Teclado	Teclado Roland	\$500.00	Instrumentos
5	Guitarra acustica yamaha	Guitarra de 6 cuerdas con cuerdas de nylon	\$123.00	Instrumentos
8	Flauta transversa	Flauta orquestal	\$324.00	Instrumentos
9	Tamborin	Tambor urbano tribal	\$432.00	Percusion
10	Pedalera Ibanez TS-808	Pedalera para distorsión de guitarra eléctrica	\$80.00	Accesorios

Nuevo Producto
Editar Producto
Eliminar Producto

Ilustración 18. Botones para operaciones CRUD

4.7. Estadísticas y Reportes

La funcionalidad de estadísticas y reportes proporciona a los usuarios autorizados (administradores) una visión general del rendimiento del negocio, presentando datos clave sobre ventas, ingresos y productos. Esta funcionalidad es crucial para la toma de decisiones informadas y la planificación estratégica.

- **Estadísticas generales:** La aplicación muestra estadísticas clave sobre el negocio, incluyendo:
 - **Total de productos:** El número total de productos registrados en el sistema.
 - **Total de ventas:** El número total de transacciones de venta realizadas.
 - **Ingresos totales:** El valor monetario total de todas las ventas.
- **Ingresos mensuales:** La aplicación calcula y muestra los ingresos generados durante el mes en curso, lo que permite a los usuarios evaluar el rendimiento reciente del negocio.
- **Cálculo de datos:** Los datos para los reportes se calculan dinámicamente consultando la base de datos utilizando Entity Framework Core. Se realizan consultas para contar productos, contar ventas y sumar los totales de las ventas.

- **Registro de eventos:** Se utiliza el `LoggingService` para registrar información relevante sobre la carga de los datos de los reportes, lo que facilita el seguimiento del uso de la funcionalidad y la resolución de problemas.

Informes y Estadísticas

Cantidad Total de Productos:	7
Número Total de Ventas:	46
Ganancia Total Histórica:	\$63,744.00
Ganancia del Mes Actual:	\$46,744.00

Ilustración 19. Vista de estadísticas y reportes

4.8. Internacionalización

La aplicación ha sido diseñada para ser accesible a usuarios de diferentes partes del mundo, lo que se logra mediante la capacidad de cambiar el idioma de la interfaz.

- **Soporte de múltiples idiomas:** El sistema no se limita a un único idioma, sino que ofrece la posibilidad de elegir entre varias opciones. En este caso, se incluyen el español y el inglés, pero la arquitectura permite añadir fácilmente otros idiomas en el futuro.
- **Selección de idioma por el usuario:** Para brindar una experiencia personalizada, la aplicación permite que cada usuario elija su idioma preferido. Esta elección se realiza a través de un menú de configuración, donde se presenta una lista clara de los idiomas disponibles.
- **Componente de localización:** El corazón de esta funcionalidad es un componente que gestiona la adaptación del texto a cada idioma.

- **Gestión del idioma activo:** Este componente mantiene un registro del idioma que el usuario ha seleccionado en cada momento.
- **Acceso a las traducciones:** Utiliza un sistema de almacenamiento organizado para acceder a las traducciones de todas las palabras y frases que aparecen en la pantalla. Esto asegura que los botones, etiquetas, mensajes y cualquier otro texto se muestren correctamente en el idioma elegido.
- **Actualización automática:** Cuando el usuario cambia el idioma, este componente se encarga de notificar a la interfaz para que se actualice automáticamente, mostrando todo el texto en la nueva lengua seleccionada.
- **Adaptación de la interfaz:** Proporciona las traducciones necesarias para que la aplicación se muestre correctamente en el idioma seleccionado por el usuario.
- **Guardado de la preferencia:** Para mayor comodidad, la aplicación recuerda la elección de idioma del usuario. De esta manera, cada vez que se inicie el programa, se mostrará automáticamente en el idioma preferido, sin necesidad de configurarlo nuevamente.
- **Aplicación del idioma:** Al iniciar la aplicación o al cambiar la configuración, el sistema se asegura de que todas las partes del programa utilicen el idioma correcto, no solo para el texto, sino también para otros elementos que dependen de la cultura, como las fechas y los números.
- **Manejo de errores:** El sistema está preparado para el caso en que, por algún motivo, el idioma seleccionado por el usuario no esté disponible. En esta situación, la aplicación mostrará un idioma por defecto (en este caso, español) y registrará el error para que pueda ser corregido.
- **Registro de eventos:** Se lleva un registro detallado de todas las acciones relacionadas con la configuración del idioma, lo que facilita la identificación y solución de posibles problemas.

Order Details

Customer Information

First Name:: Jose

Last Name:: Maitan

Address:: Alicante

Payment Method:: Tarjeta de Débito

Guitarra x 1 \$500.00

Guitarra acustica yamaha x 2 \$246.00

Teclado x 6 \$3,000.00

Flauta transversa x 2 \$648.00

Order Total: \$4,394.00

Export PDF

Ilustración 20. Ejemplo de cambio de idioma en UI

4.9. Persistencia de Interfaz

La persistencia de la interfaz se refiere a la capacidad de la aplicación para recordar las preferencias del usuario en cuanto a su apariencia y comportamiento, de modo que se mantengan de una sesión a otra. En el contexto del código proporcionado, esta funcionalidad se centra en recordar el idioma preferido.

- **Guardado de la preferencia:** La aplicación guarda la elección del idioma del usuario para que no tenga que seleccionarlo cada vez que la inicia. Esta información se almacena de forma que se conserve incluso al cerrar y volver a abrir el programa.
- **Carga de la preferencia:** Al iniciarse, la aplicación consulta si el usuario ha seleccionado previamente un idioma. Si es así, carga esta información y la utiliza para mostrar la interfaz en el idioma elegido. Si no hay ninguna preferencia guardada, se utiliza un idioma por defecto.
- **Aplicación de la preferencia:** El idioma cargado se utiliza para configurar el sistema de localización, asegurando que la aplicación se muestre desde el principio en el idioma deseado por el usuario.

Es importante destacar que el concepto de persistencia de la interfaz puede extenderse a otras opciones de configuración, como el tamaño y la posición de las ventanas, la disposición de los elementos en la pantalla, los colores utilizados, etc. En general, se trata de ofrecer al usuario la posibilidad de personalizar la aplicación y que estas personalizaciones se mantengan a lo largo del tiempo.

5. Conclusiones

El desarrollo de este sistema de punto de venta ha permitido alcanzar una comprensión profunda de los desafíos y las soluciones involucradas en la creación de una aplicación de escritorio moderna y funcional. Se ha logrado implementar un conjunto de funcionalidades esenciales que abarcan la gestión de productos, el procesamiento de ventas, la generación de reportes y la configuración de la aplicación.

Sin embargo, debido a las limitaciones de tiempo, algunas de las funcionalidades propuestas inicialmente no pudieron ser implementadas en su totalidad. Entre ellas se encuentran:

- **Mejoras en la interfaz de usuario:** Si bien se ha desarrollado una interfaz funcional, existen áreas en las que se podría mejorar la experiencia del usuario, haciéndola más intuitiva y visualmente atractiva.
- **Integración con pasarelas de pago:** La aplicación actualmente permite registrar ventas, pero no se ha integrado con sistemas de pago electrónicos para automatizar el proceso de transacción.
- **Gestión de inventario avanzada:** Se ha implementado una gestión básica de productos, pero se podría ampliar para incluir funcionalidades más sofisticadas como el seguimiento del stock en tiempo real, alertas de bajo inventario y la gestión de proveedores.
- **Reseñas y valoraciones de productos:** La funcionalidad para que los clientes puedan dejar comentarios y valoraciones sobre los productos no se ha implementado.

- **Opciones de configuración adicionales:** Se podrían añadir más opciones para que los usuarios personalicen la aplicación según sus necesidades, como la configuración de impresoras de tickets o la gestión de descuentos.
- **Estadísticas y reportes más detallados:** Si bien se generan reportes de ventas, se podrían incluir análisis más profundos sobre el rendimiento del negocio, como la identificación de los productos más vendidos o el análisis de las tendencias de ventas a lo largo del tiempo.

Para mejorar la aplicación en el futuro, se podrían considerar las siguientes acciones:

- **Refactorización del código:** Revisar y reorganizar el código para mejorar su legibilidad, mantenibilidad y rendimiento.
- **Adopción de patrones de diseño:** Utilizar patrones de diseño establecidos para resolver problemas comunes de desarrollo y mejorar la estructura de la aplicación.
- **Exploración de nuevas tecnologías:** Investigar y utilizar las últimas tecnologías y herramientas de desarrollo de WPF para aprovechar al máximo las capacidades de la plataforma.

En resumen, se ha desarrollado una base sólida para un sistema de punto de venta, pero aún existen oportunidades para expandir su funcionalidad y mejorar su calidad. El trabajo futuro podría centrarse en abordar las áreas mencionadas anteriormente para crear una aplicación aún más completa y robusta.