

# NovaMarket



8 JUNIO

---

**I.E.S Politécnico Hermenegildo Lanz**

**Creado por:** *Santiago Gallardo Martín*

**Ciclo Formativo:** *Grado Superior en Desarrollo de Aplicaciones  
Multiplataforma*

**Curso:** *2024/2025*

**Grupo:** *2º DAM Dual*



## Índice

1.	Introducción .....	3
2.	Justificación del proyecto .....	3
3.	Aspectos generales.....	3
4.	Contenidos .....	5
4.1.	Acceso a datos .....	5
4.2.	Base de datos .....	5
4.3.	Despliegue .....	5
5.	Desarrollo de implementación .....	6
5.1.	Backend (PHP + MySQL).....	6
5.2.	Aplicación de escritorio (JavaFX) .....	6
5.3.	Aplicación móvil (Android) .....	6
6.	Pruebas e incidencias .....	7
6.1.	Información general .....	7
6.2.	Detalles de la prueba .....	7
6.3.	Resumen de errores y problemas encontrados .....	9
6.4.	Resultados de las pruebas .....	9
7.	Manual de usuario .....	10
7.1.	Aplicación de escritorio .....	10
7.2.	Aplicación móvil.....	12
7.3.	Recomendaciones .....	14
8.	Conclusión .....	15
9.	Referencias .....	15

## 1. Introducción

Este proyecto de fin de curso tienen como objetivo el desarrollo de un sistema de gestión de ventas en tiempo real compuesto por dos aplicaciones: una aplicación de escritorio desarrollada en JavaFX con la función de administrar las ventas y el poder realizar pedidos en una tienda física con la ventana donde se utiliza el lector de código de barras y una aplicación móvil Android dirigida a los clientes para poder realizar pedidos de forma sencilla.

## 2. Justificación del proyecto

El proyecto surge con la intención de simular el funcionamiento de un sistema de ventas similar al que se encuentra en supermercados u otros comercios, donde se utilizan lectores de códigos de barras para registrar productos y generar ventas. La aplicación permite realizar ventas escaneando productos y gestionarlas desde la misma interfaz, proporcionando una experiencia ágil y moderna.

Además, la posibilidad de realizar estas operaciones desde un dispositivo móvil y que se reflejen en tiempo real en la aplicación de escritorio hace que sea una solución muy versátil para pequeñas empresas, permitiendo una sincronización eficaz y evitando errores o duplicidades en el registro de pedidos. Esta funcionalidad facilita la gestión centralizada de la actividad comercial, mejorando la eficiencia operativa.

## 3. Aspectos generales

El sistema esta compuesto por:



**JavaFX y SceneBuilder →**

Para la aplicación de escritorio se ha hecho uso de JavaFX con el IDE NetBeans y SceneBuilder para el diseño de las ventanas



**Kotlin en AndroidStudio →**

Para la aplicación móvil se ha hecho uso de Kotlin con el IDE AndroidStudio.



**Node.js →**

Para hacer el servidor de WebSocket se ha utilizado Node.js.



**AWS y Docker →**

Se ha usado AWS para hostear el contenedor Docker (LAMP) el cual contiene la base de datos y la API con la que se trabaja.

**InstallBuilder →**

Para que los usuarios puedan usar la aplicación de escritorio se ha hecho uso de InstallBuilder que permite generar el .exe para instalar la aplicación en el dispositivo.

### **Funcionamiento:**

#### **Aplicación de escritorio:**

La aplicación de escritorio tiene la opción de añadir ventas a través del sistema de venta que puedes ver en cualquier supermercado leyendo los códigos de barras y como el panel central de operaciones. Desde ella, se puede visualizar en tiempo real cada venta que se registra desde los dispositivos móviles. La interfaz principal muestra una tabla con las ventas, detallando el nombre del producto, la cantidad, el precio unitario y el total de la operación.

Además de la visualización de ventas, la aplicación permite la gestión completa del sistema: se pueden añadir, editar o eliminar productos y usuarios, así como ver las sesiones que ha tenido el usuario.

La conexión con la base de datos se realiza mediante JDBC, garantizando una integración directa y segura con MySQL.

#### **Aplicación de móvil:**

La aplicación móvil está pensada para ser utilizada por los clientes que registran ventas directamente. Incluye una pantalla de login y registro para autenticación del usuario, y una sección donde se puede seleccionar el producto y añadirlo a la cesta, un apartado para poder editar el perfil y la página de inicio que muestra información general de la empresa.

#### **Ambas aplicaciones:**

Las actualizaciones en tiempo real se reciben a través de un servidor WebSocket externo, que comunica los eventos generados en la aplicación móvil con la aplicación de escritorio y viceversa.

Cada vez que se realiza una venta desde el móvil o desde la misma aplicación de escritorio, el servidor emite un mensaje que se recibe inmediatamente en la aplicación, actualizando automáticamente.

## 4. Contenidos

### 4.1. Acceso a datos

El acceso a los datos en la aplicación de móvil se hace mediante la API. Se utilizan claves primarias, claves foráneas y restricciones para asegurar la integridad referencial de la información. La lógica de negocio accede a las tablas para realizar operaciones como:

- Consultas de productos disponibles.
- Registro y autenticación de usuarios.
- Registro de pedidos y sus productos.
- Gestión de códigos de barras vinculados a productos.

En el backend de la aplicación de escritorio se utilizan controladores o DAOs que encapsulan las operaciones sobre la base de datos.

### 4.2. Base de datos

La base de datos del sistema se llama `sistema_tienda` y está compuesta por las siguientes tablas principales:

- **usuarios**: Contiene información de los usuarios del sistema (clientes y empleados), como nombre, correo electrónico, rol y estado de actividad.
- **sesiones**: Registra las sesiones activas de los usuarios, incluyendo dispositivo, fecha de inicio y última actividad.
- **productos**: Registra todos los productos disponibles en la tienda, incluyendo nombre, precio, código de barras, imagen en base64 y stock.
- **codigos\_barras**: Almacena los códigos generados, la fecha de generación y el usuario que los generó. Se relaciona con usuarios.
- **ventas**: Almacena las ventas realizadas, con sus respectivos clientes y empleados. Incluye información sobre el método de pago, tipo de venta y estado.
- **detalle\_venta**: Registra los productos vendidos en cada venta, su cantidad, precio unitario y subtotal. Se relaciona con ventas y productos.

Se han definido relaciones entre tablas utilizando claves foráneas, así como restricciones ON DELETE CASCADE o SET NULL según el caso, para mantener la coherencia al eliminar registros.

### 4.3. Despliegue

El sistema está en un servidor remoto. Las recomendaciones para el despliegue son:

#### Entorno LAMP

- **Servidor web**: Apache.
- **Servidor de base de datos**: MariaDB.
- **Lenguaje backend**: PHP.
- **phpMyAdmin**: Para administración visual de la base de datos.

**Pasos básicos:**

1. **Importación de base de datos:** A través de phpMyAdmin, importar el archivo .sql para crear las tablas y restricciones.
2. **Configuración del entorno:** Definir correctamente las variables de conexión (host, usuario, contraseña, nombre\_bd) en el archivo de configuración del backend.
3. **Seguridad:** Las contraseñas guardadas en la base de datos están cifradas.
4. **Integración con otros sistemas:** El sistema está integrado con WebSocket para actualizaciones en tiempo real.

## 5. Desarrollo de implementación

El desarrollo se ha dividido en los siguientes módulos:

### 5.1. Backend (PHP + MySQL)

- **Estructura REST API** desarrollada con PHP, conectada a la base de datos sistema\_tienda.
- **Autenticación y autorización de usuarios** mediante verificación con la API.
- **API para ventas, productos, usuarios, sesiones y los detalles para la venta.**
- **Servidor WebSocket en Node.js** con conexión directa a la base de datos para emitir cambios en tiempo real.

### 5.2. Aplicación de escritorio (JavaFX)

- Interfaz intuitiva que permite consultar productos, registrar ventas y gestionar usuarios.
- La tabla de ventas se actualiza automáticamente mediante conexión WebSocket.
- Utiliza Properties y Bindings para una gestión dinámica de los elementos de la interfaz.

### 5.3. Aplicación móvil (Android)

- Desarrollada en Kotlin, incorpora fragmentos como LoginFragment, HomeFragment y PedidosFragment.
- Se usa OkHttpClient para comunicarse con el backend PHP.
- WebSocket en segundo plano para actualizar pedidos.

## 6. Pruebas e incidencias

### 6.1. Información general

- **Nombre de la aplicación (escritorio):** NovaMarket Escritorio
- **Nombre de la aplicación (móvil):** NovaMarket Móvil
- **Versión de ambas aplicaciones:** 1.0
- **Fecha de inicio de pruebas:** 29/05/2025
- **Hora de inicio:** 17:15
- **Hora de finalización:** 17:40

### Objetivos de la prueba

#### Aplicación de escritorio (JavaFX):

1. Visualización en tiempo real de una venta generada desde móvil.
2. Registro de nuevo producto.
3. Modificación del stock de un producto existente.
4. Eliminación de un producto.
5. Consulta del historial de ventas.
6. Visualización de sesiones activas.
7. Gestión de usuarios.

#### Aplicación móvil (Android):

1. Registro de usuario.
2. Inicio de sesión.
3. Escaneo de producto (simulado).
4. Envío de venta a servidor.
5. Recepción de respuesta del backend y confirmación.
6. Visualización del pedido realizado.
7. Sincronización con app de escritorio (evento WebSocket).

### 6.2. Detalles de la prueba

#### Caso de prueba 1 (Escritorio): Visualización de venta desde móvil

- **Pasos:**
  1. Ejecutar NovaMarket Escritorio.
  2. Iniciar sesión como administrador.
  3. Mantener la vista de ventas abierta.
  4. Realizar una venta desde el móvil.
- **Resultado esperado:** La venta aparece automáticamente en la tabla.
- **Resultado obtenido:** Correcto.
- **Estado:** Cerrado.

### Caso de prueba 2 (Escritorio): Registro de nuevo producto

- **Pasos:**
    1. Acceder a la pestaña "ADMIN".
    2. Entrar en el apartado "Productos".
    3. Pulsar "Insertar", rellenar los campos y guardar.
  - **Resultado esperado:** Producto registrado y visible en la tabla.
  - **Resultado obtenido:** Correcto.
  - **Estado:** Cerrado.
- 

### Caso de prueba 3 (Móvil): Registro de usuario

- **Pasos:**
    1. Abrir NovaMarket Móvil.
    2. Ir a "Registrarse".
    3. Introducir los datos y enviar.
  - **Resultado esperado:** Usuario registrado en el sistema.
  - **Resultado obtenido:** Correcto.
  - **Estado:** Cerrado.
- 

### Caso de prueba 4 (Móvil): Envío de venta

- **Pasos:**
    1. Iniciar sesión en la app.
    2. Añadir un producto al carrito.
    3. Confirmar pedido.
  - **Resultado esperado:** Venta procesada, confirmación mostrada.
  - **Resultado obtenido:** Correcto.
  - **Estado:** Cerrado.
- 

### Caso de prueba 5 (Ambas apps): Sincronización por WebSocket

- **Pasos:**
    1. Con ambas apps abiertas, realizar venta desde el móvil.
    2. Observar si aparece instantáneamente en la tabla ventas de la app de escritorio.
  - **Resultado esperado:** Venta sincronizada sin recargar.
  - **Resultado obtenido:** Correcto.
  - **Estado:** Cerrado.
-



#### Caso de prueba 6 (Escritorio): Eliminación de un producto

- **Pasos:**
    1. Seleccionar producto en la tabla.
    2. Pulsar “Eliminar”.
  - **Resultado esperado:** Producto desaparece de la tabla y la base de datos.
  - **Resultado obtenido:** Correcto.
  - **Estado:** Cerrado.
- 

#### Caso de prueba 7 (Móvil): Inicio de sesión

- **Pasos:**
  1. Iniciar la app.
  2. Introducir credenciales válidas.
  3. Pulsar “Iniciar sesión”.
- **Resultado esperado:** Acceso concedido, se muestra pantalla principal.
- **Resultado obtenido:** Correcto.
- **Estado:** Cerrado.

### 6.3. Resumen de errores y problemas encontrados

No se han encontrado errores críticos.


### 6.4. Resultados de las pruebas

- **Número total de pruebas realizadas:** 7
- **Errores encontrados:** 0
- **Experiencia del usuario:** Fluida, clara y sin bloqueos. Las actualizaciones en tiempo real fueron especialmente valoradas. Se consideró añadir una animación de carga para la tabla de ventas.

## 7. Manual de usuario

### 7.1. Aplicación de escritorio

1. **Inicio de sesión:** Introduce tu correo y contraseña. Si son correctos, accederás al panel principal.



Iniciar sesión:

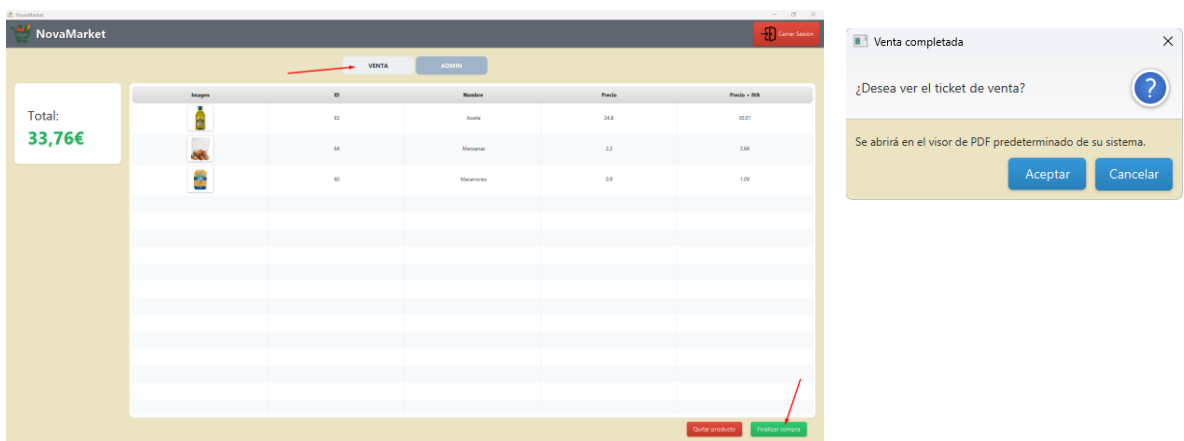
Usuario:

Contraseña:

Iniciar sesión

2. **Registro de ventas:** En la pestaña “VENTA” se pueden registrar ventas utilizando el lector de códigos de barras.

Cuando se hayan escaneado los productos del cliente y quieres finalizar la venta tienes que pulsar en el botón inferior “Finalizar compra”, le aparecerá una ventana informándole de que se ha abierto en el navegador una ventana para poder visualizar el ticket y desde ahí podrá imprimirlo.



NovaMarket

VENTA

Total: 33,76€

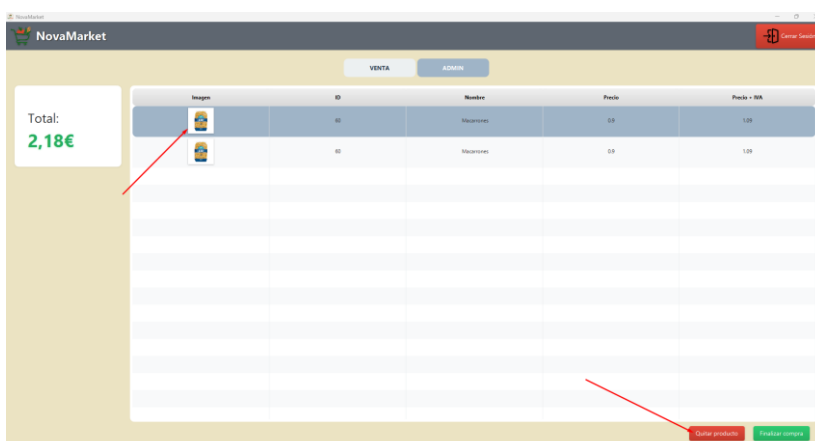
Imagen	ID	Nombre	Precio	Precio + IVA
	01	Aceite	24.0	26.01
	04	Manzanas	2.2	2.46
	05	Manzanas	0.9	1.09

¿Desea ver el ticket de venta?

Se abrirá en el visor de PDF predeterminado de su sistema.

Aceptar Cancelar

Si quiere eliminar un producto escaneado simplemente tiene que pulsar el producto escaneado que quiera eliminar de la tabla y posteriormente pulsar en el botón “Eliminar” y el producto se eliminara correctamente.



NovaMarket

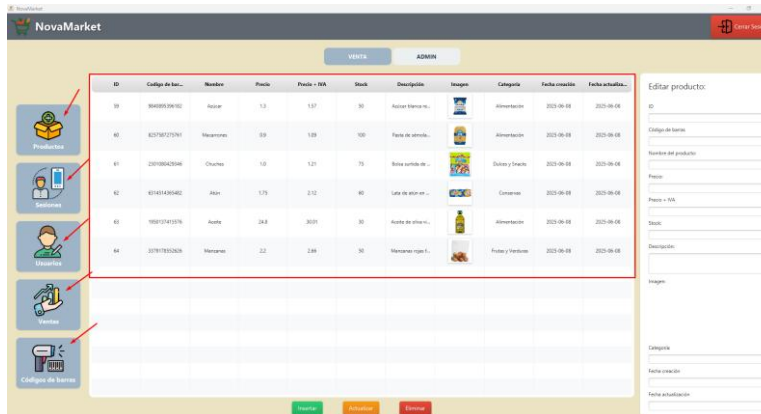
VENTA

Total: 2,18€

Imagen	ID	Nombre	Precio	Precio + IVA
	05	Manzanas	0.9	1.09
	05	Manzanas	0.9	1.09

Eliminar producto Finalizar compra

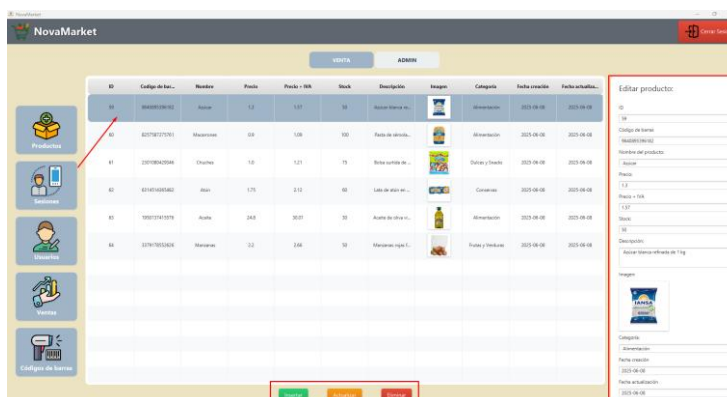
3. **Administración de ventas productos y usuarios:** En la pestaña “ADMIN”, solo los usuarios con los permisos adecuados pueden acceder. Una vez dentro, basta con pulsar sobre la sección que se desea administrar (productos, ventas, usuarios, etc.) y la tabla se rellenará automáticamente con la información correspondiente.



**Insertar:** Para insertar una venta, producto o usuario, hay que pulsar el botón “Insertar”, situado en la parte inferior central. Al hacerlo, se añadirá un nuevo ítem a la tabla. Si haces clic sobre ese ítem, podrás ver y editar su información en el panel derecho. Si modificas algún campo, para guardar los cambios debes pulsar el botón “Actualizar”.

**Actualizar:** Al igual que en el caso anterior, para actualizar un ítem debes seleccionarlo en la tabla. Su información aparecerá a la derecha. Modifica los campos que necesites (respetando las restricciones establecidas) y pulsa el botón “Actualizar”. Verás cómo el ítem se actualiza al instante.

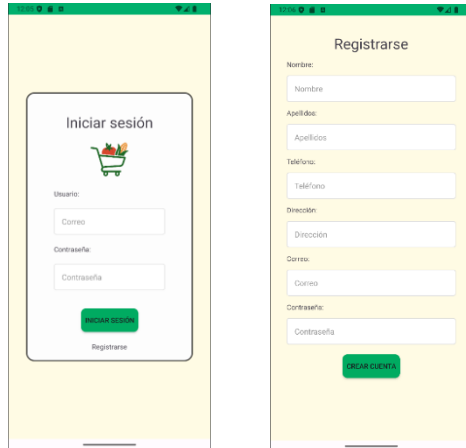
**Eliminar:** Para eliminar un ítem, selecciónalo en la tabla y pulsa el botón “Eliminar”, situado en la parte inferior central. En el caso de los usuarios, también se eliminarán automáticamente todas las sesiones asociadas.



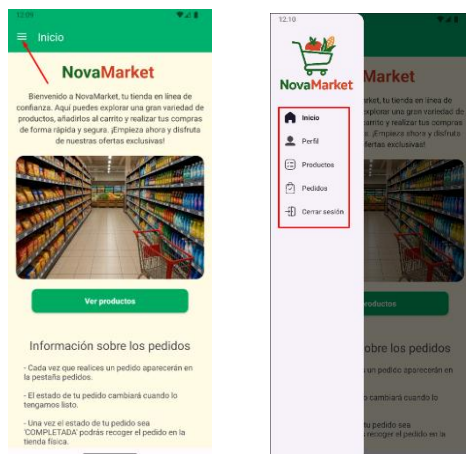
**Importante:** No podrás eliminar un producto que esté vinculado a una venta. Es decir, si una venta contiene ese producto, su eliminación no estará permitida para mantener la integridad de los datos.

## 7.2. Aplicación móvil

En la aplicación móvil puedes crear un usuario pulsando el botón **“Registrar”**. Si ya tienes una cuenta creada, pulsa el botón **“Iniciar sesión”** y, si has introducido correctamente tus datos, accederás al interior de la app.



Una vez dentro, verás una pequeña introducción con información sobre la empresa. Para navegar entre las distintas secciones, pulsa el botón situado en la parte superior izquierda; se desplegará un menú con varias opciones disponibles.



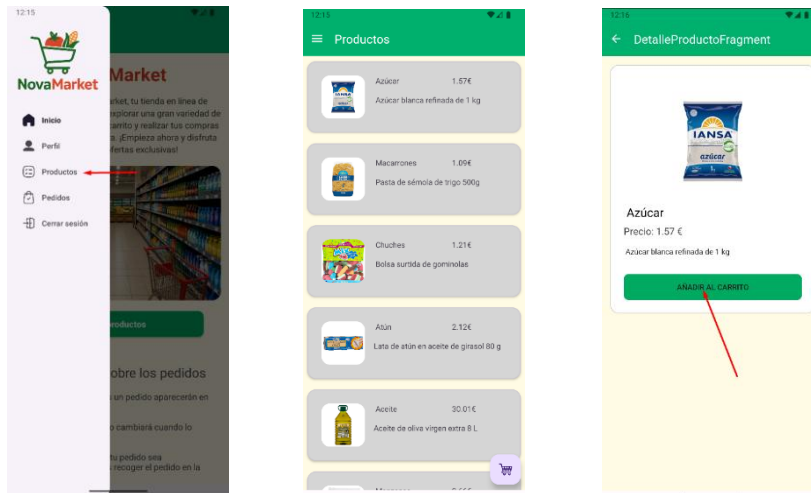
Desde la app puedes:

- Consultar el catálogo de productos.
- Ver el estado de los pedidos que hayas realizado.
- Visualizar y editar la información de tu cuenta personal.

## Crear un pedido

Para crear un pedido:

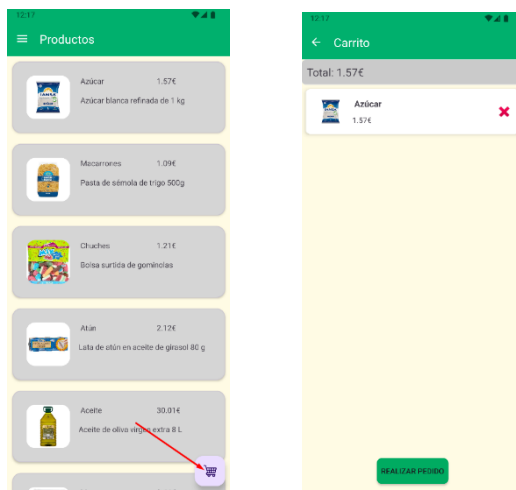
1. Accede a la pestaña **Productos**.
2. Pulsa sobre el producto que deseas añadir al carrito.
3. Se abrirá una nueva pantalla con más información del producto y un botón “**Añadir al carrito**”.
4. Al pulsar ese botón, el producto se añadirá al carrito.
5. Para volver a la lista de productos, pulsa la flecha situada en la parte superior izquierda.



## Gestionar el carrito

Para acceder al carrito, debes estar en la pestaña de productos. En la parte inferior izquierda encontrarás un botón con el icono de un carrito de la compra. Al pulsarlo, podrás:

- Ver los productos añadidos.
- Eliminar cualquier producto pulsando en la cruz correspondiente.
- Finalizar el pedido pulsando el botón “**Finalizar pedido**”.



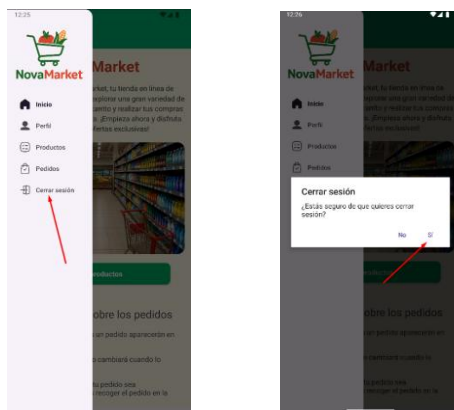
## Cancelar un pedido

Puedes cancelar un pedido desde la pestaña **“Pedidos”**, siempre y cuando su estado sea **“EN PROCESO”**. Para ello, pulsa el botón rojo **“Cancelar pedido”** y el pedido se cancelará correctamente.



## Cerrar sesión

Para cerrar sesión, abre el menú de navegación y pulsa la opción **“Cerrar sesión”**. Serás redirigido automáticamente a la pantalla de inicio de sesión.



## 7.3. Recomendaciones

- Mantén conexión a internet estable: Tanto la aplicación de escritorio como la aplicación móvil requieren conexión a internet para realizar operaciones con el servidor, enviar o recibir ventas y sincronizar datos en tiempo real.
- Utiliza credenciales seguras: Se recomienda registrar usuarios con contraseñas robustas y no compartirlas con otros usuarios. Cambia tu contraseña periódicamente desde la sección de perfil.
- Actualiza siempre la información antes de salir: Si has editado productos, usuarios o ventas, asegúrate de pulsar el botón **“Actualizar”** antes de salir del apartado correspondiente para evitar perder los cambios.
- Verifica los datos antes de finalizar una venta: Antes de confirmar una compra, revisa los productos añadidos al carrito, el total y los datos del usuario.

## 8. Conclusión

El desarrollo del proyecto NovaMarket ha supuesto una experiencia completa en el diseño, implementación y despliegue de un sistema multiplataforma de gestión de ventas en tiempo real. Se ha conseguido integrar con éxito una aplicación de escritorio en JavaFX y una aplicación móvil en Android, comunicadas mediante WebSocket y respaldadas por un backend PHP alojado en un contenedor Docker.

La aplicación de escritorio cumple su función como panel central de administración, permitiendo la gestión de usuarios, productos y ventas con una interfaz amigable y actualizaciones automáticas. Por su parte, la aplicación móvil ofrece a los clientes una forma cómoda y moderna de realizar pedidos desde cualquier lugar.

Durante las pruebas, el sistema ha demostrado ser estable, ágil y eficaz, cumpliendo todos los objetivos planteados inicialmente. La sincronización en tiempo real ha sido especialmente destacada por su utilidad en contextos comerciales reales. Este proyecto ha servido como base sólida para aplicar conocimientos de programación, diseño de bases de datos, arquitectura cliente-servidor, comunicaciones en tiempo real y buenas prácticas de desarrollo.

## 9. Referencias

Docker Hub → <https://hub.docker.com/>

ChatGPT → <https://chatgpt.com/>

DeepSeek → <https://www.deepseek.com/>