



# **UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

Unidad Académica Profesional Tianguistenco

## **Base de datos II**

**Mtra.** Maritza Flores Dominguez

### **Integrantes:**

JESÚS FABRICIO SÁNCHEZ MARTÍNEZ

JESUS SANTANA HERNANDEZ

ANGEL DAVID ARRIAGA GONZALEZ

JOEL EMETERIO MORENO

2024-A



## Documentación del Proyecto: Tiendita

### 1. Descripción General del Proyecto

**Tiendita** es una aplicación Java para la gestión de una pequeña tienda. Permite realizar operaciones básicas como agregar y eliminar productos, gestionar clientes y procesar pedidos. Utiliza principios de programación orientada a objetos y una interfaz gráfica de usuario (GUI) para facilitar la interacción.

### 2. Estructura del Proyecto

El proyecto está organizado en varios paquetes y clases, cada uno con una responsabilidad específica. Los paquetes principales son:

- **modelo:** Contiene las clases que representan las entidades del negocio (Producto, Cliente, Pedido).
- **servicio:** Contiene las clases que implementan la lógica de negocio (ProductoServicio).
- **vista:** Contiene las clases que manejan la interfaz gráfica de usuario (ProductoVista, ClienteVista, PedidoVista).

### 3. Instrucciones de Instalación

#### Requisitos Previos

- Java Development Kit (JDK) 8 o superior
- Un IDE como IntelliJ IDEA, Eclipse o NetBeans
- Maven (si el proyecto usa Maven para la gestión de dependencias)

### 4. Guía de Uso

#### Funcionalidades Principales

- **Agregar Productos:** Permite agregar nuevos productos a la tienda.
- **Eliminar Productos:** Permite eliminar productos existentes de la tienda.



## 5. Descripción de Clases

### Clase Principal: Main

La clase **Main** es el punto de entrada de la aplicación. Inicializa la aplicación y muestra el menú principal.

### Clase Producto

La clase **Producto** representa un producto en la tienda. Contiene atributos como nombre, precio y cantidad. Incluye métodos para obtener y establecer estos atributos.

- **Atributos:**
  - **nombre:** Nombre del producto.
  - **precio:** Precio del producto.
  - **cantidad:** Cantidad disponible del producto.
- **Métodos:**
  - **getNombre():** Obtiene el nombre del producto.
  - **setNombre(String nombre):** Establece el nombre del producto.
  - **getPrecio():** Obtiene el precio del producto.
  - **setPrecio(double precio):** Establece el precio del producto.
  - **getCantidad():** Obtiene la cantidad del producto.
  - **setCantidad(int cantidad):** Establece la cantidad del producto.
  -

## 6. Servicios

### Clase ProductoServicio

La clase **ProductoServicio** gestiona las operaciones relacionadas con productos, como agregar, eliminar y listar productos.

- **Métodos:**
  - **agregarProducto(Producto producto):** Agrega un nuevo producto a la lista de productos.
  - **eliminarProducto(String nombre):** Elimina un producto de la lista de productos por su nombre.
  - **listarProductos():** Devuelve una lista de todos los productos disponibles.



## 7. Vistas

Las vistas manejan la interacción con el usuario para las operaciones relacionadas con productos, clientes y pedidos.

### Clase ProductoVista

- **Descripción:** Maneja la interfaz gráfica para las operaciones relacionadas con productos.
- **Funcionalidades:**
  - Permite agregar nuevos productos.
  - Permite eliminar productos existentes.
  - Permite listar todos los productos disponibles.

## 8. Ejecución de la Aplicación

Para ejecutar la aplicación, sigue los pasos de instalación y asegúrate de que todas las dependencias estén correctamente configuradas. Usa tu IDE para ejecutar la clase principal y navega por las opciones del menú para interactuar con las diferentes funcionalidades de la aplicación.

## 9. Interfaz Gráfica de Usuario (GUI)

La GUI de la aplicación está diseñada para ser intuitiva y fácil de usar. Las principales pantallas incluyen:

- **Inicio de sesión**

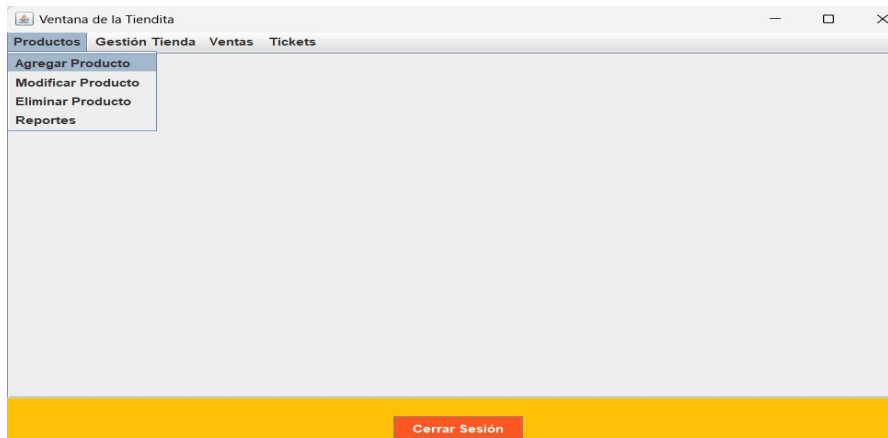
La imagen muestra una ventana de software titulada "Ventana de Inicio de Sesión". El formulario contiene los siguientes elementos:

- Un campo de texto etiquetado "Usuario:" con el valor "jesus" ingresado. A la izquierda del campo hay un ícono de un hombre.
- Un campo de texto etiquetado "Contraseña:" con tres puntos de ocultación "...". A la izquierda del campo hay un ícono de una cerradura.
- Un campo de texto etiquetado "Verificar Contraseña:" con tres puntos de ocultación "...".
- Una barra de botones en la parte inferior con cuatro botones: "Iniciar Sesión" (verde), "Crear Usuario" (azul), "Ver Usuarios" (naranja) y "Borrar Usuario" (rojo).

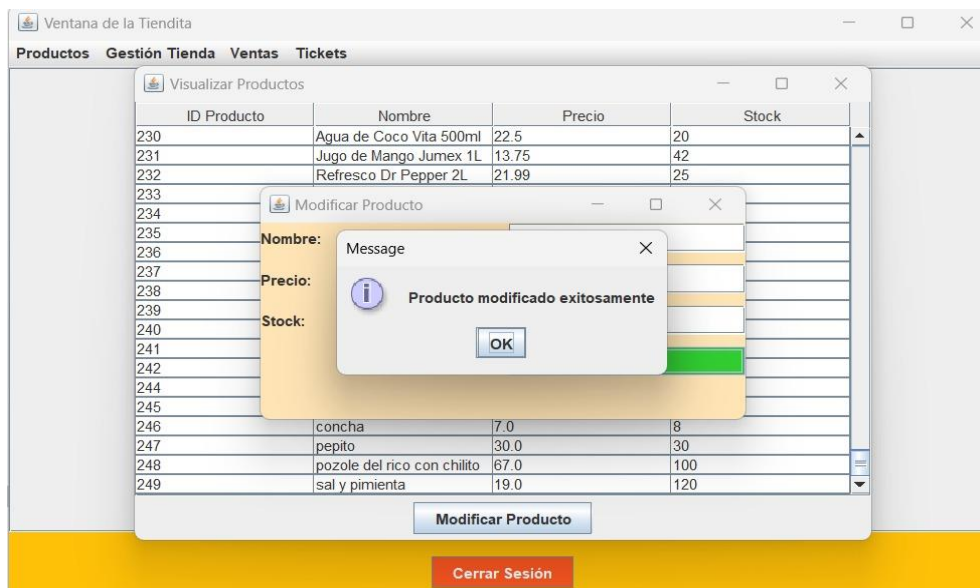


Aquí nos permite iniciar sesión mediante un usuario en una interfaz amigable , además de que si no tenemos creado el usuario podemos o tenemos la oportunidad de crearlo también poder ver cuantos usuarios tenemos para poder gestionar la tiendita y para finalizar tenemos también la opción de borrar usuarios que ya no queremos o simplemente quitarlos

- **Pantalla Principal:** Ofrece opciones para gestionar productos, clientes y pedidos. Además de agregar productos , modificarlos e incluso eliminarlos .



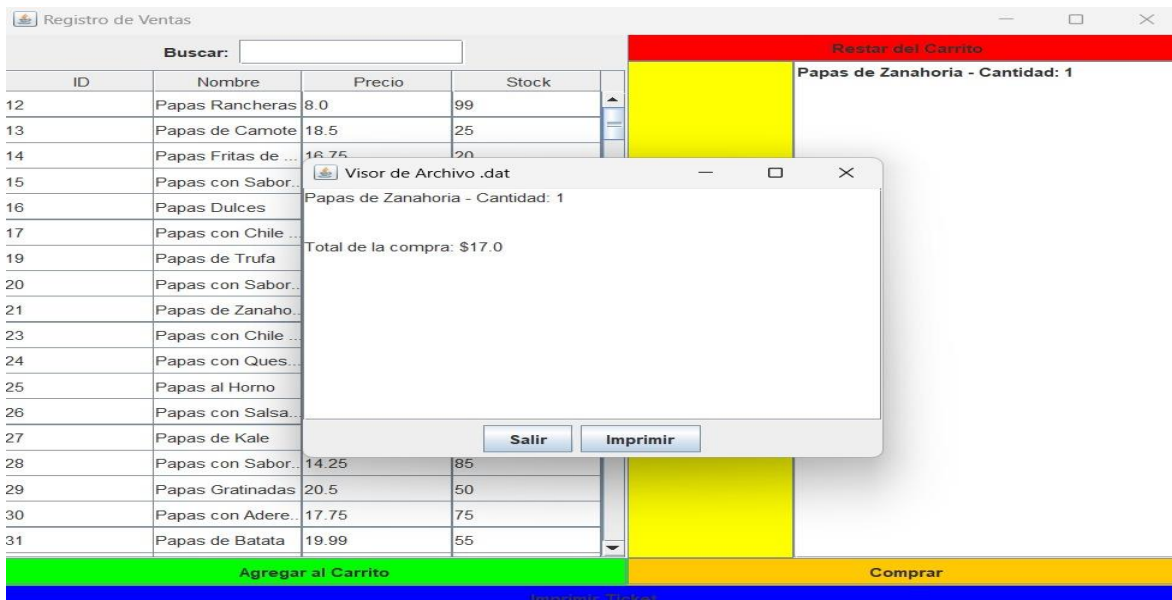
- **Pantalla de Productos:** Permite al usuario agregar, eliminar y listar productos, aquí tenemos una imagen de como esta implementada en la interfaz grafica .





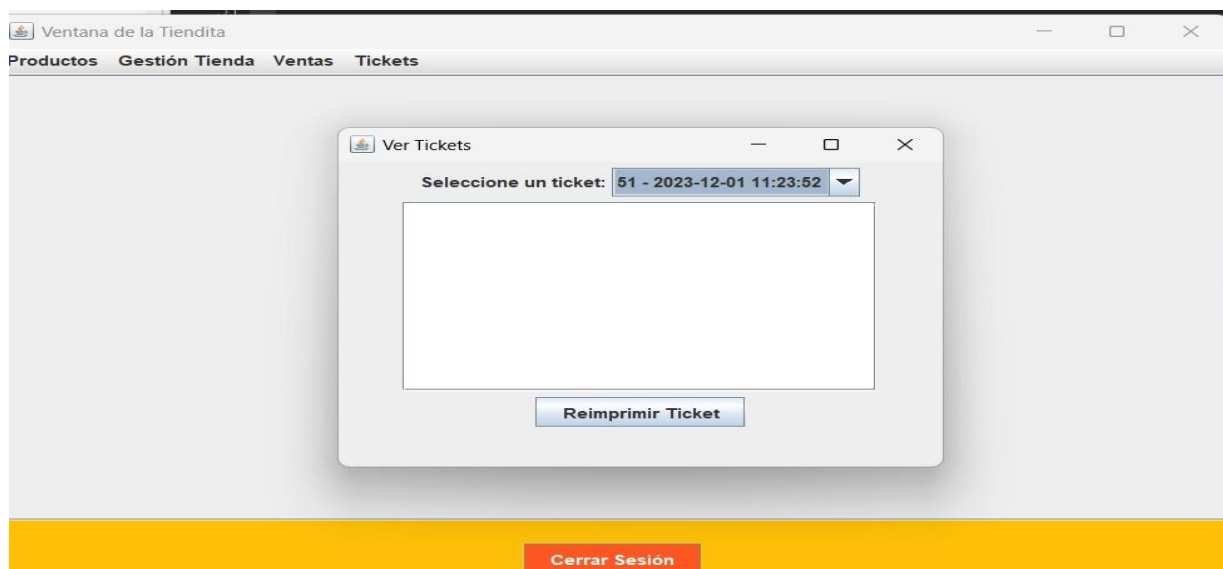
- **Pantalla Registro de ventas**

Aquí se puede visualizar cada venta que se va teniendo y poder ver que tantas ventas tenemos además de las cantidades de productos vendidos y su total .



- **Pantalla de Ticket**

Aquí podemos tener la opción de imprimir el ticket de compra de cada cliente el total de los productos y que productos compro





## 10. Contacto

Para cualquier pregunta o soporte adicional, por favor contacta al desarrollador del proyecto a través del siguiente correo electrónico: [binarytitans4@gmail.com].

## 11. Notas Adicionales

- **Mantenimiento:** Se recomienda revisar y actualizar periódicamente las dependencias y bibliotecas utilizadas en el proyecto.
- **Mejoras Futuras:** Algunas posibles mejoras incluyen la integración con una base de datos para persistir la información y la implementación de nuevas funcionalidades como informes de ventas.

## 12. Diagramas

### Diagrama de Clases

El diagrama muestra el diseño de la base de datos para la aplicación "Tiendita", que incluye las tablas usuarios, productos, ventas, y tickets. A continuación se describe cada tabla y sus relaciones:

#### Tabla usuarios

id (INT): Identificador único para cada usuario.

nombre\_usuario (VARCHAR(255)): Nombre del usuario.

contraseña (VARCHAR(70)): Contraseña del usuario.

estado (ENUM): Estado del usuario, que puede tener varios valores (por ejemplo, activo, inactivo).

ventas\_nombre\_producto (VARCHAR(60)): Relación con la tabla ventas mediante el nombre del producto.

productos\_id\_producto (INT): Relación con la tabla productos mediante el ID del producto.

La tabla usuarios se relaciona con las tablas ventas y productos para gestionar qué productos son vendidos y por quién.



### **Tabla productos**

id\_producto (INT): Identificador único para cada producto.

nombre (VARCHAR(255)): Nombre del producto.

precio (DOUBLE): Precio del producto.

stock (INT): Cantidad de productos disponibles en el inventario.

precio\_compra (DOUBLE): Precio de compra del producto.

envios\_id (INT): Relación con otra tabla de envíos (no mostrada en el diagrama).

La tabla productos almacena información sobre los productos disponibles en la tienda. Se relaciona con la tabla ventas para registrar las ventas de estos productos.

### **Tabla ventas**

nombre\_producto (VARCHAR(60)): Nombre del producto vendido.

cantidad (INT): Cantidad de productos vendidos.

ganancias (DECIMAL(10,2)): Ganancia obtenida por la venta.

tickets\_id\_ticket (INT): Relación con la tabla tickets mediante el ID del ticket.

productos\_id\_producto (INT): Relación con la tabla productos mediante el ID del producto.

La tabla ventas registra cada transacción de venta, incluyendo la cantidad de productos vendidos y las ganancias obtenidas. Se relaciona con las tablas productos y tickets.

### **Tabla tickets**

id\_ticket (INT): Identificador único para cada ticket.

fecha (DATETIME): Fecha y hora en que se emitió el ticket.

contenido (TEXT): Detalles del contenido del ticket.

La tabla tickets almacena los detalles de cada transacción, incluyendo la fecha y una descripción detallada del ticket. Se relaciona con la tabla ventas para asociar las ventas con un ticket específico.





## Relaciones Entre Tablas

usuarios  $\leftrightarrow$  productos: La relación productos\_id\_producto en la tabla usuarios indica los productos que un usuario puede gestionar o vender.

usuarios  $\leftrightarrow$  ventas: La relación ventas\_nombre\_producto en la tabla usuarios conecta los usuarios con los productos vendidos.

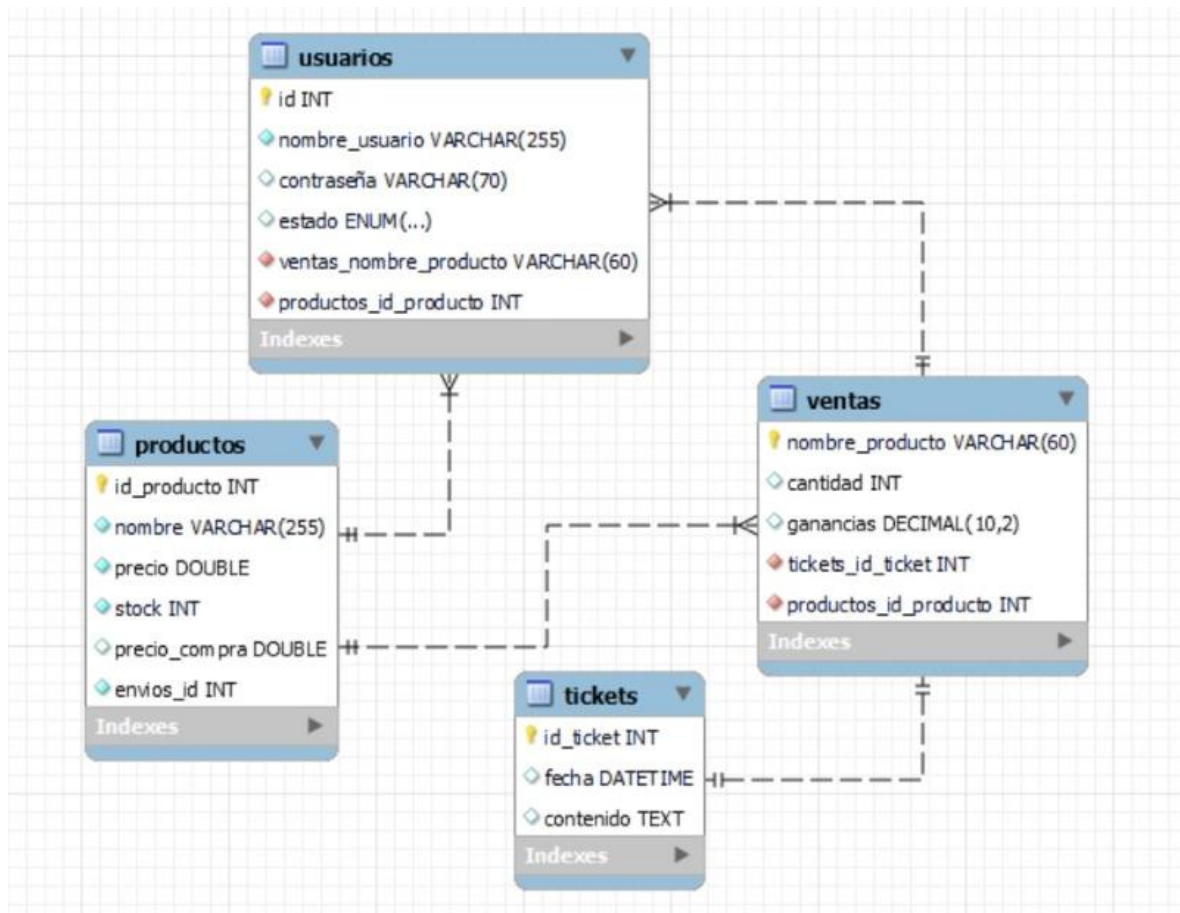
productos  $\leftrightarrow$  ventas: La relación productos\_id\_producto en la tabla ventas indica qué producto específico ha sido vendido.

ventas  $\leftrightarrow$  tickets: La relación tickets\_id\_ticket en la tabla ventas conecta cada venta con un ticket específico.

## Diagramas Relacionales

Las líneas punteadas en el diagrama representan las relaciones entre las tablas. Estas relaciones son clave para asegurar la integridad referencial de la base de datos y permitir la gestión efectiva de usuarios, productos, ventas y tickets dentro de la aplicación "Tiendita".

## Diagrama para la base de datos del Proyecto Tienda





## 13. Descripción Detallada de Funcionalidades

### Gestión de Productos

- **Agregar Producto:** El usuario puede ingresar los detalles del producto, como nombre, precio y cantidad. El producto se añade a la lista de productos disponibles.
- **Eliminar Producto:** El usuario puede eliminar un producto existente de la lista por su nombre.
- **Listar Productos:** El usuario puede ver una lista de todos los productos disponibles en la tienda.

## CONCLUSION

El proyecto "Tiendita" es una aplicación Java robusta y bien estructurada diseñada para gestionar las operaciones de una pequeña tienda. La aplicación permite la gestión integral de productos, clientes y pedidos a través de una interfaz gráfica de usuario intuitiva y fácil de usar. Utiliza principios sólidos de programación orientada a objetos y una arquitectura modular, lo que facilita su mantenimiento y escalabilidad.

### Aspectos Destacados:

#### 1. Estructura Modular:

- **Modelo:** Representa las entidades del negocio como Producto, Cliente y Pedido.
- **Servicio:** Implementa la lógica de negocio para manejar productos, clientes y pedidos.
- **Vista:** Gestiona la interfaz gráfica de usuario para interactuar con el sistema.

#### 2. Funcionalidades Clave:

- **Gestión de Productos:** Agregar, eliminar y listar productos.

#### 3. Interfaz de Usuario (GUI):

- Pantallas dedicadas para la gestión de productos, clientes y pedidos.
- Funcionalidades adicionales como el registro de ventas y la impresión de tickets de compra.

#### 4. Base de Datos:

- Diseño relacional que asegura la integridad referencial entre usuarios, productos, ventas y tickets.



- Tablas bien definidas que soportan las operaciones del sistema.

### **Mejores Prácticas y Mantenimiento:**

- **Actualización Periódica:** Mantener actualizadas las dependencias y bibliotecas utilizadas en el proyecto.
- **Mejoras Futuras:** Integrar una base de datos para persistencia de datos y añadir nuevas funcionalidades como informes de ventas.

### **Contacto y Soporte:**

Para cualquier pregunta o soporte adicional, los usuarios pueden contactar al desarrollador a través del correo electrónico proporcionado.

El proyecto "Tiendita" es una solución completa para la gestión de tiendas pequeñas, ofreciendo una amplia gama de funcionalidades que cubren las necesidades básicas de operación y administración, mientras mantiene una estructura organizada y fácil de mantener.

