



Universidad Autónoma del Estado de México
Unidad Académica Profesional Tianguistenco

Ingeniería en software

Unidad de aprendizaje:
ESTRUCTURAS DE DATOS

Profesor:
MARITZA FLORES DOMINGUEZ

Alumno:
Marcos Gabriel Mandujano Sánchez

Fecha de entrega: 16/marzo/2025

Reporte de Práctica: Caja Registradora

1. Introducción

El presente reporte describe el desarrollo de la práctica "**Caja Registradora**", la cual consiste en la implementación de un programa en Java que simula el funcionamiento de una caja registradora utilizando pilas. El programa permite agregar y eliminar artículos, calcular el total de la compra, aplicar descuentos y generar un ticket de compra con formato.

2. Objetivos

- Implementar una interfaz gráfica en Java para simular una caja registradora.
- Utilizar pilas para almacenar y gestionar los artículos ingresados.
- Validar los datos de entrada (nombre, cantidad, precio) para evitar errores.
- Calcular el importe total de la compra y aplicar descuentos cuando corresponda.
- Generar un ticket de compra con un formato específico que incluya encabezado, cuerpo y pie de página.

3. Metodología

3.1. Herramientas utilizadas

- **Lenguaje de programación:** Java.
- **Entorno de desarrollo:** IntelliJ IDEA, Eclipse o cualquier IDE compatible con Java.
- **Control de versiones:** GitHub (para el repositorio privado del proyecto).

3.2. Estructura del proyecto

El proyecto se organizó en tres paquetes principales:

1. **Paquete GUI:** Contiene la clase CajaRegistradoraGUI, que implementa la interfaz gráfica del programa.
2. **Paquete Pilas:** Contiene las clases CajaRegistradoraPila y Articulo, que gestionan la lógica de las pilas y los artículos.
3. **Clase Main:** Es el punto de entrada del programa y se encarga de ejecutar la interfaz gráfica.

3.3. Funcionalidades implementadas

- **Agregar artículos:** El usuario puede ingresar el nombre, cantidad y precio de un artículo, el cual se almacena en una pila.
- **Eliminar artículos:** El usuario puede eliminar el último artículo agregado a la pila.

- **Generar ticket:** El programa genera un ticket con el formato solicitado, incluyendo el encabezado, los detalles de los artículos, el descuento aplicado (si corresponde), el importe total y un mensaje de agradecimiento.
- **Validación de datos:** Se validan los campos de entrada para asegurar que los datos ingresados sean correctos (por ejemplo, que la cantidad y el precio sean números válidos).

4. Resultados

4.1. Interfaz gráfica

La interfaz gráfica del programa es sencilla y consta de los siguientes elementos:

- Campos de texto para ingresar el nombre, cantidad y precio del artículo.
- Botones para agregar artículos, eliminar artículos y generar el ticket.
- Un área de texto para mostrar el ticket generado.

4.2. Funcionamiento del programa

- **Agregar artículos:** El usuario ingresa los datos del artículo y hace clic en "Agregar artículo". El artículo se almacena en la pila y se muestra un mensaje de confirmación.
- **Eliminar artículos:** El usuario hace clic en "Eliminar artículo" para eliminar el último artículo agregado.
- **Generar ticket:** El usuario hace clic en "Generar Ticket" para ver el ticket con los detalles de la compra.

4.3. Ejemplo de ticket generado

A continuación, se muestra un ejemplo de cómo se ve el ticket generado por el programa:

=== Tienda 'Mi Tienda' ===

Dirección: Calle Falsa 123, Ciudad, País

Teléfono: +123 456 7890

=====

Artículo	Cantidad	Precio	Resultado
Lápiz	2	\$1.5	\$3.0
Cuaderno	3	\$5.0	\$15.0

Descuento aplicado: \$0.0			
Importe Total: \$18.0			
Total a Pagar: \$18.0			

Usuario invitado

16/03/2025 08:31:31 a. m.

=====

¡Gracias por su compra!

5. Conclusiones

- El programa cumple con los objetivos planteados, permitiendo gestionar artículos, calcular el importe total y generar un ticket con formato.
- El uso de pilas fue adecuado para este proyecto, ya que permitió manejar los artículos de manera eficiente.
- La interfaz gráfica, aunque sencilla, es funcional y permite al usuario interactuar con el programa de manera intuitiva.
- Se logró implementar validaciones para evitar errores en la entrada de datos, lo que mejora la robustez del programa.