

# **“Sistema de control de pedidos de una fábrica de zapatos”**

**CURSO: Experiencias Formativas En Situaciones Reales De Trabajo**

**DOCENTE:**  
**Giannina Milagritos Luyo Fajardo**

**SEGUNDO CICLO**

**SEMESTRE 2023 – 01**

**Coordinador:**

**Angel David Veliz Mendoza**

**Integrantes:**

**Angel David Veliz Mendoza**

**Piero Alexander Caro Jara**

**Ericka Pierina Rodas Loaiza**

**Cesar Alejandro Fiestas Ezcurra**

**CARRERAS PROFESIONALES**

**CIBERTEC  
LIMA CENTRO, 2023**

## Índice

- ❖ Introducción
- ❖ Justificación
- ❖ Objetivos
- ❖ Definición
- ❖ Entregables
- ❖ Conclusiones
- ❖ Recomendaciones
- ❖ Glosario
- ❖ Bibliografía
- ❖ Anexos

## Introducción

En el presente trabajo hemos descrito un pequeño sistema la que llamaremos ZAPATEX para el control de zapatos, por lo cual describimos un listado de clientes con su respectivo nombre, apellido, dirección, teléfono y DNI, productos con nombre, precio, stock actual, stock mínimo y stock máximo y ventas con código del cliente, código del producto, fecha de Venta, Importe Sub Total, importe IGV y pago Total este programa puede listar, ingresar, modificar, consultar y eliminar sobre la tabla Cliente análogamente sobre la tabla Producto en las características antes mencionadas.

La parte de ventas podemos efectuar una venta para la compra de zapatos pero con un límite de stock que sería nuestro control entonces para tener stock el programa tiene una opción de almacén en los cuales podemos ingresar productos al almacén sin salir del stock máximo pero el programa, no solo tiene la opción de vender o almacenar sino también de generar un reporte por ejemplo listado general de ventas que te genera un listado de ventas en lo que figura el cliente, producto, fecha y los importes subtotal y IGV y por último el pago total, además también el sistema también tiene un listado de productos con stock por debajo del stock mínimo, listado de productos por unidades vendidas acumuladas y listado de productos por importe total acumulado.

Finalmente, en la opción sistema solo encontraras la opción salir del sistema que cierra el sistema, pero recordemos que todos los cambios siguen guardados cuando vuelves abrir el sistema dado que los cambios se guardan en archivos .txt y estos son cargados por el sistema.

## Justificación

ZAPATEX es un proyecto para el control de zapatos pues tiene características limitadas de stock donde el cliente puede observar la cantidad del producto antes de que se agote; pero no siempre los modelos de zapatos tienen que agotarse, sino que este sistema puede variar la cantidad de producto por ejemplo aumentar el stock de actual hasta llegar al stock máximo del producto para una buena tranquilidad para el cliente.

Esta solución de poder bajar o subir el stock de cada zapato es impactante porque al cambiar la cantidad de los zapatos también se podrá cambiar el precio pero la tipo de zapato siempre va ser la misma porque tiene un código que lo identifica; la necesidad que tienen los clientes por los diferentes modelos de zapatos está incluida en el sistema por lo cual no sería una déficit para la venta al contrario sería un beneficio porque la venta se podría realizar sin ningún inconveniente y además el sistema puede ingresar un producto nuevo generando un nuevo código pues el cliente siempre tiene la razón cuando realiza su pedido.

El beneficiario directo en nuestro caso sería las tiendas ZAPATEX no tendrían que preocuparse por cantidad de zapatos sino al contrario podrían realizar la venta con un stock programado; por otro lado, las tiendas tienen que realizar un reporte por productos por unidades vendidas que deben lograr, pero esto mayormente se nota en la demanda de las personas por lo cual el sistema puede cambiar esta cantidad de zapatos que tienen en el stock actual a beneficio propio.

Estas configuraciones tienen beneficios indirectos con los clientes pues el sistema tiene una boleta electrónica en lo cual solo debes ingresar los códigos del cliente y el código del producto y por último la cantidad de producto que llevará el cliente. En la parte de stock el cliente estaría atento a cualquier agotamiento y lo que son las demandas por fiestas como navidad pues el cliente compraría a tiempo antes de que el producto se agote.

## Objetivos

S: Ventas, Promociones

M: Un 15% más de ventas, de 1000 a 1150.

A: Es realista, se puede lograr con buena promoción

R: Se está buscando aumentar las ventas en cada tienda generando promociones.

T: En 2 meses

Los objetivos que tiene este sistema son:

- a) “Aumentar el número de ventas en cada local en un 15% más, durante los próximos 2 meses, con el objetivo de aumentar las ventas y generar ingresos”.
  
- b) “Promocionar los zapatos mediante las ofertas en cada tienda en un 15% más en un plazo de 2 meses”.

## Definición

El menú principal tiene los siguientes ítems Mantenimiento, Ventas, Almacén, Reportes y Sistema explicaremos brevemente las funciones de cada una de ellas.

- **SISTEMA**

En el menú SISTEMA solamente no permite salir del sistema. Es la única opción que posee.



- **MANTENIMIENTO**

Encontraremos 2 funciones:



### 1) Clientes:

Clientes tiene varias funciones en la encontraremos:

- LISTAR: Solo muestra toda la lista de clientes que tiene el sistema

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia 1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409

Ahora si la lista ya esta actualizada y quieres actualizarlo nuevamente el sistema te bota un aviso.

Sistema de Pedidos de Zapatos | Cliente

Código :	<input type="text"/>	Aceptar	<a href="#">Listar</a>
Nro DNI :	<input type="text"/>		<a href="#">Ingresar</a>
Nombres :	<input type="text"/>		<a href="#">Modificar</a>
Apellidos :	<input type="text"/>		<a href="#">Consultar</a>
Teléfono :	<input type="text"/>		<a href="#">Eliminar</a>
Dirección :	<input type="text"/>		



Información

No hay nuevos clientes para mostrar.

[Aceptar](#)

Código	Nombre	Teléfono	DNI
1001	Angel	658060	47467409
1003	Javier Jose	658123	46467409
1004	Carlos Guillermo	658060	47467400
1005	Gerald Andres	658456	47467449
1006	Janet Soledad	951658125	47467451
1007	Alberto Rene	951651414	47467301
1008	Doris Talia	951657589	47467789
1009	Richard Frank	951650256	47467259
1010	Angel David	951658060	47467409

- INGRESAR: Aquí al presionar ingresar nos genera automáticamente un código en este 1011 y se nos habilita las opciones para llenar datos.

Sistema de Pedidos de Zapatos | Cliente



Código :	<input type="text" value="1011"/>	<input type="button" value="Aceptar"/>	<input type="button" value="Listar"/>
Nro DNI :	<input type="text" value="47467410"/>		<input type="button" value="Ingresar"/>
Nombres :	<input type="text" value="Calos"/>		<input type="button" value="Modificar"/>
Apellidos :	<input type="text" value="Veliz Apolinario"/>		<input type="button" value="Consultar"/>
Teléfono :	<input type="text" value="951658060"/>		<input type="button" value="Eliminar"/>
Dirección :	<input type="text" value="Jr Ancash 3522"/>		

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia 1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409

Una vez llenado los datos damos click en aceptar y automáticamente vemos que en la lista por la parte posterior se agrego el cliente 1011

The screenshot shows a software interface titled "Sistema de Pedidos de Zapatos | Cliente". On the left, there is a form with fields for Código (1012), Nro DNI, Nombres, Apellidos, Teléfono, and Dirección. To the right of the form is a large red and white sneaker graphic. Below the form is a table listing 11 clients. The last row of the table, client number 1011, has a blue background.

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409
1011	Calos	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

Tengamos en cuenta que el sistema sigue con la funcionalidad de ingresar cliente por ello que genera el código 1011.

- MODIFICAR: para modificar primero tenemos que seleccionar con el mouse el cliente a modificar y luego dar click en modificar y ya con ello podremos cambiar algún dato del cliente.

The screenshot shows the same software interface as the previous one. The client entry form now has Código (1011), Nro DNI (47467410), Nombres (Calos), Apellidos (Veliz Apolinario), Teléfono (951658060), and Dirección (Jr Ancash 3522). The client list table remains the same, showing 11 clients, with the last row (client 1011) having a blue background.

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409
1011	Calos	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

Supongamos que modificaremos agregando el segundo nombre a Carlos pues le pondremos Carlos Jose, daremos aceptar y se actualizara la lista.

The screenshot shows a software application window titled "Sistema de Pedidos de Zapatos | Cliente". On the left, there is a form with fields for "Código", "Nro DNI", "Nombres", "Apellidos", "Teléfono", and "Dirección". To the right of the form is a large red and white graphic of a sneaker. At the top right are five buttons: "Listar", "Ingresar", "Modificar", "Consultar", and "Eliminar". Below the graphic is a table listing 11 clients with columns for Código, Nombre, Apellido, Dirección, Teléfono, and DNI. A modal window is open in the center, containing a message "Mensaje" and a button "Aceptar".

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia 1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1011	Calos Jose	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

En un inicio hemos dicho que debe seleccionarse un cliente pero que pasaría sino seleccionamos ningún cliente pues el sistema te dará un aviso que debe seleccionar el cliente a modificar.

This screenshot shows the same application window as the previous one, but the modal in the center has changed. It now displays an information icon and the text "Seleccione en la tabla al cliente a modificar". Below this is a table with the same 11 client entries as the previous screenshot. At the bottom of the modal is a button labeled "Aceptar".

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia 1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1011	Calos Jose	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

- **CONSULTAR:** Para consultar tiene que darle click en consultar te dará un aviso que debes ingresar el código de cliente.

Sistema de Pedidos de Zapatos | Cliente

Código :	<input type="text"/>	Aceptar	Listar
Nro DNI :	<input type="text"/>	Ingresar	
Nombres :	<input type="text"/>	Modificar	
Apellidos :	<input type="text"/>	Consultar	
Teléfono :	<input type="text"/>	Eliminar	
Dirección :	<input type="text"/>		



Mensaje

Ingrresa el código que deseas consultar

Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1003	Javier Jose	Blanco Martinez	Psj. Mercurio 158	951658125	46467409
1004	Carlos Guillerm	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1005	Gerald Andres	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1006	Janet Soledad	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409
1011	Calos Jose	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

Ingresas unos de los códigos que vez en la tabla y te botara todos los datos, pero tengamos en cuenta que no se puede modificar solo es ver.

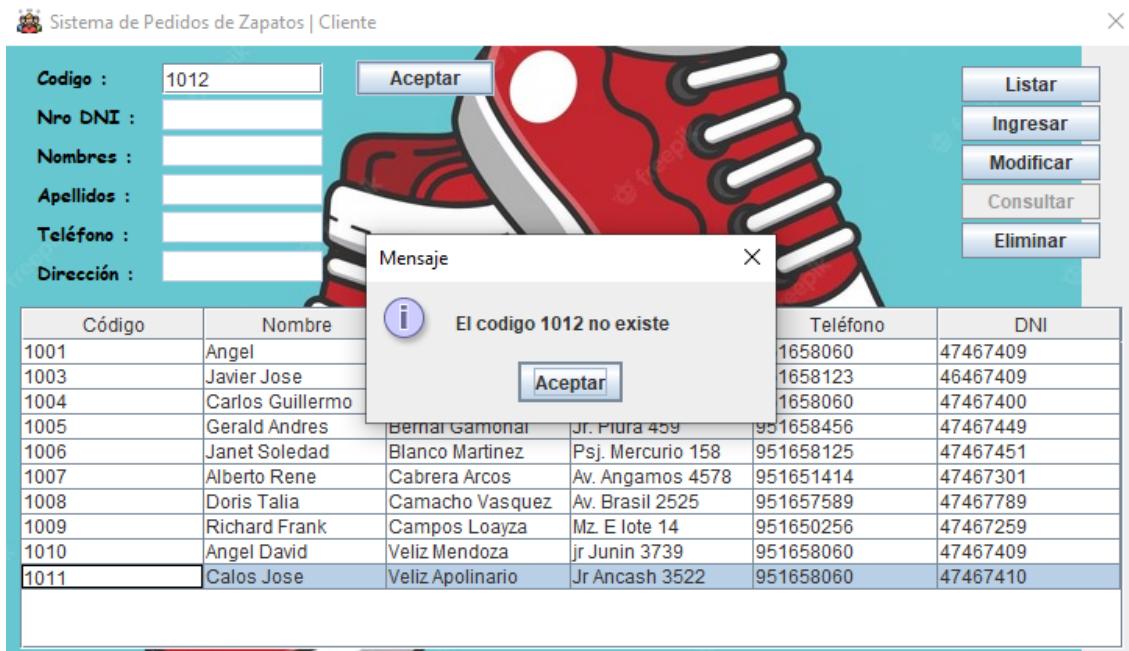
Sistema de Pedidos de Zapatos | Cliente

Código :	<input type="text" value="1011"/>	Aceptar	Listar
Nro DNI :	<input type="text" value="47467410"/>	Ingresar	
Nombres :	<input type="text" value="Calos Jose"/>	Modificar	
Apellidos :	<input type="text" value="Veliz Apolinario"/>	Consultar	
Teléfono :	<input type="text" value="951658060"/>	Eliminar	
Dirección :	<input type="text" value="Jr Ancash 3522"/>		

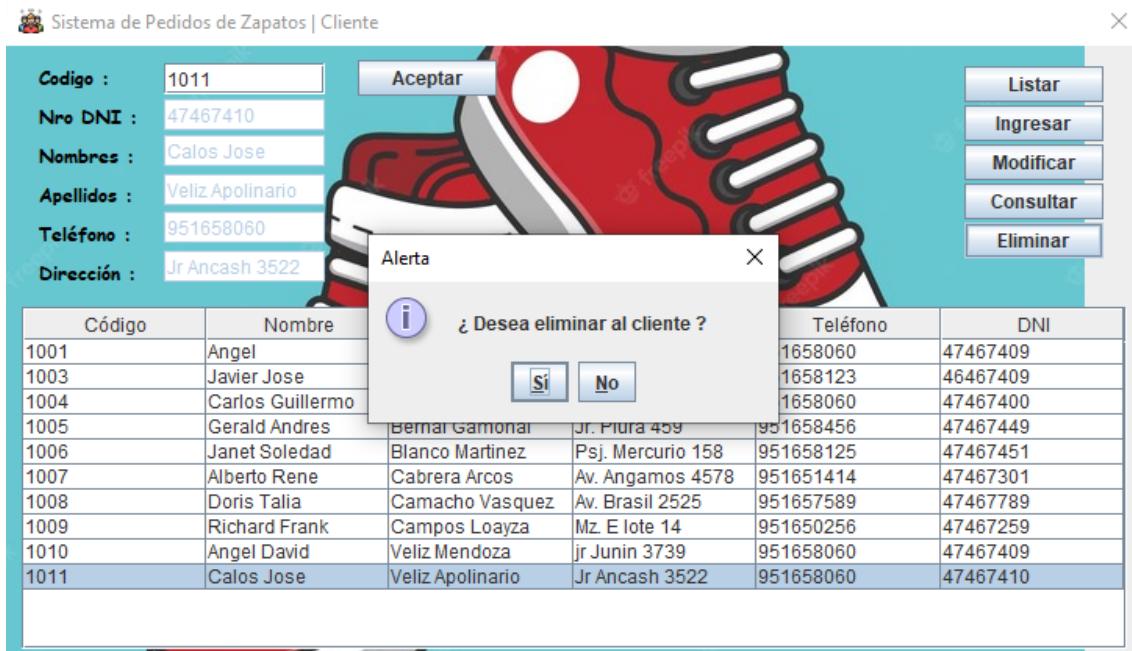


Código	Nombre	Apellido	Dirección	Teléfono	DNI
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loayza	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409
1011	Calos Jose	Veliz Apolinario	Jr Ancash 3522	951658060	47467410

Que pasaría si ingresamos un código que no esta en la lista pues el sistema nos bota un aviso que hemos ingresado mal el código.



- **ELIMINAR:** En esta opción elimina al cliente de la barra sombreada pero antes de eliminar te da un aviso si esta seguro de eliminar si pones no la lista se mantiene.



Si la opción es si es presionado entonces se elimina el cliente y automáticamente se actualiza la lista

Sistema de Pedidos de Zapatos | Cliente

Código :	Aceptar	Listar			
Nro DNI :		Ingresar			
Nombres :		Modificar			
Apellidos :		Consultar			
Teléfono :		Eliminar			
Dirección :					
1001	Angel	Veliz Mendoza	Jr Junin 3739	951658060	47467409
1003	Javier Jose	Acero Sosa	Av. Bolivia 1357	951658123	46467409
1004	Carlos Guillermo	Ascencios Estrada	Jr. Pasco 1478	951658060	47467400
1005	Gerald Andres	Bernal Gamonal	Jr. Piura 459	951658456	47467449
1006	Janet Soledad	Blanco Martinez	Psj. Mercurio 158	951658125	47467451
1007	Alberto Rene	Cabrera Arcos	Av. Angamos 4578	951651414	47467301
1008	Doris Talia	Camacho Vasquez	Av. Brasil 2525	951657589	47467789
1009	Richard Frank	Campos Loaya	Mz. E lote 14	951650256	47467259
1010	Angel David	Veliz Mendoza	jr Junin 3739	951658060	47467409

## 2) Productos:

Clientes tiene varias funciones en la encontraremos:

- INGRESAR: Solo muestra toda la lista de clientes que tiene el sistema

Sistema de Pedidos de Zapatos | Producto

Código :	Nombre :	Precio :	Stock Actual :	Stock Mínimo :	Stock Máximo :	Listar	Ingresar
2001	BOTINES	30.0	468	1	1000		
2002	ZAPATILLAS	1.0	80	25	110		
2003	SANDALIAS	250.0	159	1	1000		
2004	TACOS	15.0	12	1	200		
2005	BOTAS	5.0	32	20	1000		
2006	PANTUFLAS	10.0	49	20	150		
2007	BAILARINA	10.0	8	0	100		

Ahora ingresamos los datos y presionar en listar y automáticamente se actualiza la lista ingresando el producto con el código generado.

Sistema de Pedidos de Zapatos | Producto

Código :	2003		Ingresar
Nombre :	SANDALIAS		Modificar
Precio :	250.0		Consultar
Stock Actual :	159		Eliminar
Stock Mínimo :	1		
Stock Máximo :	1000		

**Listar**

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1	1000
2002	ZAPATILLAS	1.0	80	25	110
2003	SANDALIAS	250.0	159	1	1000
2004	TACOS	15.0	12	1	200
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100

Veamos que si nos falta algún dato el sistema nos bota el aviso correspondiente al primer dato que falte

- **Falta Nombre**

Sistema de Pedidos de Zapatos | Producto

Código :	2008		Ingresar
Nombre :			Modificar
Precio :			Consultar
Stock Actual :			Eliminar
Stock Mínimo :			
Stock Máximo :			

**Información**

X INGRESE NOMBRE

Aceptar

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1000	
2002	ZAPATILLAS	1.0	80	110	
2003	SANDALIAS	250.0	159	1000	
2004	TACOS	15.0	12	200	
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100

### • Falta Precio

Sistema de Pedidos de Zapatos | Producto

The screenshot shows a software interface for managing shoe products. On the left, there's a form with fields for Código (2008), Nombre (Pantuflas), Precio (empty), Stock Actual (empty), Stock Mínimo (empty), and Stock Máximo (empty). A green checkmark icon is visible next to the Código field. On the right, there's a large image of a red and white sneaker. At the top right are buttons for Ingresar, Modificar, Consultar, and Eliminar. Below the image is a table of product data:

CÓDIGO	NOMBRE	STOCK MIN	STOCK MAX
2001	BOTINES	1000	
2002	ZAPATILLAS	110	
2003	SANDALIAS	1000	
2004	TACOS	200	
2005	BOTAS	5.0	32
2006	PANTUFLAS	10.0	49
2007	BAILARINA	10.0	8
		20	1000
		20	150
		0	100

A modal dialog box titled "Información" with a red "X" icon contains the message "INGRESE PRECIO" and a "Aceptar" button.

### • Falta Stock Actual

Sistema de Pedidos de Zapatos | Producto

The screenshot shows the same software interface as the previous one. The form now has a filled Nombre (Pantuflas) and a filled Precio (12.00). The Stock Actual field is still empty. The rest of the form and the modal dialog are identical to the first screenshot, showing the same table of product data and the "INGRESE STOCK ACTUAL" message in the modal.

- Falta Stock Mínimo

Sistema de Pedidos de Zapatos | Producto

The screenshot shows a software interface for managing shoe inventory. On the left, there's a form with fields for Código (2008), Nombre (Pantuflas), Precio (12.00), Stock Actual (20), Stock Mínimo (empty), and Stock Máximo (empty). A green checkmark icon is next to the Código field. On the right, there's a list of products with columns for Código, Nombre, Stock Actual, Stock Min, and Stock Max. A modal dialog box titled "Información" with a red X icon displays the message "INGRESE STOCK MINIMO". Below the message is an "Aceptar" button. The background shows a large image of a red and white sneaker.

CÓDIGO	NOMBRE	STOCK MIN	STOCK MAX
2001	BOTINES	1000	
2002	ZAPATILLAS	110	
2003	SANDALIAS	1000	
2004	TACOS	200	
2005	BOTAS	5.0	32
2006	PANTUFLAS	10.0	49
2007	BAILARINA	10.0	8

- Falta Stock Máximo

Sistema de Pedidos de Zapatos | Producto

This screenshot is similar to the previous one but shows a maximum stock error. The product details form has the same fields and values. The modal dialog box now displays the message "INGRESE STOCK MAXIMO". The background image of the red and white sneaker is partially visible at the bottom.

CÓDIGO	NOMBRE	STOCK MIN	STOCK MAX
2001	BOTINES	1000	
2002	ZAPATILLAS	110	
2003	SANDALIAS	1000	
2004	TACOS	200	
2005	BOTAS	5.0	32
2006	PANTUFLAS	10.0	49
2007	BAILARINA	10.0	8

- **MODIFICAR:** para modificar primero tenemos que seleccionar con el mouse el producto a modificar y luego dar click en modificar y ya con ello podremos cambiar algún dato del cliente.

Sistema de Pedidos de Zapatos | Producto

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1	1000
2002	ZAPATILLAS	1.0	80	25	110
2003	SANDALIAS	250.0	159	1	1000
2004	TACOS	15.0	12	1	200
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100
2008	Pantuflas	12.0	20	3	30

Supongamos que modificaremos Pantuflas por PANTUFLA, daremos Listar y se actualizara la lista.

Sistema de Pedidos de Zapatos | Producto

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1	1000
2002	ZAPATILLAS	1.0	80	25	110
2003	SANDALIAS	250.0	159	1	1000
2004	TACOS	15.0	12	1	200
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100
2008	PANTUFLA	12.0	20	3	30

**CONSULTAR:** Para consultar tiene que darle click en consultar y como vemos siempre la lista tendrá una parte sombreada que es el dato que se da a consultar sino haciendo click en algún producto podremos consultar sus datos, recordemos que solo es para ver.

Sistema de Pedidos de Zapatos | Producto

The screenshot shows a software interface titled "Sistema de Pedidos de Zapatos | Producto". On the left, there is a form with fields for "Código" (2008), "Nombre" (PANTUFLA), "Precio" (12.0), "Stock Actual" (20), "Stock Mínimo" (3), and "Stock Máximo" (30). A green checkmark icon is next to the "Nombre" field. To the right of the form is a large image of a red and white sneaker. On the far right, there are four buttons: "Ingresar", "Modificar", "Consultar" (which is highlighted in blue), and "Eliminar". Below the form is a table with columns: CÓDIGO, NOMBRE, PRECIO, STOCK ACTUAL, STOCK MIN, and STOCK MAX. The table contains the following data:

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1	1000
2002	ZAPATILLAS	1.0	80	25	110
2003	SANDALIAS	250.0	159	1	1000
2004	TACOS	15.0	12	1	200
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100
2008	PANTUFLA	12.0	20	3	30

- **ELIMINAR:** En esta opción elimina al cliente de la barra sombreada pero antes de eliminar te da un aviso si estás seguro de eliminar si pones no la lista se mantiene.

Sistema de Pedidos de Zapatos | Producto

The screenshot shows the same software interface as the previous one, but with an additional "Alerta" (Alert) dialog box in the center. The dialog has an "i" icon and the question "¿DESEAS ELIMINAR ESTE REGISTRO?" (Do you want to delete this record?). Below the question are two buttons: "Sí" (Yes) and "No". The rest of the interface remains the same, including the form on the left and the table of products on the right.

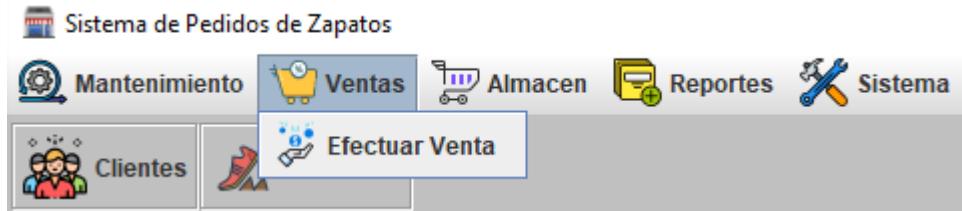
Si la opción es si es presionado entonces se elimina el cliente y automáticamente se actualiza la lista

Sistema de Pedidos de Zapatos | Producto

CÓDIGO	NOMBRE	PRECIO	STOCK ACTUAL	STOCK MIN	STOCK MAX
2001	BOTINES	30.0	468	1	1000
2002	ZAPATILLAS	1.0	80	25	110
2003	SANDALIAS	250.0	159	1	1000
2004	TACOS	15.0	12	1	200
2005	BOTAS	5.0	32	20	1000
2006	PANTUFLAS	10.0	49	20	150
2007	BAILARINA	10.0	8	0	100

## • VENTAS

Encontraremos una función:



### 1) Efectuar Venta

En esta opción el vendedor podrá vender la cantidad de zapatos que requiera el cliente para lo cual se debe ingresar el código del cliente y dar click en check verde sino es el cliente se presiona el X para agregar un nuevo código, análogamente para el producto; una vez realizado la venta presionando el efectuar venta generará una lista que indica BOLETA DE PAGO mostrando Código del Cliente, Nombre, Apellido, Código del producto, Nombre, Cantidad adquirida, Precio Unitario, Importe Subtotal, importe IGV y el importe a Pagar; así como vemos en el siguiente GUI.

 Ventas

Nroº Venta : 3042 14/06/2023

<b>DATOS DEL CLIENTE</b>	<b>DATOS DEL PRODUCTO</b>
 Cliente : <input type="text" value="1003"/> <span style="float: right;"> </span>	 Producto : <input type="text" value="2001"/> <span style="float: right;"> </span>
Nombre : <input type="text" value="Javier Jose"/>	Nombre Producto : <input type="text" value="BOTINES"/>
Apellidos : <input type="text" value="Acer Sosa"/>	Precio S/ : <input type="text" value="30.0"/>
<b>Boletas Electronica :</b> <span style="float: right;"><b>NUEVA VENTA</b></span>	
<b>EFFECTUAR VENTA</b>	
*****BOLETA DE PAGO*****	
CODIGO DEL CLIENTE : 1003 NOMBRE DEL CLIENTE : Javier Jose APELLIDO DEL CLIENTE : Acer Sosa CODIGO DEL PRODUCTO : 2001 NOMBRE DEL PRODUCTO : BOTINES CANTIDADES ADQUIRIDAS : 5 UNIDADES PRECIO UNITARIO : 30.0 IMPORTE SUBTOTAL : 150.0 IMPORTE IGV ( 18% ) : 27.0 IMPORTE TOTAL A PAGAR : 177.0	

## ● ALMACÉN

Encontraremos una función:



### 1) Agregar

Escribiremos el código del producto y daremos click en check verde para ver los datos de Nombre y los Stock (Actual, mínimo y máximo) y seguidamente podemos agregar la cantidad a ingresar

Almacen

Codigo del Producto:	<input type="text" value="2005"/>			Stock Actual:	<input type="text" value="32"/>
Nombre del Producto:	<input type="text" value="BOTAS"/>		Stock Maximo:	<input type="text" value="1000"/>	
Cantidad a Ingresar:	<input type="text" value="50"/>		Stock Minimo:	<input type="text" value="20"/>	
<input type="button" value="Ingresar Stock"/>			<input type="button" value="Nuevo Ingreso"/>		

Luego presionamos en Ingresar Stock y nos dará un mensaje de actualización

Almacen

Codigo del Producto:	<input type="text" value="2005"/>			Stock Actual:	<input type="text" value="32"/>
Nombre del Producto:	<input type="text" value="BOTAS"/>			Mensaje ¡El Stock Actual se Actualizo!	
Cantidad a Ingresar:	<input type="text" value="5"/>		<input type="button" value="Aceptar"/>		
<input type="button" value="Ingresar Stock"/>			<input type="button" value="Nuevo Ingreso"/>		

Para verificar el Stock daremos click en el check verde y visualizaremos que el stock actual ha cambiado

Almacen

Codigo del Producto:	<input type="text" value="2005"/>			Stock Actual:	<input type="text" value="82"/>
Nombre del Producto:	<input type="text" value="BOTAS"/>		Stock Maximo:	<input type="text" value="1000"/>	
Cantidad a Ingresar:	<input type="text" value="50"/>		Stock Minimo:	<input type="text" value="20"/>	
<input type="button" value="Ingresar Stock"/>			<input type="button" value="Nuevo Ingreso"/>		

El otro botón de Nuevo Ingreso nos limpia toda la pantalla para poder escribir un nuevo código que vayamos a modificar

The screenshot shows a window titled "Almacen". It contains several input fields and buttons. At the top left is a shopping cart icon. To its right is the title "Almacen". On the far right is a close button (X). Below the title are four pairs of fields: "Codigo del Producto:" with an input field, a green checkmark button, a red X button, and a "Stock Actual:" input field; "Nombre del Producto:" with an input field and a "Stock Maximo:" input field; "Cantidad a Ingresar:" with an input field and a "Stock Minimo:" input field. At the bottom are two buttons: a blue "Ingresar Stock" button and a purple "Nuevo Ingreso" button.

- **REPORTES**

Encontraremos una función:



### 1) Generar Reporte

En este ítem tendremos 4 tipos de reporte

#### a) Listado general de ventas

Aquí encontraremos el número de ventas con las siguientes características:

- Número de Venta
- Código de Cliente
- Código de Producto
- Fecha de Venta
- Importe Sub Total
- Importe IGV
- Pago Total

Reportes X

**Seleccione Reporte:** Listado general de ventas Generar Reporte

Pago Total	:413.0
Venta: 40	
Numero de Venta	: 3040
Codigo de Cliente	: 1003
Codigo de Producto	:2001
Fecha de Venta	: 11/01/2023
Importe SubTotal	: 180.0
Importe IGV	:32.4
Pago Total	:212.4
Venta: 41	
Numero de Venta	: 3041
Codigo de Cliente	: 1003
Codigo de Producto	:2001
Fecha de Venta	: 14/06/2023
Importe SubTotal	: 150.0
Importe IGV	:27.0
Pago Total	:177.0

### b) Listado de productos con stock por debajo del stock máximo

En este reporte la GUI nos mostrara los productos en stock con menores al stock máximo, imprimiendo.

- Código del producto
- Nombre del producto
- Stock actual del producto
- Stock minimo del producto

Reportes X

**Seleccione Reporte:** Listado de productos con stock por debajo del stock máximo Generar Reporte

Stock Minimo del Producto: 1	
Producto: 4	
Codigo del Producto: 2004	
Nombre del Producto: TACOS	
Stock Actual del Producto: 12	
Stock Minimo del Producto: 1	
Producto: 5	
Codigo del Producto: 2005	
Nombre del Producto: BOTAS	
Stock Actual del Producto: 82	
Stock Minimo del Producto: 20	
Producto: 7	
Codigo del Producto: 2007	
Nombre del Producto: BAILARINA	
Stock Actual del Producto: 8	
Stock Minimo del Producto: 0	

### c) Listado de productos por unidades vendidas acumuladas

En este reporte la GUI nos mostrara un listado de productos en un listado, la cantidad total de ventas por producto, imprimiendo.

- Código del producto
- Nombre del producto
- Cantidades Acumuladas

The screenshot shows a software window titled "Reportes". At the top, there is a dropdown menu labeled "Seleccione Reporte:" with the option "Listado de productos por unidades vendidas acumuladas" selected. To the right of the dropdown is a "Generar Reporte" button. The main area displays three entries, each representing a product with its details:

- Producto: 5  
Codigo del Producto: 2005  
Nombre del Producto: BOTAS  
Cantidades Acumuladas: 3
- Producto: 6  
Codigo del Producto: 2006  
Nombre del Producto: PANTUFLAS  
Cantidades Acumuladas: 35
- Producto: 7  
Codigo del Producto: 2007  
Nombre del Producto: BAILARINA  
Cantidades Acumuladas: 0

### d) Listado de productos por importe total acumulado

En este reporte la GUI nos mostrara un listado pago total acumulado por cada producto, para asi tener la ganancia total que da cada producto.

- Código del producto
- Nombre del producto
- Cantidad de importe total acumulado

The screenshot shows a software window titled "Reportes". At the top, there is a dropdown menu labeled "Seleccione Reporte:" with the option "Listado de productos por importe total acumulado" selected. To the right of the dropdown is a "Generar Reporte" button. The main area displays three entries, each representing a product with its details:

- Producto: 5  
Codigo del producto: 2005  
Nombre del producto: BOTAS  
Cantidad de importe acumulado: 354.0
- Producto: 6  
Codigo del producto: 2006  
Nombre del producto: PANTUFLAS  
Cantidad de importe acumulado: 1062.0
- Producto: 7  
Codigo del producto: 2007  
Nombre del producto: BAILARINA  
Cantidad de importe acumulado: 0.0

## Entregables

### Cliente.java

```
package clases;

public class Cliente {

    //Atributos Privados
    private int codigo;
    private String nombres, apellidos, direccion, telef, dni;

    //Constructor
    public Cliente(int codigo, String nombres, String apellidos,
String direccion, String telef, String dni) {
        this.codigo = codigo;
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.direccion = direccion;
        this.telef = telef;
        this.dni = dni;
    }

    //Métodos públicos set/get
    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNombres() {
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    public String getTelef() {
        return telef;
    }
}
```

```
public void setTelef(String telef) {
    this.telef = telef;
}

public String getDni() {
    return dni;
}

public void setDni(String dni) {
    this.dni = dni;
}
}

Producto.java
package clases;

public class Producto {
    //Atributos privados
    private int codProducto, stockActual, stockMin, stockMax;
    private String nombre;
    private double precio;

    //Constructor
    public Producto(int codProducto, String nombre, int stockActual,
int stockMin, int stockMax, double precio) {
        this.codProducto = codProducto;
        this.stockActual = stockActual;
        this.stockMin = stockMin;
        this.stockMax = stockMax;
        this.nombre = nombre;
        this.precio = precio;
    }

    //Métodos públicos set/get
    public int getCodProducto() {
        return codProducto;
    }

    public void setCodProducto(int codProducto) {
        this.codProducto = codProducto;
    }

    public int getStockActual() {
        return stockActual;
    }

    public void setStockActual(int stockActual) {
        this.stockActual = stockActual;
    }

    public int getStockMin() {
        return stockMin;
    }

    public void setStockMin(int stockMin) {
        this.stockMin = stockMin;
    }

    public int getStockMax() {
        return stockMax;
    }
}
```

```
public void setStockMax(int stockMax) {
    this.stockMax = stockMax;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public double getPrecio() {
    return precio;
}

public void setPrecio(double precio) {
    this.precio = precio;
}
}
```

**Venta.java**

```
package clases;

public class Venta {
    //Atributos Privados
    private int codVenta, codProducto, codCliente, cantVenta;
    private String fechaVenta;
    private double precioVenta;

    //Constructor
    public Venta(int codVenta, int codProducto, int codCliente, int cantVenta, String fechaVenta,
                 double precioVenta) {
        this.codVenta = codVenta;
        this.codProducto = codProducto;
        this.codCliente = codCliente;
        this.cantVenta = cantVenta;
        this.fechaVenta = fechaVenta;
        this.precioVenta = precioVenta;
    }

    //Métodos públicos set/get
    public int getCodVenta() {
        return codVenta;
    }
    public void setCodVenta(int codVenta) {
        this.codVenta = codVenta;
    }
    public int getCodProducto() {
        return codProducto;
    }
    public void setCodProducto(int codProducto) {
        this.codProducto = codProducto;
    }
    public int getCodCliente() {
        return codCliente;
    }
    public void setCodCliente(int codCliente) {
```

```

        this.codCliente = codCliente;
    }
    public int getCantVenta() {
        return cantVenta;
    }
    public void setCantVenta(int cantVenta) {
        this.cantVenta = cantVenta;
    }
    public String getFechaVenta() {
        return fechaVenta;
    }
    public void setFechaVenta(String fechaVenta) {
        this.fechaVenta = fechaVenta;
    }
    public double getPrecioVenta() {
        return precioVenta;
    }
    public void setPrecioVenta(double precioVenta) {
        this.precioVenta = precioVenta;
    }
}
//-----
// Importe SubTotal
public double subTotal() {
    return cantVenta * precioVenta;
}

// Importe del IGV
public double desctIgv() {
    return subTotal()*0.18;
}

// Importe Total a Pagar
public double pagoTotal() {
    return subTotal()+desctIgv();
}
}

```

**ArregloCliente.java**

```

package arreglos;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.ArrayList;

import clases.Cliente;

public class ArregloCliente {

    //Declaramos la lista de Arreglo
    private ArrayList<Cliente> cl;

    //Creamos el constructor del arreglo
    public ArregloCliente() {
        /*Creamos la lista de Arreglo*/
        cl = new ArrayList<Cliente>();
        cargarClientes();
    }
}

```

```
//Método para adicionar objetos al arreglo
public void adicionar (Cliente x) {
    cl.add(x);
    grabarClientes();
}

//Método para calcular el tamaño del arreglo
public int tamanio() {
    return cl.size();
}

//Método para obtener el índice del objeto en el arreglo
public Cliente obtener(int i) {
    return cl.get(i);
}

//Método para buscar un objeto en el arreglo
public Cliente buscar(String dni) {
    for(int i = 0; i<tamanio(); i++) {
        if(dni == obtener(i).getDni()){
            return obtener(i);
        }
    }
    return null;
}

public Cliente buscar(int codigo) {
    for(int i = 0; i<tamanio(); i++) {
        if(codigo == obtener(i).getCodigo()){
            return obtener(i);
        }
    }
    return null;
}

//Método para eliminar un objeto del arreglo
public void eliminar(Cliente x) {
    cl.remove(x);
    grabarClientes();
}

//Método para generar el número correlativo
public int numCorre() {
    if(tamanio() == 0)
        return 1001;
    else
        return obtener(tamanio()-1).getCodigo()+1;
}

public void actualizarLista() {
    grabarClientes();
}

private void grabarClientes() {
    try {
        PrintWriter pw;
        String linea;
        Cliente x;
```

```

        pw = new PrintWriter(new
FileWriter("clientes.txt"));
        for(int i=0; i<tamanio(); i++) {
            x = obtener(i);
            linea = x.getCodigo() + ";" +
                x.getNombres() + ";" +
                x.getApellidos() + ";" +
                x.getDireccion() + ";" +
                x.getTelef() + ";" +
                x.getDni();
            pw.println(linea);
        }
        pw.close();
    }
    catch (Exception e) {
    }
}

private void cargarClientes() {
    try {
        BufferedReader br;
        String linea, nombres, apellido, direccion, telef,
dni;
        String[] s;
        int codigo;
        br = new BufferedReader(new
FileReader("clientes.txt"));
        while ((linea=br.readLine()) != null) {
            s = linea.split(";");
            codigo = Integer.parseInt(s[0].trim());
            nombres = s[1];
            apellido = s[2];
            direccion = s[3];
            telef = s[4];
            dni = s[5];
            adicionar(new Cliente(codigo, nombres,
apellido, direccion, telef, dni));
        }
        br.close();
    }
    catch (Exception e) {
    }
}
}

```

**ArregloProducto.java**

```

package arreglos;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.ArrayList;

import clases.Cliente;
import clases.Producto;

public class ArregloProductos {
    //Declaramos el ArrayList
    private ArrayList<Producto> pro;

```

```
//Constructor del Arreglo
public ArregloProductos() {
    pro = new ArrayList<Producto>();
    cargarProductos();}

//Método Adicionar
public void adicionar(Producto x) {
    pro.add(x);
    grabarProductos();}

//Método Tamaño
public int tamanio() {
    return pro.size();
}

//Método Obtener
public Producto obtener(int i) {
    return pro.get(i);
}

//Método Buscar
public Producto buscar (int codigo) {
    for(int i = 0; i<tamanio(); i++) {
        if (codigo == obtener(i).getCodProducto()) {
            return obtener(i);
        }
    }
    return null;
}

//Método Eliminar
public void eliminar(Producto x) {
    pro.remove(x);
    grabarProductos();}

//Actualizar Archivo
public void actualizarArchivo() {
    grabarProductos();
}

//Método Generar Número Correlativo
public int numCorrelativo() {
if(tamanio() == 0)
    return 2001;
else
    return obtener(tamanio()-1).getCodProducto()+1 ; }

//Datos a Grabar

private void grabarProductos() {
    try {
        PrintWriter pw;
        String linea;
        Producto x;
        pw = new PrintWriter (new
FileWriter("Productos.txt"));
        for (int i=0; i<tamanio(); i++) {
            x = obtener(i);
            linea = x.getCodProducto()+";" +
            linea;
        }
        pw.println(linea);
        pw.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
        x.getNombre() + ";" +
        x.getPrecio() + ";" +
        x.getStockActual() + ";" +
        x.getStockMax() + ";" +
        x.getStockMin();
    pw.println(linea);
}
pw.close();

}
catch (Exception e) {

}

}
//Datos Cargar
private void cargarProductos() {
    try {
        BufferedReader br;
        String linea;
        String[] s;
        double precio;
        String nombre;
        int codigo, StockActual, StockMax, StockMin;
        br = new BufferedReader(new
FileReader("Productos.txt"));
        while((linea = br.readLine()) !=null) {
            s= linea.split(";");
            codigo = Integer.parseInt(s[0].trim());
            nombre= (s[1].trim());
            precio =
Double.parseDouble(s[2].trim());
            StockActual =
Integer.parseInt(s[3].trim());
            StockMin =
Integer.parseInt(s[4].trim());
            StockMax =
Integer.parseInt(s[5].trim());

                adicionar(new Producto(codigo,nombre,
StockActual, StockMin, StockMax, precio ));
        }
        br.close();
    }
    catch (Exception e) {

}
}
}
```

### ArregloVentas.java

```
package arreglos;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.ArrayList;

import clases.Cliente;
import clases.Producto;
```

```
public class ArregloProductos {
    //Declaramos el ArrayList
    private ArrayList<Producto> pro;

    //Constructor del Arreglo
    public ArregloProductos() {
        pro = new ArrayList<Producto>();
        cargarProductos();
    }

    //Método Adicionar
    public void adicionar(Producto x) {
        pro.add(x);
        grabarProductos();
    }

    //Método Tamaño
    public int tamano() {
        return pro.size();
    }

    //Método Obtener
    public Producto obtener(int i) {
        return pro.get(i);
    }

    //Método Buscar
    public Producto buscar (int codigo) {
        for(int i = 0; i<tamano(); i++) {
            if (codigo == obtener(i).getCodProducto()) {
                return obtener(i);
            }
        }
        return null;
    }

    //Método Eliminar
    public void eliminar(Producto x) {
        pro.remove(x);
        grabarProductos();
    }

    //Actualizar Archivo
    public void actualizarArchivo() {
        grabarProductos();
    }

    //Método Generar Número Correlativo
    public int numCorrelativo() {
        if(tamano() == 0)
            return 2001;
        else
            return obtener(tamano()-1).getCodProducto()+1 ;
    }

    //Datos a Grabar

    private void grabarProductos() {
        try {
            PrintWriter pw;
            String linea;
            Producto x;
```

```

        pw = new PrintWriter (new
FileWriter("Productos.txt"));
    for (int i=0; i<tamanio(); i++) {
        x = obtener(i);
        linea = x.getCodProducto()+";" +
            x.getNombre()+";" +
            x.getPrecio()+";" +
            x.getStockActual()+";" +
            x.getStockMax()+";" +
            x.getStockMin();
        pw.println(linea);
    }
    pw.close();
}

}
catch (Exception e) {

}

}
//Datos Cargar
private void cargarProductos() {
try {
    BufferedReader br;
    String linea;
    String[] s;
    double precio;
    String nombre;
    int codigo, StockActual, StockMax, StockMin;
    br = new BufferedReader(new
FileReader("Productos.txt"));
    while((linea = br.readLine()) !=null) {
        s= linea.split(";");
        codigo = Integer.parseInt(s[0].trim());
        nombre= (s[1].trim());
        precio =
Double.parseDouble(s[2].trim());
        StockActual =
Integer.parseInt(s[3].trim());
        StockMin =
Integer.parseInt(s[4].trim());
        StockMax =
Integer.parseInt(s[5].trim());

        adicionar(new Producto(codigo,nombre,
StockActual, StockMin, StockMax, precio ));
    }
    br.close();
}
catch (Exception e) {

}
}
}

```

**Fecha.java**

```

package libreria;

import java.util.Calendar;
import java.util.GregorianCalendar;

```

```
public class Fecha {

    // Métodos static que retornan valor (sin parámetros)
    public static String fecha() {
        int ds;
        String dia;
        Calendar c = new GregorianCalendar();
        ds = c.get(Calendar.DAY_OF_WEEK);
        switch (ds) {
            case 1: dia = "Domingo"; break;
            case 2: dia = "Lunes"; break;
            case 3: dia = "Martes"; break;
            case 4: dia = "Miércoles"; break;
            case 5: dia = "Jueves"; break;
            case 6: dia = "Viernes"; break;
            default: dia = "Sábado"; break;
        }
        return dia + ", " + enTextoFecha(fechaActual());
    }

    public static String fechaActual() {
        int dd, mm, aa;
        Calendar c = new GregorianCalendar();
        dd = c.get(Calendar.DAY_OF_MONTH);
        mm = c.get(Calendar.MONTH) + 1;
        aa = c.get(Calendar.YEAR);
        return ajustar(dd) + "/" + ajustar(mm) + "/" + aa;
    }

    // Métodos static que retornan valor (con parámetros)
    public static String enTextoFecha(String fecha) {
        String[] s = fecha.split("/");
        if (s.length == 3)
            return s[0] + " de " + obtenerMes(s[1]) + " de " +
s[2];
        else
            return "";
    }

    public static String obtenerMes(String mes) {
        switch (mes) {
            case "01": return "Enero";
            case "02": return "Febrero";
            case "03": return "Marzo";
            case "04": return "Abril";
            case "05": return "Mayo";
            case "06": return "Junio";
            case "07": return "Julio";
            case "08": return "Agosto";
            case "09": return "Setiembre";
            case "10": return "Octubre";
            case "11": return "Noviembre";
            default: return "Diciembre";
        }
    }

    public static String ajustar(int numero) {
        return String.format("%02d", numero);
    }
}
```

**DlgCliente.java**

```
package principal;

import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import arreglos.ArregloCliente;
import clases.Cliente;

import javax.swing.table.DefaultTableModel;

import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.ActionEvent;
import javax.swing.ImageIcon;
import javax.swing.SwingConstants;
import java.awt.event.MouseListener;
import java.awt.Toolkit;

public class DlgCliente extends JDialog implements ActionListener,
MouseListener {

    private static final long serialVersionUID = 1L;

    private final JPanel contentPanel = new JPanel();
    private JLabel lblCodigo;
    private JLabel lblNombres;
    private JLabel lblApellidos;
    private JLabel lblDireccion;
    private JLabel lblTelefono;
    private JLabel lblDni;
    private JTextField txtDNI;
    private JTextField txtNombres;
    private JTextField txtApellidos;
    private JTextField txtTelefono;
    private JTextField txtDireccion;
    private JTextField txtCodigo;
    private JButton btnIngresar;
    private JButton btnModificar;
    private JButton btnConsultar;
    private JButton btnEliminar;
    private JButton btnListar;
    private JScrollPane scpTabla;
    private DefaultTableModel modelo;
    private JTable tblTabla;
```

```
/*
 * Launch the application.
 */
public static void main(String[] args) {
    try {
        DlgCliente dialog = new DlgCliente();

        dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        dialog.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// Variable Global
private int tipoOperacion;

// Constantes para los tipos de operaciones
public final static int INGRESAR = 0;
public final static int MODIFICAR = 1;
public final static int CONSULTAR = 2;
public final static int ELIMINAR = 3;

/*
 * Create the dialog.
 */
public DlgCliente() {

    setIconImage(Toolkit.getDefaultToolkit().getImage(DlgCliente.class.getResource("/iconos/usuarios.png")));
    setResizable(false);
    setTitle("Sistema de Ventas | Cliente");
    setBounds(100, 100, 743, 431);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);

    lblCodigo = new JLabel("Codigo :");
    lblCodigo.setBounds(21, 13, 70, 14);
    lblCodigo.setFont(new Font("Comic Sans MS", Font.BOLD,
12));
    lblCodigo.setForeground(Color.BLACK);
    contentPanel.add(lblCodigo);

    lblNombres = new JLabel("Nombres :");
    lblNombres.setBounds(21, 63, 70, 14);
    lblNombres.setFont(new Font("Comic Sans MS", Font.BOLD,
12));
    lblNombres.setForeground(Color.BLACK);
    contentPanel.add(lblNombres);

    lblApellidos = new JLabel("Apellidos :");
    lblApellidos.setBounds(21, 88, 70, 14);
    lblApellidos.setFont(new Font("Comic Sans MS", Font.BOLD,
12));
    lblApellidos.setForeground(Color.BLACK);
    contentPanel.add(lblApellidos);

    lblDireccion = new JLabel("Dirección :");
    lblDireccion.setBounds(21, 138, 70, 14);
    contentPanel.add(lblDireccion);
}
```

```
lblDireccion.setFont(new Font("Comic Sans MS", Font.BOLD,
12));
lblDireccion.setForeground(Color.BLACK);
contentPanel.add(lblDireccion);

lblTelefono = new JLabel("Teléfono :");
lblTelefono.setBounds(21, 113, 70, 14);
lblTelefono.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
lblTelefono.setForeground(Color.BLACK);
contentPanel.add(lblTelefono);

lblDni = new JLabel("Nro DNI :");
lblDni.setBounds(21, 38, 70, 14);
lblDni.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
lblDni.setForeground(Color.BLACK);
contentPanel.add(lblDni);

txtCodigo = new JTextField();
txtCodigo.setBounds(101, 11, 103, 20);
contentPanel.add(txtCodigo);
txtCodigo.setColumns(10);

txtDNI = new JTextField();
txtDNI.setBounds(101, 34, 103, 20);
txtDNI.setColumns(10);
contentPanel.add(txtDNI);

txtNombres = new JTextField();
txtNombres.setBounds(101, 57, 103, 20);
txtNombres.setColumns(10);
contentPanel.add(txtNombres);

txtApellidos = new JTextField();
txtApellidos.setBounds(101, 82, 103, 20);
txtApellidos.setColumns(10);
contentPanel.add(txtApellidos);

txtTelefono = new JTextField();
txtTelefono.setBounds(101, 106, 103, 20);
txtTelefono.setColumns(10);
contentPanel.add(txtTelefono);

txtDireccion = new JTextField();
txtDireccion.setBounds(101, 131, 103, 20);
txtDireccion.setColumns(10);
contentPanel.add(txtDireccion);

btnIngresar = new JButton("Ingresar");
btnIngresar.setBounds(613, 38, 89, 23);
btnIngresar.addActionListener(this);
contentPanel.add(btnIngresar);

btnModificar = new JButton("Modificar");
btnModificar.setBounds(613, 63, 89, 23);
btnModificar.addActionListener(this);
contentPanel.add(btnModificar);

btnConsultar = new JButton("Consultar");
btnConsultar.setBounds(613, 88, 89, 23);
btnConsultar.addActionListener(this);
```

```
contentPanel.add(btnConsultar);

btnEliminar = new JButton("Eliminar");
btnEliminar.setBounds(613, 113, 89, 23);
btnEliminar.addActionListener(this);
contentPanel.add(btnEliminar);

btnListar = new JButton("Listar");
btnListar.setBounds(613, 13, 89, 23);
btnListar.addActionListener(this);
contentPanel.add(btnListar);

scpTabla = new JScrollPane();
scpTabla.setBounds(10, 167, 707, 219);
contentPanel.add(scpTabla);

tblTabla = new JTable();
tblTabla.addMouseListener(this);
tblTabla.setFillsViewportHeight(true);
scpTabla.setViewportView(tblTabla);

btnAceptar = new JButton("Aceptar");
btnAceptar.addActionListener(this);
btnAceptar.setBounds(225, 9, 89, 23);
contentPanel.add(btnAceptar);

modelo = new DefaultTableModel();
modelo.addColumn("Código");
modelo.addColumn("Nombre");
modelo.addColumn("Apellido");
modelo.addColumn("Dirección");
modelo.addColumn("Teléfono");
modelo.addColumn("DNI");
tblTabla.setModel(modelo);

modelo.setRowCount(0);

txtCodigo.setEnabled(false);
btnAceptar.setEnabled(false);

lblNewLabel = new JLabel("");
lblNewLabel.setIcon(new ImageIcon("D:\\INSTITUTO\\CIBERTEC\\2_CICLO\\ALGORITMOS Y ESTRUCTURA
DE DATOS\\PROYECTO\\Proyecto 2-AED -Ventas\\Sistema De
Ventas\\bin\\iconos\\fondozapato.jpg"));
lblNewLabel.setBounds(-10, -41, 737, 486);
contentPanel.add(lblNewLabel);
//limpieza();
listar(cli.tamanio()-1);
//editarFila();
habilitarCampos(false);
//habilitarBotones(false);

setLocationRelativeTo(null);
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btnAceptar) {
        actionPerformedBtnAceptar(e);
    }
}
```



```
        Cliente cl = new
Cliente(codigo, nombre, apellido, direccion, telefono, dni);

        cli.adicionar(cl);

        btnIngresar.setEnabled(true);

        listar(tblTabla.getRowCount());
                limpieza();

        habilitarCampos(false);

        btnAceptar.setEnabled(false);
    } else {
        error("El DNI "
+ dni + " ya existe", txtDNI);
    }

    // Boton MODIFICAR
if(btnModificar.isEnabled()
== false) {
    cli.buscar(codigo);
    Cliente x =
x.setNombres(nombre);

    x.setApellidos(apellido);

    x.setDireccion(direccion);
    x.setTelef(telefono);
    x.setDni(dni);
    cli.actualizarLista();

    btnModificar.setEnabled(true);

    btnAceptar.setEnabled(false);

    listar(tblTabla.getSelectedRow());
    habilitarCampos(false);
    limpiezaGeneral();
}

} else {
    error("La dirección
ingresada es incorrecta", txtDireccion);
}

} else {
    error("El Teléfono ingresado es
incorrecto", txtTelefono);
}

} else {
    error("El apellido ingresado es
incorrecto", txtApellidos);
}

} else {
    error("El nombre ingresado es incorrecto",
txtNombres);
```

```
        }
    } else {
        error("El número del DNI es incorrecto", txtDNI);
    }
}

//Método para listar toda la información de los clientes
protected void actionPerformedBtnListar(ActionEvent e){
    if(tblTabla.getRowCount() == 0) {
        JOptionPane.showMessageDialog(this,"No hay lista de
clientes para mostrar. Ingrese un nuevo Cliente.", "Información", 1);
    } else {
        JOptionPane.showMessageDialog(this,"No hay nuevos
clientes para mostrar.", "Información", 1);
    }
}

protected void actionPerformedBtnIngresar(ActionEvent e) {
    habilitarCampos(true);
    btnIngresar.setEnabled(false);
    btnAceptar.setEnabled(true);
    limpieza();
    txtDNI.requestFocus();
}

protected void actionPerformedBtnModificar(ActionEvent e){
    //String xx = Integer.toString(leerCodigo());
    if(leerDni().length() > 0) {
        habilitarCampos(true);
        btnModificar.setEnabled(false);
        btnAceptar.setEnabled(true);
        txtDNI.requestFocus();
    } else {
        error("Seleccione en la tabla al cliente a
modificar", txtCodigo);
    }
}

protected void actionPerformedBtnConsultar(ActionEvent e) {
    habilitarCampos(false);
    limpiezaGeneral();
    btnAceptar.setEnabled(true);
    btnConsultar.setEnabled(false);
    mensaje("Ingresa el código que deseas consultar");
    txtCodigo.setEnabled(true);
    txtCodigo.requestFocus();
}

protected void actionPerformedBtnEliminar(ActionEvent e){
    if(cli.tamanio() == 0) {
        btnEliminar.setEnabled(false);
        mensaje("No existen clientes para eliminar");
    } else {
        editarFila();
        int ok = confirmar("¿ Desea eliminar al cliente ?");
        if(ok == 0) {
            Cliente x = cli.buscar(leerCodigo());
            int posFila = tblTabla.getSelectedRow();
            if(posFila == cli.tamanio()-1)
                posFila--;
            cli.eliminar(x);
        }
    }
}
```

```
        listar(posFila);
        editarFila();
    }
    btnAceptar.setEnabled(false);
    limpiezaGeneral();
}
}

void limpieza() {
    txtCodigo.setText(" " + cli.num_corre());
    txtNombres.setText("");
    txtApellidos.setText("");
    txtDireccion.setText("");
    txtTelefono.setText("");
    txtDNI.setText("");
}

void limpiezaGeneral() {
    txtCodigo.setText("");
    txtNombres.setText("");
    txtApellidos.setText("");
    txtDireccion.setText("");
    txtTelefono.setText("");
    txtDNI.setText("");
}

void editarFila() {
    if(cli.tamanio() == 0)
        limpieza();
    else {
        Cliente x = cli.obtener(tblTabla.getSelectedRow());
        txtCodigo.setText(" " + x.getCodigo());
        txtDNI.setText(" " + x.getDni());
        txtNombres.setText(x.getNombres());
        txtApellidos.setText(x.getApellidos());
        txtTelefono.setText(x.getTelef());
        txtDireccion.setText(x.getDireccion());
    }
}

void listar(int posFila) {
    modelo.setRowCount(0);
    Cliente x;
    for(int i=0; i<cli.tamanio(); i++) {
        x = cli.obtener(i);
        Object[] fila = {
            x.getCodigo(),
            x.getNombres(),
            x.getApellidos(),
            x.getDireccion(),
            x.getTelef(),
            x.getDni()
        };
        modelo.addRow(fila);
    }
    tblTabla.getSelectionModel().setSelectionInterval(posFila,
posFila);
}

void mensaje(String s) {
```

```
        JOptionPane.showMessageDialog(this, s, "Mensaje", 1);
    }

    void error(String s, JTextField txt) {
        mensaje(s);
        txt.setText("");
        txt.requestFocus();
    }

    void habilitarCampos(boolean estado) {
        txtNombres.setEnabled(estado);
        txtApellidos.setEnabled(estado);
        txtDireccion.setEnabled(estado);
        txtTelefono.setEnabled(estado);
        txtDNI.setEnabled(estado);
    }

    void habilitarBotones(boolean estado) {
        btnIngresar.setEnabled(estado);
        btnModificar.setEnabled(estado);
        btnConsultar.setEnabled(estado);
        btnEliminar.setEnabled(estado);
    }

    public void mouseClicked(MouseEvent e) {
        if (e.getSource() == tblTabla) {
            mouseClickedTblTabla(e);
        }
    }

    public void mouseEntered(MouseEvent e) {}

    public void mouseExited(MouseEvent e) {}

    public void mousePressed(MouseEvent e) {}

    public void mouseReleased(MouseEvent e) {}

    protected void mouseClickedTblTabla(MouseEvent e) {
        habilitarCampos(false);
        habilitarBotones(true);
        editarFila();
    }

    int leerCodigo() {
        return Integer.parseInt(txtCodigo.getText().trim());
    }

    String leerDni() {
        return txtDNI.getText().trim();
    }

    String leerNombre() {
        return txtNombres.getText().trim();
    }

    String leerApellidos() {
        return txtApellidos.getText().trim();
    }

    String leerTelefono() {
        return txtTelefono.getText().trim();
    }
```

```
    }

    String leerDireccion() {
        return txtDireccion.getText().trim();
    }

    int confirmar(String s) {
        return JOptionPane.showConfirmDialog(this, s, "Alerta", 0,
1, null);
    }
}
```

### DlgProducto.java

```
package principal;

import clases.*;
import arreglos.*;
import java.awt.Cursor;
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.ImageIcon;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Toolkit;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;

public class DlgProducto extends JDialog implements ActionListener,
MouseListener {

    private static final long serialVersionUID = 1L;
    private JPanel contentPanel = new JPanel();
    private JTextField txtCodigo;
    private JTextField txtNombre;
    private JTextField txtPrecio;
    private JTextField txtStockActual;
    private JTextField txtStockMin;
    private JTextField txtStockMax;
    private JTable tblTabla;
    private JButton btnConsultar;
    private JButton btnIngresar;
    private JButton btnEliminar;
    private JButton btnModificar;
    private JButton btnListar;
    private DefaultTableModel modelo;

    /**
     * Launch the application.

```

```
/*
public static void main(String[] args) {
    try {
        DlgProducto dialog = new DlgProducto();

        dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        dialog.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * Create the dialog.
 */
public DlgProducto() {

    setIconImage(Toolkit.getDefaultToolkit().getImage(DlgProducto.class.getResource("/iconos/zapatos_grande.png")));
    setResizable(false);
    setTitle("Sistema de Ventas | Producto");
    setBounds(100, 100, 737, 433);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);
    {

        JLabel lblNewLabel = new JLabel("Codigo :");
        lblNewLabel.setForeground(Color.BLACK);
        lblNewLabel.setFont(new Font("Comic Sans MS",
Font.BOLD, 12));
        lblNewLabel.setBounds(46, 14, 70, 14);
        contentPanel.add(lblNewLabel);
    }
    {

        JLabel lblNombre = new JLabel("Nombre :");
        lblNombre.setForeground(Color.BLACK);
        lblNombre.setFont(new Font("Comic Sans MS",
Font.BOLD, 12));
        lblNombre.setBounds(46, 37, 70, 14);
        contentPanel.add(lblNombre);
    }
    {

        JLabel lblPrecio = new JLabel("Precio : ");
        lblPrecio.setForeground(Color.BLACK);
        lblPrecio.setFont(new Font("Comic Sans MS",
Font.BOLD, 12));
        lblPrecio.setBounds(46, 60, 70, 14);
        contentPanel.add(lblPrecio);
    }
    {

        JLabel lblStockActual = new JLabel("Stock Actual
:");
        lblStockActual.setForeground(Color.BLACK);
        lblStockActual.setFont(new Font("Comic Sans MS",
Font.BOLD, 12));
        lblStockActual.setBounds(46, 85, 92, 14);
        contentPanel.add(lblStockActual);
    }
}
```

```
        JLabel lblStockMinimo = new JLabel("Stock Mínimo  
:");  
        lblStockMinimo.setForeground(Color.BLACK);  
        lblStockMinimo.setFont(new Font("Comic Sans MS",  
Font.BOLD, 12));  
        lblStockMinimo.setBounds(46, 109, 92, 14);  
        contentPanel.add(lblStockMinimo);  
    }  
    {  
        JLabel lblStockMaximo = new JLabel("Stock Máximo  
:");  
        lblStockMaximo.setForeground(Color.BLACK);  
        lblStockMaximo.setFont(new Font("Comic Sans MS",  
Font.BOLD, 12));  
        lblStockMaximo.setBounds(46, 134, 103, 14);  
        contentPanel.add(lblStockMaximo);  
    }  
    {  
        txtCodigo = new JTextField();  
        txtCodigo.setEditable(false);  
        txtCodigo.setColumns(10);  
        txtCodigo.setBounds(148, 14, 103, 20);  
        contentPanel.add(txtCodigo);  
    }  
    {  
        txtNombre = new JTextField();  
        txtNombre.setEditable(false);  
        txtNombre.setColumns(10);  
        txtNombre.setBounds(148, 37, 103, 20);  
        contentPanel.add(txtNombre);  
    }  
    {  
        txtPrecio = new JTextField();  
        txtPrecio.setEditable(false);  
        txtPrecio.setColumns(10);  
        txtPrecio.setBounds(148, 60, 103, 20);  
        contentPanel.add(txtPrecio);  
    }  
    {  
        txtStockActual = new JTextField();  
        txtStockActual.setEditable(false);  
        txtStockActual.setColumns(10);  
        txtStockActual.setBounds(148, 85, 103, 20);  
        contentPanel.add(txtStockActual);  
    }  
    {  
        txtStockMin = new JTextField();  
        txtStockMin.setEditable(false);  
        txtStockMin.setColumns(10);  
        txtStockMin.setBounds(148, 109, 103, 20);  
        contentPanel.add(txtStockMin);  
    }  
    {  
        txtStockMax = new JTextField();  
        txtStockMax.setEditable(false);  
        txtStockMax.setColumns(10);  
        txtStockMax.setBounds(148, 133, 103, 20);  
        contentPanel.add(txtStockMax);  
    }  
    {  
        btnListar = new JButton("Listar");  
    }
```

```
        btnListar.addActionListener(this);
        btnListar.setEnabled(false);
        btnListar.setBounds(282, 131, 89, 23);
        contentPanel.add(btnListar);
    }
}

btnIngresar = new JButton("Ingresar");
btnIngresar.addActionListener(this);
btnIngresar.setBounds(615, 25, 89, 23);
contentPanel.add(btnIngresar);
}

btnModificar = new JButton("Modificar");
btnModificar.addActionListener(this);
btnModificar.setBounds(615, 50, 89, 23);
contentPanel.add(btnModificar);
}

btnConsultar = new JButton("Consultar");
btnConsultar.addActionListener(this);
btnConsultar.setBounds(615, 75, 89, 23);
contentPanel.add(btnConsultar);
}

btnEliminar = new JButton("Eliminar");
btnEliminar.addActionListener(this);
btnEliminar.setBounds(615, 100, 89, 23);
contentPanel.add(btnEliminar);
}

JScrollPane scpTabla = new JScrollPane();
scpTabla.setBounds(10, 159, 699, 219);
contentPanel.add(scpTabla);
{
    tblTabla = new JTable();
    tblTabla.setSurrendersFocusOnKeystroke(true);
    tblTabla.setFillsViewportHeight(true);
    tblTabla.addMouseListener(this);
    scpTabla.setViewportView(tblTabla);
}
}

modelo = new DefaultTableModel();
modelo.addColumn("CÓDIGO");
modelo.addColumn("NOMBRE");
modelo.addColumn("PRECIO");
modelo.addColumn("STOCK ACTUAL");
modelo.addColumn("STOCK MIN");
modelo.addColumn("STOCK MAX");

tblTabla.setModel(modelo);

txtCodigo.setEditable(false);

lblcheck = new JLabel("");
lblcheck.addMouseListener(this);
lblcheck.setIcon(new ImageIcon(DlgProducto.class.getResource("/iconos/check_icon_24.png")));
;
lblcheck.setBounds(274, 0, 48, 57);
contentPanel.add(lblcheck);
```

```
{  
    lblNewLabel_1 = new JLabel("");  
    lblNewLabel_1.setIcon(new  
ImageIcon(DlgProducto.class.getResource("/iconos/fondozapato.jpg")));  
    lblNewLabel_1.setBounds(0, 0, 721, 393);  
    contentPanel.add(lblNewLabel_1);  
}  
habilitarEntradas(false);  
listar();  
habilitarEntradas(false);  
}  
  
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == btnListar) {  
        do_btnListarActionPerformed(e);  
    }  
    if (e.getSource() == btnModificar) {  
        do_btnModificarActionPerformed(e);  
    }  
    if (e.getSource() == btnEliminar) {  
        do_btnEliminarActionPerformed(e);  
    }  
    if (e.getSource() == btnIngresar) {  
        do_btnIngresarActionPerformed(e);  
    }  
    if (e.getSource() == btnConsultar) {  
        do_btnConsultarActionPerformed(e);  
    }  
}  
  
//Declarar Arreglo Productos  
ArregloProductos ap = new ArregloProductos();  
private JLabel lblcheck;  
private JLabel lblNewLabel_1;  
  
protected void do_btnConsultarActionPerformed(ActionEvent e)  
{  
    btnConsultar.setEnabled(false);  
    btnIngresar.setEnabled(false);  
    btnModificar.setEnabled(false);  
    btnListar.setEnabled(false);  
    btnEliminar.setEnabled(false);  
    habilitarEntradas(false);  
    txtCodigo.setEditable(true);  
    txtCodigo.requestFocus();  
    limpiarcampos();  
}  
  
protected void do_btnIngresarActionPerformed(ActionEvent e)  
{  
    btnIngresar.setEnabled(false);  
    btnModificar.setEnabled(true);  
    btnConsultar.setEnabled(true);  
    btnListar.setEnabled(true);  
    btnEliminar.setEnabled(true);  
    limpieza();  
    habilitarEntradas(true);  
    txtNombre.requestFocus();  
}  
protected void do_btnEliminarActionPerformed(ActionEvent e)  
{
```

```

        btnIngresar.setEnabled(true);
        btnModificar.setEnabled(true);
        btnListar.setEnabled(false);
        if(ap.tamanio()==0)
            mensaje("NO EXISTEN PRODUCTOS");
        else {
            editarFila();
            habilitarEntradas(false);
            int ok = confirmar ("¿DESEAS ELIMINAR ESTE
REGISTRO?");
            if (ok ==0) {
                ap.eliminar(ap.buscar(leerCodigo()));
                listar();
                editarFila();
            }
        }
    }
protected void do_btnModificarActionPerformed(ActionEvent e)
{
    btnIngresar.setEnabled(true);
    btnModificar.setEnabled(false);
    if(ap.tamanio() == 0) {
        btnListar.setEnabled(false);
        habilitarEntradas(false);
        mensaje("NO EXISTEN PRODUCTOS");
    }
    else {
        editarFila();
        btnListar.setEnabled(true);
        habilitarEntradas(true);
        txtNombre.requestFocus();
    }
}

protected void do_btnListarActionPerformed(ActionEvent arg0)
{
    int codigo = leerCodigo();
    String nombre = leerNombre();
    if(nombre.length()>0) {
        try {
            double precio = leerPrecio();
            try {
                int StockActual = leerStockActual();
                try {
                    int StockMin = leerStockMin();
                    try {
                        int StockMax = leerStockMax();

if(btnIngresar.isEnabled()==false) {
                    Producto nuevo = new
Producto(codigo, nombre, StockActual, StockMin, StockMax, precio);
                    ap.adicionar(nuevo);
                }
                btnIngresar.setEnabled(true);
            }
        }
        if(btnModificar.isEnabled()==false) {
            Producto p =
ap.buscar(codigo);
            p.setNombre(nombre);
            p.setPrecio(precio);
        }
    }
}

```

```
        p.setStockActual(StockActual);

        p.setStockMin(StockMin);

        p.setStockMax(StockMax);

        ap.actualizarArchivo();

        btnModificar.setEnabled(true);

    }

    listar();
    habilitarEntradas(false);
}

catch (Exception e) {
    error("INGRESE STOCK MAXIMO", txtStockMax);
}

catch (Exception e) {
    error("INGRESE STOCK MINIMO",
txtStockMin);
}

catch (Exception e) {
    error("INGRESE STOCK ACTUAL",
txtStockActual);
}

catch (Exception e) {
    error("INGRESE PRECIO", txtPrecio);
}

else
    error("INGRESE NOMBRE ", txtNombre);
}

void listar() {
    int posFila = 0;
    if (modelo.getRowCount() > 0)
        posFila = tblTabla.getSelectedRow();
    if (modelo.getRowCount() == ap.tamanio() - 1)
        posFila = ap.tamanio() - 1;
    if (posFila == ap.tamanio())
        posFila--;
    modelo.setRowCount(0);
    Producto p;
    for (int i=0; i<ap.tamanio(); i++) {
        p = ap.obtener(i);
        Object[] fila = { p.getCodProducto(),
                          p.getNombre(),
                          p.getPrecio(),
                          p.getStockActual(),
                          p.getStockMin(),
                          p.getStockMax() };
        modelo.addRow(fila);
    }
    if (ap.tamanio() > 0)

        tblTabla.getSelectionModel().setSelectionInterval(posFila,
posFila);
```

```
}

void editarFila() {
    if (ap.tamanio() == 0)
        limpieza();
    else {
        Producto p = ap.obtener(tblTabla.getSelectedRow());
        txtCodigo.setText(""+ p.getCodProducto());
        txtNombre.setText(p.getNombre());
        txtPrecio.setText(""+p.getPrecio());
        txtStockActual.setText(""+p.getStockActual());
        txtStockMin.setText(""+p.getStockMin());
        txtStockMax.setText(""+p.getStockMax());
    }
}

void limpiarcampos() {
    txtCodigo.setText("");
    txtNombre.setText("");
    txtPrecio.setText("");
    txtStockActual.setText("");
    txtStockMin.setText("");
    txtStockMax.setText("");
}

void limpieza() {
    txtCodigo.setText(""+ ap.numCorrelativo());
    txtNombre.setText("");
    txtPrecio.setText("");
    txtStockActual.setText("");
    txtStockMin.setText("");
    txtStockMax.setText("");
}

// Métodos tipo void (con parámetros)
void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Información", 0);
}
//METODO PARA CONFIRMAR UNA ELIMINACION
int confirmar(String s) {
    return JOptionPane.showConfirmDialog(this, s, "Alerta", 0,
1, null);
}
void error(String s, JTextField txt) {
    mensaje(s);
    txt.setText("");
    txt.requestFocus();
}
/*void habilitarCodigo(boolean sino) {
    txtCodigo.setEditable(sino);
}*/
void habilitarEntradas(boolean sino) {
    txtNombre.setEditable(sino);
    txtPrecio.setEditable(sino);
    txtStockActual.setEditable(sino);
    txtStockMin.setEditable(sino);
    txtStockMax.setEditable(sino);
}
void habilitarBotones(boolean sino) {
    btnIngresar.setEnabled(sino);
    btnModificar.setEnabled(sino);
    btnEliminar.setEnabled(sino);
```

```
        }
    // Métodos que retornan valor (sin parámetros)
    int leerCodigo() {
        return Integer.parseInt(txtCodigo.getText().trim());
    }
    String leerNombre() {
        return txtNombre.getText().trim();
    }
    double leerPrecio() {
        return Double.parseDouble(txtPrecio.getText().trim());
    }
    int leerStockActual() {
        return Integer.parseInt(txtStockActual.getText().trim());
    }
    int leerStockMin() {
        return Integer.parseInt(txtStockMin.getText().trim());
    }
    int leerStockMax() {
        return Integer.parseInt(txtStockMax.getText().trim());
    }

    //Boton Check
    public void mouseClicked(MouseEvent e) {
        if (e.getSource() == tblTabla) {
            do_tblTabla_mouseClicked(e);
        }
        if (e.getSource() == lblcheck) {
            do_lblcheck_mouseClicked(e);
        }
    }
    public void mouseEntered(MouseEvent e) {
        lblcheck.setCursor(new Cursor(Cursor.HAND_CURSOR));
    }
    public void mouseExited(MouseEvent e) {
    }
    public void mousePressed(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e) {
    }

    protected void do_tblTabla_mouseClicked(MouseEvent e) {
        habilitarEntradas(false);
        habilitarBotones(true);
        editarFila();
    }

    protected void do_lblcheck_mouseClicked(MouseEvent e) {
        int codigo = leerCodigo();
        if(btnConsultar.isEnabled() == false) {
            btnConsultar.setEnabled(true);
            btnIngresar.setEnabled(true);
            btnModificar.setEnabled(true);
            btnEliminar.setEnabled(true);
            Producto p = ap.buscar(codigo);
            if (p != null) {
                txtNombre.setText(p.getNombre());
                txtPrecio.setText(""+p.getPrecio());
                txtStockActual.setText(""+p.getStockActual());
                txtStockMin.setText(""+p.getStockMin());
                txtStockMax.setText(""+p.getStockMax());
            }
        }
    }
}
```

```
        else {
            mensaje("EL CODIGO NO EXISTE");
            limpieza();
        }
    }
}
```

**DlgVentas.java**

```
package principal;

import clases.*;
import arreglos.*;
import libreria.*;
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstantsConstants;
import java.awt.Color;
import java.awt.Cursor;

import javax.swing.ImageIcon;
import javax.swing.JSeparator;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.awt.Toolkit;

public class DlgVentas extends JDialog implements ActionListener,
MouseListener {

    private static final long serialVersionUID = 1L;
    private final JPanel contentPanel = new JPanel();
    private JLabel lblCodigoCli;
    private JLabel lblNombreCli;
    private JTextField txtNombreCli;
    private JLabel lblCodigoPro;
    private JLabel lblNombrePro;
    private JLabel lblCantidad;
    private JLabel lblPrecioPro;
    private JTextField txtNombreProd;
    private JTextField txtCantUnid;
    private JTextField txtPrecioUni;
    private JLabel lblNewLabel_7;
    private JTextArea txtS;
    private JScrollPane scrollPane;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_8;
    private JLabel lblNewLabel_9;
```

```
private JLabel lblNewLabel_10;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    try {
        DlgVentas dialog = new DlgVentas();

        dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        dialog.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * Create the dialog.
 */
public DlgVentas() {

    setIconImage(Toolkit.getDefaultToolkit().getImage(DlgVentas.class
            .getResource("/iconos/venta_icon.png")));
    setForeground(new Color(0, 51, 102));
    setTitle("Ventas");
    setBounds(100, 100, 780, 718);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setBackground(new Color(240, 248, 255));
    contentPanel.setForeground(new Color(0, 51, 102));
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);

    lblCodigoCli = new JLabel("Cliente :");
    lblCodigoCli.setForeground(Color.BLACK);

    lblCodigoCli.setHorizontalAlignment(SwingConstants.CENTER);
    lblCodigoCli.setFont(new Font("Tahoma", Font.BOLD, 12));
    lblCodigoCli.setBounds(20, 79, 121, 14);
    contentPanel.add(lblCodigoCli);

    lblNombreCli = new JLabel("Nombre :");
    lblNombreCli.setForeground(Color.BLACK);

    lblNombreCli.setHorizontalAlignment(SwingConstants.CENTER);
    lblNombreCli.setFont(new Font("Tahoma", Font.BOLD, 12));
    lblNombreCli.setBounds(20, 121, 128, 14);
    contentPanel.add(lblNombreCli);

    txtNombreCli = new JTextField();
    txtNombreCli.setEditable(false);
    txtNombreCli.setBounds(145, 121, 103, 20);
    contentPanel.add(txtNombreCli);
    txtNombreCli.setColumns(10);

    lblCodigoPro = new JLabel("Producto :");
    lblCodigoPro.setForeground(Color.BLACK);
    lblCodigoPro.setFont(new Font("Tahoma", Font.BOLD, 12));
    lblCodigoPro.setBounds(391, 79, 103, 14);
    contentPanel.add(lblCodigoPro);
```

```
lblNombrePro = new JLabel("Nombre Producto :");
lblNombrePro.setForeground(Color.BLACK);
lblNombrePro.setFont(new Font("Tahoma", Font.BOLD, 12));
lblNombrePro.setBounds(360, 121, 134, 14);
contentPanel.add(lblNombrePro);

lblCantidad = new JLabel("Cantidad :");
lblCantidad.setHorizontalAlignment(SwingConstants.CENTER);
lblCantidad.setForeground(Color.BLACK);
lblCantidad.setFont(new Font("Tahoma", Font.BOLD, 12));
lblCantidad.setBounds(598, 139, 153, 14);
contentPanel.add(lblCantidad);

lblPrecioPro = new JLabel("Precio S/ :");
lblPrecioPro.setForeground(Color.BLACK);
lblPrecioPro.setFont(new Font("Tahoma", Font.BOLD, 12));
lblPrecioPro.setBounds(370, 166, 107, 14);
contentPanel.add(lblPrecioPro);

txtNombreProd = new JTextField();
txtNombreProd.setEditable(false);
txtNombreProd.setBounds(499, 119, 103, 20);
contentPanel.add(txtNombreProd);
txtNombreProd.setColumns(10);

txtCantUnid = new JTextField();
txtCantUnid.setBounds(631, 164, 86, 20);
contentPanel.add(txtCantUnid);
txtCantUnid.setColumns(10);

txtPrecioUni = new JTextField();
txtPrecioUni.setEditable(false);
txtPrecioUni.setBounds(499, 164, 103, 20);
contentPanel.add(txtPrecioUni);
txtPrecioUni.setColumns(10);

lblNewLabel_7 = new JLabel("Boletas Electronica :");
lblNewLabel_7.setForeground(Color.BLACK);
lblNewLabel_7.setFont(new Font("Tahoma", Font.BOLD, 14));
lblNewLabel_7.setBounds(20, 200, 153, 14);
contentPanel.add(lblNewLabel_7);

scrollPane = new JScrollPane();
scrollPane.setBounds(10, 243, 741, 425);
contentPanel.add(scrollPane);

txtS = new JTextArea();
txtS.setForeground(new Color(0, 51, 102));
scrollPane.setViewportView(txtS);

btnVenta = new JButton("EFFECTUAR VENTA");
btnVenta.setFont(new Font("Tahoma", Font.BOLD, 14));
btnVenta.setForeground(new Color(0, 51, 102));
scrollPane.setColumnHeaderView(btnVenta);
btnVenta.addActionListener(this);

lblNewLabel_2 = new JLabel("");
lblNewLabel_2.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/usuarios_grande.png")));
;
lblNewLabel_2.setBounds(20, 51, 55, 73);
```

```
contentPanel.add(lblNewLabel_2);

lblNewLabel_8 = new JLabel("");
lblNewLabel_8.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/zapatos_grande.png")));
lblNewLabel_8.setBounds(342, 51, 55, 73);
contentPanel.add(lblNewLabel_8);

lblNewLabel_9 = new JLabel("DATOS DEL CLIENTE");
lblNewLabel_9.setFont(new Font("Tahoma", Font.BOLD, 12));
lblNewLabel_9.setForeground(Color.BLUE);

lblNewLabel_9.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_9.setBounds(20, 39, 228, 14);
contentPanel.add(lblNewLabel_9);

lblNewLabel_10 = new JLabel("DATOS DEL PRODUCTO");
lblNewLabel_10.setFont(new Font("Tahoma", Font.BOLD, 13));
lblNewLabel_10.setForeground(new Color(153, 50, 204));

lblNewLabel_10.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_10.setBounds(332, 40, 270, 14);
contentPanel.add(lblNewLabel_10);

lblApellidoCli = new JLabel("Apellidos :");

lblApellidoCli.setHorizontalAlignment(SwingConstants.CENTER);
lblApellidoCli.setForeground(Color.BLACK);
lblApellidoCli.setFont(new Font("Tahoma", Font.BOLD, 12));
lblApellidoCli.setBounds(20, 160, 128, 14);
contentPanel.add(lblApellidoCli);

txtApellidoCli = new JTextField();
txtApellidoCli.setEditable(false);
txtApellidoCli.setColumns(10);
txtApellidoCli.setBounds(145, 160, 103, 20);
contentPanel.add(txtApellidoCli);

txtCodVenta = new JTextField();
txtCodVenta.setEditable(false);
txtCodVenta.setBounds(100, 11, 73, 20);
contentPanel.add(txtCodVenta);
txtCodVenta.setColumns(10);

lblCodVenta = new JLabel("Nro\u00B0 Venta :");
lblCodVenta.setHorizontalAlignment(SwingConstants.CENTER);
lblCodVenta.setForeground(Color.BLACK);
lblCodVenta.setFont(new Font("Tahoma", Font.BOLD, 12));
lblCodVenta.setBounds(0, 14, 90, 14);
contentPanel.add(lblCodVenta);

lblFecha = new JLabel("");
lblFecha.setHorizontalAlignment(SwingConstants.CENTER);
lblFecha.setForeground(Color.BLACK);
lblFecha.setFont(new Font("Tahoma", Font.BOLD, 12));
lblFecha.setBounds(619, 12, 145, 17);
contentPanel.add(lblFecha);

txtCodigoCli = new JTextField();
txtCodigoCli.setBounds(145, 77, 103, 20);
contentPanel.add(txtCodigoCli);
```

```
txtCodigoCli.setColumns(10);

txtCodigoPro = new JTextField();
txtCodigoPro.setColumns(10);
txtCodigoPro.setBounds(499, 77, 103, 20);
contentPanel.add(txtCodigoPro);
setLocationRelativeTo(null);
setVisible(true);

lblFecha.setText(""+fechaActual());
txtCodVenta.setText(""+av.numeroCorrelativo());

separator = new JSeparator();
separator.setBackground(Color.WHITE);
separator.setForeground(Color.BLACK);
separator.setBounds(625, 121, 103, 90);
contentPanel.add(separator);

separator_1 = new JSeparator();
separator_1.setOrientation(SwingConstants.VERTICAL);
separator_1.setForeground(Color.BLACK);
separator_1.setBackground(Color.WHITE);
separator_1.setBounds(621, 121, 31, 85);
contentPanel.add(separator_1);

separator_2 = new JSeparator();
separator_2.setOrientation(SwingConstants.VERTICAL);
separator_2.setForeground(Color.BLACK);
separator_2.setBackground(Color.WHITE);
separator_2.setBounds(727, 121, 22, 85);
contentPanel.add(separator_2);

separator_3 = new JSeparator();
separator_3.setForeground(Color.BLACK);
separator_3.setBackground(Color.WHITE);
separator_3.setBounds(624, 205, 103, 25);
contentPanel.add(separator_3);

lblCheckCliente = new JLabel("");
lblCheckCliente.addMouseListener(this);
lblCheckCliente.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/check_icon_24.png")));
lblCheckCliente.setBounds(258, 72, 31, 25);
contentPanel.add(lblCheckCliente);

lblCheckProducto = new JLabel("");
lblCheckProducto.addMouseListener(this);
lblCheckProducto.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/check_icon_24.png")));
lblCheckProducto.setBounds(607, 72, 31, 25);
contentPanel.add(lblCheckProducto);

lblBorrarProducto = new JLabel("");
lblBorrarProducto.addMouseListener(this);
lblBorrarProducto.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/error_24.png")));
lblBorrarProducto.setBounds(636, 72, 31, 25);
contentPanel.add(lblBorrarProducto);

lblBorrarCliente = new JLabel("");
lblBorrarCliente.addMouseListener(this);
```

```
lblBorrarCliente.setIcon(new
ImageIcon(DlgVentas.class.getResource("/iconos/error_24.png")));
lblBorrarCliente.setForeground(Color.WHITE);
lblBorrarCliente.setBounds(287, 73, 31, 24);
contentPanel.add(lblBorrarCliente);

btnNuevo = new JButton("NUEVA VENTA");
btnNuevo.setFont(new Font("Tahoma", Font.BOLD, 12));
btnNuevo.setBackground(new Color(0, 204, 0));
btnNuevo.addActionListener(this);
btnNuevo.setBounds(619, 217, 132, 23);
contentPanel.add(btnNuevo);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btnNuevo) {
        do_btnNuevoActionPerformed(e);
    }
    if (e.getSource() == btnVenta) {
        do_btnVentaActionPerformed(e);
    }
}
private JLabel lblApellidoCli;
private JTextField txtApellidoCli;
private JTextField txtCodVenta;
private JLabel lblCodVenta;

//DECLARAR GLOBALES
ArregloVentas av = new ArregloVentas();
ArregloCliente ac = new ArregloCliente();
ArregloProductos ap = new ArregloProductos();

private JLabel lblFecha;
private JTextField txtCodigoCli;
private JTextField txtCodigoPro;
private JSeparator separator;
private JSeparator separator_1;
private JSeparator separator_2;
private JSeparator separator_3;
private JLabel lblCheckCliente;
private JLabel lblCheckProducto;
private JLabel lblBorrarProducto;
private JLabel lblBorrarCliente;
private JButton btnVenta;
private JButton btnNuevo;

protected void do_btnVentaActionPerformed(ActionEvent e)
{
    Venta v1= new
Venta(leerCodigoVenta(),leerProducto(), leerCliente(),
leerCantidad(),fechaActual(),leerPrecio());
    av.adicionar(v1);
    listar(v1);
    limpieza();
}
void listar(Venta x)
{
    imprimir("\t\t\t*****BOLETA DE
PAGO*****");
    imprimir("");
}
```

```

                imprimir("\t\t\tCODIGO DEL CLIENTE\t:");
"+leerCliente());
                imprimir("");
                imprimir("\t\t\tNOMBRE DEL CLIENTE\t:");
"+leerNombreCliente());
                imprimir("");
                imprimir("\t\t\tAPELLIDO DEL CLIENTE\t:");
"+leerApellidoCliente());
                imprimir("");
                imprimir("\t\t\tCODIGO DEL PRODUCTO\t:");
"+leerProducto());
                imprimir("");
                imprimir("\t\t\tNOMBRE DEL PRODUCTO\t:");
"+leerNombreProducto());
                imprimir("");
                imprimir("\t\t\tCANTIDADES ADQUIRIDAS\t:");
"+leerCantidad()+" UNIDADES");
                imprimir("");
                imprimir("\t\t\tPRECIO UNITARIO\t: "+leerPrecio());
                imprimir("");
                imprimir("\t\t\tIMPORTE SUBTOTAL\t: "+x.subTotal());
                imprimir("");
                imprimir("\t\t\tIMPORTE IGV ( 18% )\t:");
"+x.desctIgv());
                imprimir("");
                imprimir("\t\t\tIMPORTE TOTAL A PAGAR\t:");
"+x.pagoTotal());
                imprimir("\n");
}

//Métodos tipo void

void imprimir(String s)
{txtS.append(s + "\n");}

void mensaje(String s)
{JOptionPane.showMessageDialog(this, s);}

void error(String s, JTextField txt) {
    mensaje(s);
    txt.setText("");
    txt.requestFocus();
}

void limpieza() {
    txtCodVenta.setText(""+ av.numeroCorrelativo());
    txtNombreCli.setText("");
    txtApellidoCli.setText("");
    txtCodigoCli.setText("");
    txtCodigoPro.setText("");
    txtNombreProd.setText("");
    txtPrecioUni.setText("");
    txtCodigoCli.requestFocus();
}

//Métodos que retornan valor (Sin Parametros)
int leerCodigoVenta() {return
Integer.parseInt(txtCodVenta.getText().trim());}
int leerCliente() {return
Integer.parseInt(txtCodigoCli.getText().trim());}
String leerNombreCliente() {return txtNombreCli.getText();}
```

```
String leerApellidoCliente() {return txtApellidoCli.getText();}  
int leerProducto() {return  
Integer.parseInt(txtCodigoPro.getText().trim());}  
String leerNombreProducto() {return txtNombreProd.getText();}  
double leerPrecio() {return  
Double.parseDouble(txtPrecioUni.getText().trim());}  
int leerCantidad() { return  
Integer.parseInt(txtCantUnid.getText().trim());}  
  
String fechaActual() {  
    int dd, mm, aa;  
    Calendar c = new GregorianCalendar();  
    dd = c.get(Calendar.DAY_OF_MONTH);  
    mm = c.get(Calendar.MONTH) + 1;  
    aa = c.get(Calendar.YEAR);  
    return Fecha.ajustar(dd) + "/" + Fecha.ajustar(mm) + "/" +  
aa;  
}  
public void mouseClicked(MouseEvent e) {  
    if (e.getSource() == lblCheckCliente) {  
        do_lblCheckCliente_mouseClicked(e);  
    }  
    if (e.getSource() == lblCheckProducto) {  
        do_lblCheckProducto_mouseClicked(e);  
    }  
    if (e.getSource() == lblBorrarCliente) {  
        do_lblBorrarCliente_mouseClicked(e);  
    }  
    if (e.getSource() == lblBorrarProducto) {  
        do_lblBorrarProducto_mouseClicked(e);  
    }  
}  
  
//Evento Mouse Clicked  
public void mouseEntered(MouseEvent e) {  
    lblCheckCliente.setCursor(new Cursor(Cursor.HAND_CURSOR));  
    lblCheckProducto.setCursor(new  
Cursor(Cursor.HAND_CURSOR));  
    lblBorrarProducto.setCursor(new  
Cursor(Cursor.HAND_CURSOR));  
    lblBorrarCliente.setCursor(new  
Cursor(Cursor.HAND_CURSOR));  
}  
public void mouseExited(MouseEvent e) {}  
public void mousePressed(MouseEvent e) {}  
public void mouseReleased(MouseEvent e) {}  
protected void do_lblBorrarProducto_mouseClicked(MouseEvent e) {  
    txtCodigoPro.setText("");  
    txtNombreProd.setText("");  
    txtPrecioUni.setText("");  
    txtCodigoPro.requestFocus();  
}  
protected void do_lblBorrarCliente_mouseClicked(MouseEvent e) {  
    txtCodigoCli.setText("");  
    txtNombreCli.setText("");  
    txtApellidoCli.setText("");  
    txtCodigoCli.requestFocus();  
}
```

```

protected void do_lblCheckProducto_mouseClicked(MouseEvent e) {
    int codigo = Integer.parseInt(txtCodigoPro.getText());
    Producto p = ap.buscar(codigo);
    if (p != null) {
        txtNombreProd.setText(p.getNombre());
        txtPrecioUni.setText(""+p.getPrecio());
    }
    else {
        mensaje("EL CODIGO NO EXISTE");
        limpieza();
    }
}

protected void do_lblCheckCliente_mouseClicked(MouseEvent e) {
    int codigo = Integer.parseInt(txtCodigoCli.getText());
    Cliente c = ac.buscar(codigo);
    if (c != null) {
        txtNombreCli.setText(c.getNombres());
        txtApellidoCli.setText(c.getApellidos());
    }
    else {
        mensaje("EL CODIGO NO EXISTE");
        limpieza();
    }
}

protected void do_btnNuevoActionPerformed(ActionEvent e) {
    limpieza();
    txtS.setText("");
    txtCantUnid.setText("");
}
}

```

**DlgAlmacen.java**

```

package principal;

import java.awt.BorderLayout;
import java.awt.FlowLayout;

import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import arreglos.ArregloCliente;
import arreglos.ArregloProductos;
import arreglos.ArregloVentas;
import clases.Cliente;
import clases.Producto;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import java.awt.Color;
import java.awt.Cursor;

import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import java.awt.Font;

```

```
import javax.swing.ImageIcon;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.awt.Toolkit;

public class DlgAlmacen extends JDialog implements ActionListener,
MouseListener {

    private final JPanel contentPanel = new JPanel();
    private JTextField txtCodProducto;
    private JTextField txtNombreProducto;
    private JTextField txtCantidadIngreso;
    private JTextField txtStockMaximo;
    private JLabel lblCheckCodP;
    private JLabel lblErrorIconP;
    private JButton btnIngresar;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        try {
            DlgAlmacen dialog = new DlgAlmacen();

            dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
            dialog.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Create the dialog.
     */
    public DlgAlmacen() {

        setIconImage(Toolkit.getDefaultToolkit().getImage(DlgAlmacen.class
        .getResource("/iconos/almacen_icon.png")));
        setTitle("Almacen");
        setBounds(100, 100, 643, 260);
        getContentPane().setLayout(new BorderLayout());
        contentPanel.setBackground(new Color(240, 248, 255));
        contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
        getContentPane().add(contentPanel, BorderLayout.CENTER);
        contentPanel.setLayout(null);

        {
            JLabel lblNewLabel = new JLabel("Codigo del
Product:");
            lblNewLabel.setForeground(Color.BLACK);
            lblNewLabel.setFont(new Font("Dialog", Font.BOLD,
12));
            lblNewLabel.setBounds(10, 30, 147, 14);
            contentPanel.add(lblNewLabel);
        }

        {
            JLabel lblNewLabel_1 = new JLabel("Cantidad a
Ingresar:");
            lblNewLabel_1.setForeground(Color.BLACK);
            lblNewLabel_1.setBounds(10, 160, 147, 14);
            contentPanel.add(lblNewLabel_1);
        }

        {
            JButton btnIngresar = new JButton("Ingresar");
            btnIngresar.addActionListener(this);
            btnIngresar.setBounds(10, 290, 147, 14);
            contentPanel.add(btnIngresar);
        }
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnIngresar) {
            String codProducto = txtCodProducto.getText();
            String nombreProducto = txtNombreProducto.getText();
            String cantidadIngreso = txtCantidadIngreso.getText();
            String stockMaximo = txtStockMaximo.getText();
            String checkCodP = lblCheckCodP.getText();
            String errorIconP = lblErrorIconP.getText();

            if (codProducto.isEmpty() || nombreProducto.isEmpty() ||
                cantidadIngreso.isEmpty() || stockMaximo.isEmpty() ||
                checkCodP.isEmpty() || errorIconP.isEmpty()) {
                JOptionPane.showMessageDialog(dialog, "Por favor, complete todos los campos.");
            } else {
                // Implementacion para ingresar almacenes
                // Aqui se manejaran los datos ingresados
                // Y se actualizara el contenido del panel
                // Ejemplo:
                // JOptionPane.showMessageDialog(dialog, "Almacen ingresado exitosamente.");
            }
        }
    }

    public void mouseClicked(MouseEvent e) {
        // Manejo de click en el mouse
    }

    public void mouseEntered(MouseEvent e) {
        // Manejo de entrada del mouse
    }

    public void mouseExited(MouseEvent e) {
        // Manejo de salida del mouse
    }

    public void mousePressed(MouseEvent e) {
        // Manejo de presion del mouse
    }

    public void mouseReleased(MouseEvent e) {
        // Manejo de liberacion del mouse
    }
}
```

```
        lblNewLabel_1.setFont(new Font("Dialog", Font.BOLD,
12));
        lblNewLabel_1.setBounds(10, 138, 138, 14);
        contentPanel.add(lblNewLabel_1);
    }
}
JLabel lblNewLabel_2 = new JLabel("Nombre del
Producto:");
lblNewLabel_2.setForeground(Color.BLACK);
lblNewLabel_2.setFont(new Font("Dialog", Font.BOLD,
12));
lblNewLabel_2.setBounds(10, 78, 147, 14);
contentPanel.add(lblNewLabel_2);
}
{
JLabel lblMaxStock = new JLabel("Stock Maximo:");
lblMaxStock.setFont(new Font("Dialog", Font.BOLD,
12));
lblMaxStock.setForeground(Color.BLACK);
lblMaxStock.setBackground(Color.WHITE);
lblMaxStock.setBounds(344, 78, 99, 14);
contentPanel.add(lblMaxStock);
}
{
txtCodProducto = new JTextField();
txtCodProducto.setFont(new Font("Tahoma", Font.BOLD
| Font.ITALIC, 12));
txtCodProducto.setBounds(170, 27, 86, 20);
contentPanel.add(txtCodProducto);
txtCodProducto.setColumns(10);
}
{
txtNombreProducto = new JTextField();
txtNombreProducto.setFont(new Font("Tahoma",
Font.BOLD | Font.ITALIC, 12));
txtNombreProducto.setBounds(170, 75, 86, 20);
contentPanel.add(txtNombreProducto);
txtNombreProducto.setColumns(10);
}
{
txtCantidadIngreso = new JTextField();
txtCantidadIngreso.setFont(new Font("Tahoma",
Font.BOLD | Font.ITALIC, 12));
txtCantidadIngreso.setBounds(170, 135, 87, 20);
contentPanel.add(txtCantidadIngreso);
txtCantidadIngreso.setColumns(10);
}
txtStockMaximo = new JTextField();
txtStockMaximo.setFont(new Font("Tahoma", Font.BOLD |
Font.ITALIC, 12));
txtStockMaximo.setForeground(new Color(0, 0, 0));
txtStockMaximo.setEditable(false);
txtStockMaximo.setBounds(456, 75, 99, 20);
contentPanel.add(txtStockMaximo);
txtStockMaximo.setColumns(10);

lblCheckCodP = new JLabel("");
lblCheckCodP.addMouseListener(this);
lblCheckCodP.setIcon(new
ImageIcon(DlgAlmacen.class.getResource("/iconos/check_icon_24.png")));

```

```
lblCheckCodP.setBounds(266, 28, 25, 24);
contentPanel.add(lblCheckCodP);

lblErrorIconP = new JLabel("");
lblErrorIconP.addMouseListener(this);
lblErrorIconP.setIcon(new
ImageIcon(DlgAlmacen.class.getResource("/iconos/error_24.png")));
lblErrorIconP.setBounds(301, 28, 33, 24);
contentPanel.add(lblErrorIconP);

btnNuevoIngreso = new JButton("Nuevo Ingreso");
btnNuevoIngreso.setForeground(Color.WHITE);
btnNuevoIngreso.addActionListener(this);
btnNuevoIngreso.setBackground(new Color(153, 50, 204));
btnNuevoIngreso.setBounds(344, 187, 125, 23);
contentPanel.add(btnNuevoIngreso);

txtStockActual = new JTextField();
txtStockActual.setEditable(false);
txtStockActual.setFont(new Font("Tahoma", Font.BOLD |
Font.ITALIC, 12));
txtStockActual.setBounds(456, 30, 99, 20);
contentPanel.add(txtStockActual);
txtStockActual.setColumns(10);

lblStockActual = new JLabel("Stock Actual:");
lblStockActual.setForeground(Color.BLACK);
lblStockActual.setFont(new Font("Dialog", Font.BOLD, 12));
lblStockActual.setBounds(347, 30, 99, 14);
contentPanel.add(lblStockActual);
{

    btnIngresar = new JButton("Ingresar Stock");
    btnIngresar.setBounds(161, 187, 131, 23);
    contentPanel.add(btnIngresar);
    btnIngresar.addActionListener(this);
    btnIngresar.setForeground(Color.WHITE);
    btnIngresar.setBackground(Color.BLUE);
}

txtStockMin = new JTextField();
txtStockMin.setEditable(false);
txtStockMin.setFont(new Font("Tahoma", Font.BOLD |
Font.ITALIC, 12));
txtStockMin.setBounds(456, 136, 99, 20);
contentPanel.add(txtStockMin);
txtStockMin.setColumns(10);

lblStockMin = new JLabel("Stock Minimo:");
lblStockMin.setForeground(Color.BLACK);
lblStockMin.setFont(new Font("Dialog", Font.BOLD, 12));
lblStockMin.setBounds(344, 138, 99, 14);
contentPanel.add(lblStockMin);
setLocationRelativeTo(null);
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btnNuevoIngreso) {
        actionPerformedBtnNuevoIngreso(e);
    }
    if (e.getSource() == btnIngresar) {
        actionPerformedBtnIngresar(e);
    }
}
```

```
        }
    }

    public void mouseClicked(MouseEvent e) {
        if (e.getSource() == lblCheckCodP) {
            mouseClickedLblCheckCodP(e);
        }
        if (e.getSource() == lblErrorIconP) {
            mouseClickedLblErrorIconP(e);
        }
    }

    //Declarar Globales
    ArregloVentas av = new ArregloVentas();
    ArregloCliente ac = new ArregloCliente();
    ArregloProductos ap = new ArregloProductos();
    private JButton btnNuevoIngreso;
    private JTextField txtStockActual;
    private JLabel lblStockActual;
    private JTextField txtStockMin;
    private JLabel lblStockMin;

    //Boton ingresar stock
    protected void actionPerformedBtnIngresar(ActionEvent e) {
        cambio();
    }
    protected void actionPerformedBtnNuevoIngreso(ActionEvent e) {
        limpieza();
    }

    //Metodos mouse
    public void mouseEntered(MouseEvent e) {
        lblCheckCodP.setCursor(new Cursor(Cursor.HAND_CURSOR));
        lblErrorIconP.setCursor(new Cursor(Cursor.HAND_CURSOR));
    }
    public void mouseExited(MouseEvent e) {
    }
    public void mousePressed(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e) {
    }
    protected void mouseClickedLblErrorIconP(MouseEvent e) {
    }
    protected void mouseClickedLblCheckCodP(MouseEvent e) {
        int codigo = Integer.parseInt(txtCodProducto.getText());
        Producto p = ap.buscar(codigo);
        if (p != null) {
            txtNombreProducto.setText(p.getNombre());
            txtStockMaximo.setText(""+p.getStockMax());
            txtStockActual.setText(""+p.getStockActual());
            txtStockMin.setText(""+p.getStockMin());
        }
        else {
            mensaje("EL CODIGO NO EXISTE");
            limpieza();
        }
    }

    void cambio() {
        int codigo = Integer.parseInt(txtCodProducto.getText());
    }
}
```

```
        int Stock
= Integer.parseInt(txtCantidadIngreso.getText());
    Producto p = ap.buscar(codigo);
    if(p.getStockActual() < p.getStockMax()) {
        p.setStockActual(Stock + p.getStockActual());
        mensaje(";El Stock Actual se Actualizo!");
    }
    else {
        mensaje("El Stock Actual No puede Sobre pasar el
Stock Maximo");
    }
    ap.actualizarArchivo();
}

void mensaje(String s)
{JOptionPane.showMessageDialog(this, s);}

void limpieza() {
    txtCodProducto.setText("");
    txtNombreProducto.setText("");
    txtCantidadIngreso.setText("");
    txtStockActual.setText("");
    txtStockMaximo.setText("");
    txtStockMin.setText("");
}

}
```

### DlgReportes.java

```
package principal;

import java.awt.BorderLayout;
import java.awt.FlowLayout;

import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import arreglos.ArregloCliente;
import arreglos.ArregloProductos;
import arreglos.ArregloVentas;
import clases.Cliente;
import clases.Producto;
import clases.Venta;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JComboBox;
import java.awt.Font;
import java.awt.Color;
import java.awt.SystemColor;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import javax.swing.DefaultComboBoxModel;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class DlgReportes extends JDialog implements ActionListener {
```

```
private final JPanel contentPanel = new JPanel();
private JComboBox cboReporte;
private final JButton btnReportar = new JButton("Generar
Reporte");
private final JTextArea txtS = new JTextArea();
private final JScrollPane scrollPane = new JScrollPane();

/**
 * Launch the application.
 */
public static void main(String[] args) {
    try {
        DlgReportes dialog = new DlgReportes();

        dialog.setDefaultCloseOperation(Dialog.DISPOSE_ON_CLOSE);
        dialog.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * Create the dialog.
 */
public DlgReportes() {
    setTitle("Reportes");
    setBounds(100, 100, 769, 456);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setBackground(new Color(240, 248, 255));
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);
    {

        JLabel lblNewLabel = new JLabel("Seleccione
Reporte:");
        lblNewLabel.setForeground(Color.BLACK);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD,
12));
        lblNewLabel.setBounds(10, 29, 125, 14);
        contentPanel.add(lblNewLabel);
    }

    cboReporte = new JComboBox();
    cboReporte.setModel(new DefaultComboBoxModel(new String[]
{ "-Seleccione una opcion-", "Listado general de ventas", "Listado de
productos con stock por debajo del stock mínimo",
"Listado de productos por unidades vendidas
acumuladas", "Listado de productos por importe total
acumulado" }));
    cboReporte.setBounds(146, 26, 419, 22);
    contentPanel.add(cboReporte);
    btnReportar.addActionListener(this);
    btnReportar.setFont(new Font("Tahoma", Font.BOLD, 12));
    btnReportar.setForeground(Color.BLACK);
    btnReportar.setBounds(575, 26, 156, 23);

    contentPanel.add(btnReportar);
    scrollPane.setBounds(10, 70, 733, 336);
```

```
        contentPanel.add(scrollPane);
        scrollPane.setViewportView(txts);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnReportar) {
            actionPerformedBtnReportar(e);
        }
    }

    protected void actionPerformedBtnReportar(ActionEvent e) {
        int eleccion = cboReporte.getSelectedIndex();
        switch (eleccion) {
            case 1: {
                listadoGeneralVentas();
            }
            break;
            case 2: {
                listadodeProductosstock();
            }
            break;
            case 3: {
                listadoUnidadesProductos();
            }
            break;
            case 4: {
                listadoProImporteTotal();
            }
            break;
        default: {
            mensaje("Haga otra elección");
        }
    }
}

// Declarar Globales
ArregloVentas av = new ArregloVentas();
ArregloCliente ac = new ArregloCliente();
ArregloProductos ap = new ArregloProductos();

private void listadoGeneralVentas() {
    limpiza();
    for (int i = 0; i < av.tamanio(); i++) {
        Venta v = av.obtener(i);
        imprimir("Venta: " + (i + 1));
        if (v != null) {
            imprimir("Numero de Venta\t: " +
v.getcodVenta());
            imprimir("Codigo de Cliente\t: " +
v.getcodCliente());
            imprimir("Codigo de Producto\t: " +
v.getcodProducto());
            imprimir("Fecha de Venta\t\t: " +
v.getFechaVenta());
            imprimir("Importe SubTotal\t: " +
v.subTotal());
            imprimir("Importe IGV\t\t: " + v.desctIgv());
            imprimir("Pago Total\t\t: " + v.pagoTotal());
        }
    }
}
```

```
        imprimir("");
    }
}

private void listadodeProductosstock() {
    limpieza();
    for (int i = 0; i < ap.tamanio(); i++) {
        Producto p = ap.obtener(i);
        if (p.getStockActual() < p.getStockMin()) {
            imprimir("Producto: " + (i + 1));
            imprimir("Codigo del Producto: " +
p.getCodProducto());
            imprimir("Nombre del Producto: " +
p.getNombre());
            imprimir("Stock Actual del Producto: " +
p.getStockActual());
            imprimir("Stock Minimo del Producto: " +
p.getStockMin());
            imprimir("");
        }
    }
}

private void listadoUnidadesProductos() {
    limpieza();
    for (int i = 0; i < ap.tamanio(); i++) {
        Venta v = av.obtener(i);
        Producto p = ap.obtener(i);
        if (p != null) {
            imprimir("Producto: " + (i + 1));
            imprimir("Codigo del Producto: " +
p.getCodProducto());
            imprimir("Nombre del Producto: " +
p.getNombre());
            imprimir("Cantidades Acumuladas: " +
av.acumulacion(v));
            imprimir("");
        }
    }
}

private void listadoProImporteTotal() {
    limpieza();
    for (int i = 0; i < ap.tamanio(); i++) {
        Venta v = av.obtener(i);
        Producto p = ap.obtener(i);
        if (p != null) {
            imprimir("Producto: " + (i + 1));
            imprimir("Codigo del producto: " +
p.getCodProducto());
            imprimir("Nombre del producto: " +
p.getNombre());
            imprimir("Cantidad de importe acumulado: " +
av.acumulaiimport(v));
            imprimir("");
        }
    }
}

// Métodos tipo void
```

```
void imprimir(String s) {
    txtS.append(s + "\n");
}

void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s);
}

void error(String s, JTextField txt) {
    mensaje(s);
    txt.setText("");
    txt.requestFocus();
}

void limpieza() {
    txtS.setText("");
}
}
```

### Gui\_Principal.java

```
package principal;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.ImageIcon;
import java.awt.Toolkit;
import javax.swing.JToolBar;
import javax.swing.JButton;
import java.awt.Color;
import java.awt.SystemColor;
import javax.swing.JLabel;
import javax.swing.UIManager;
import java.awt.Font;
import javax.swing.JPopupMenu;

public class Gui_Principal extends JFrame implements ActionListener {
    private JPanel contentPane;
    private JMenuBar menuBar;
    private JMenu mnMantenimiento;
    private JMenu mnVentas;
    private JMenu mnAlmacen;
    private JMenu mnReportes;
    private JMenuItem mntmClientes;
    private JMenuItem mntmProductos;
    private JToolBar toolBar;
    private JLabel lblNewLabel;
    private JButton btnClientes;
    private JButton btnProductos;
    private JMenuItem mntmEjecutar;
    private JMenuItem mntmAgregar;
```

```
private JMenuItem mntmGenerar;
private JMenu mnSistema;
private JMenuItem mntmSalir;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Gui_Principal frame = new
Guia_Principal();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public Gui_Principal() {
    setResizable(false);

    setIconImage(Toolkit.getDefaultToolkit().getImage(Gui_Principal.
class.getResource("/iconos/fondozapato.jpg")));
    setTitle("Sistema de Ventas");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 786, 422);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    toolBar = new JToolBar();
    toolBar.setFloatable(false);
    toolBar.setBackground(new Color(192, 192, 192));
    toolBar.setBounds(0, 30, 770, 54);
    contentPane.add(toolBar);

    btnClientes = new JButton("Clientes");
    btnClientes.addActionListener(this);
    btnClientes.setBackground(Color.LIGHT_GRAY);
    btnClientes.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/usuarios_grande.png
")));
    toolBar.add(btnClientes);

    btnProductos = new JButton("Productos");
    btnProductos.addActionListener(this);
    btnProductos.setBackground(Color.LIGHT_GRAY);
    btnProductos.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/zapatos_grande.png
")));
    toolBar.add(btnProductos);

    menuBar = new JMenuBar();
    menuBar.setBounds(0, 0, 770, 30);
```

```
contentPane.add(menuBar);

mnMantenimiento = new JMenu("Mantenimiento");
mnMantenimiento.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/engranaje.png")));
menuBar.add(mnMantenimiento);

mntmClientes = new JMenuItem("Clientes");
mntmClientes.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/usuarios.png")));
mntmClientes.addActionListener(this);
mnMantenimiento.add(mntmClientes);

mntmProductos = new JMenuItem("Productos");
mntmProductos.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/zapatos.png")));
mntmProductos.addActionListener(this);
mnMantenimiento.add(mntmProductos);

mnVentas = new JMenu("Ventas");
mnVentas.addActionListener(this);
mnVentas.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/venta_icon.png")));
menuBar.add(mnVentas);

mntmEjecutar = new JMenuItem("Efectuar Venta");
mntmEjecutar.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/Efectuar_icon_24px.
png")));
mntmEjecutar.addActionListener(this);
mnVentas.add(mntmEjecutar);

mnAlmacen = new JMenu("Almacen");
mnAlmacen.addActionListener(this);
mnAlmacen.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/almacen_icon.png")));
);
menuBar.add(mnAlmacen);

mntmAgregar = new JMenuItem("Agregar");
mntmAgregar.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/Agregar_icon_24px.p
ng")));
mntmAgregar.addActionListener(this);
mnAlmacen.add(mntmAgregar);

mnReportes = new JMenu("Reportes");
mnReportes.addActionListener(this);
mnReportes.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/diapositiva.png")));
;
menuBar.add(mnReportes);

mntmGenerar = new JMenuItem("Generar Reporte");
mntmGenerar.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/GenerarRep_icon_24p
x.png")));
mntmGenerar.addActionListener(this);
mnReportes.add(mntmGenerar);

mnSistema = new JMenu("Sistema");
```

```
mnSistema.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/herramientas.png")));
);
menuBar.add(mnSistema);

mntmSalir = new JMenuItem("Salir");
mntmSalir.addActionListener(this);
mntmSalir.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/salir_icon.png")));
mnSistema.add(mntmSalir);

lblNewLabel = new JLabel("");
lblNewLabel.setIcon(new
ImageIcon(Gui_Principal.class.getResource("/iconos/fondozapato.jpg")));
;
lblNewLabel.setBounds(0, 72, 770, 354);
contentPane.add(lblNewLabel);
setLocationRelativeTo(null);
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == mntmSalir) {
        actionPerformedMntmSalir(e);
    }
    if (e.getSource() == mntmAgregar) {
        actionPerformedMntmAgregar(e);
    }
    if (e.getSource() == mntmEjecutar) {
        actionPerformedMntmEjecutar(e);
    }
    if (e.getSource() == mntmGenerar) {
        actionPerformedMntmGenerar(e);
    }
    if (e.getSource() == btnClientes) {
        actionPerformedBtnClientes(e);
    }
    if (e.getSource() == btnProductos) {
        actionPerformedBtnProductos(e);
    }

    if (e.getSource() == mntmProductos) {
        actionPerformedMntmProductos(e);
    }
    if (e.getSource() == mntmClientes) {
        actionPerformedMntmClientes(e);
    }
}

protected void actionPerformedMntmClientes(ActionEvent e) {
    DlgCliente dc = new DlgCliente();
    dc.setLocationRelativeTo(this);
    dc.setVisible(true);
}
protected void actionPerformedMntmProductos(ActionEvent e) {
    DlgProducto dp = new DlgProducto();
    dp.setLocationRelativeTo(this);
    dp.setVisible(true);
}
protected void actionPerformedBtnProductos(ActionEvent e) {
    DlgProducto bp = new DlgProducto();
    bp.setLocationRelativeTo(this);
```

```
        bp.setVisible(true);
    }
protected void actionPerformedBtnClientes(ActionEvent e) {
    DlgCliente bc = new DlgCliente();
    bc.setLocationRelativeTo(this);
    bc.setVisible(true);
}
protected void actionPerformedMntmEjecutar(ActionEvent e) {
    DlgVentas dv = new DlgVentas();
    dv.setLocationRelativeTo(this);
    dv.setVisible(true);
}
protected void actionPerformedMntmAgregar(ActionEvent e) {
    DlgAlmacen da = new DlgAlmacen();
    da.setLocationRelativeTo(this);
    da.setVisible(true);
}
protected void actionPerformedMntmGenerar(ActionEvent e) {
    DlgReportes dr = new DlgReportes();
    dr.setLocationRelativeTo(this);
    dr.setVisible(true);
}
protected void actionPerformedMntmSalir(ActionEvent e) {
    System.exit(0);
}
}
```

## Conclusiones

El sistema ZAPATEX lo hemos creado como una simulación de lo que viene hacer una tienda de venta de zapatos a un rango menor en modelos distintos haciendo parecer lo más real posible y poder adicionar nuevos, dando la satisfacción al vendedor y al cliente; en el caso de vendedor quedará satisfecho por el control de zapatos al momento de vender y al cliente con la vista completa de la cantidad de zapatos y comprobante de pago.

Concluimos en este proyecto la venta cordial de los modelos de zapatos para una tienda en nuestro caso la que llamamos ZAPATEX realizando una venta limpia sin errores y llevando a lo más real posible con las cosas básicas que hemos aprendido en el curso.

## Recomendaciones

Las recomendaciones que damos para entender este proyecto es saber los temas básicos de java que viene a ser fundamental en este sistema creado por el grupo.

Una de las formas de realizar este proyecto es primeramente realizar las funciones requeridas por el proyecto que viene a ser los objetivos principales luego de haber realizado estos objetivos podemos llevarlo a la vida real y pensar en todos los errores que podrían existir y tratar de solucionar cada caso y por último dar una decoración bonita al sistema para que tenga una presentación agraciada siguiendo estos pasos cualquier sistema podrá ser muy útil.

## Glosario

- ZAPATEX: Nombre de la tienda que vende zapatos
- CLIENTE: Persona que compra en la tienda de zapatos
- PRODUCTO: Es el material de la zapato con su respectivo tipo

## Bibliografía

- Clases Cibertec
- Java y Eclipse y desarrolle una aplicación con Java y Eclipse de Frédéric DÉLÉCHAMP
- YouTube en pildorasinformaticas

## Anexos

