UNIVERSIDAD DE LA SIERRA SUR

LICENCIATURA EN INFORMÁTICA

MATERIA: INGENIERÍA DE SOFTWARE

PROFESOR - INVESTIGADOR: ROLANDO PEDRO GABRIEL

INTEGRANTES DEL EQUIPO:

ALMARAZ VARGAS HERMILO

GONZÁLEZ RUIZ FLORICELI

LAGUNAS GONZÁLEZ STEPHANIE JACQUELINE

Índice general

1. Introducción **7**
1.1. Personal involucrado 8

2. Especificación de requisitos **10**
2.1. Alcance (Ámbito del proyecto) 10
2.2. Viabilidad y factibilidad 11
2.3. Requisitos 11
2.3.1. Requisitos funcionales 12
2.3.2. Requisitos no funcionales 13
2.3.3. Mapa de navegación 14

3. Análisis **15**
3.1. Introducción 15
3.2. Diagrama de estructura de UML 15
3.2.1. Diagrama de Flujo de datos 15
3.2.2. Diagrama de clases 21
3.2.3. Diagrama de Objetos 30
3.2.4. Diagrama de Componentes 32
3.2.5. Diagrama de Despliegue 34
3.3. Diagramas de comportamiento 36
3.3.1. Diagramas de casos de uso 36
3.3.2. Diagrama de Estado 57
3.3.3. Diagrama de Secuencia 72
3.3.4. Diagrama de Colaboración 90
3.3.5. Diagrama de Actividades 92

4. Diseño **112**
4.1. Introducción 112
4.2. Descripción del sistema 113
4.3. Arquitectura del sistema 113
4.3.1. Arquitectura física 113
4.3.2. Arquitectura lógica 119
4.4. Estándar de codificación 120
4.5. Diseño de datos 121
4.5.1. Diagrama conceptual 121
4.5.2. Diagrama Entidad-Relación 122
4.5.3. Diagrama de Modelo Relacional 122
4.5.4. Diagrama físico 123

4.5.5. Diccionario de datos 124

5. Implementación **126**
5.1. Intruducción 126
5.2. Código fuente 126

6. Conclusión **247**
6.1. Referencias 247
6.2. Anexo 1.PSP 249
6.3. Anexo 2.Galería de imagenes 250

Índice de figuras

1.1. Colaboradores . . . 8
1.2. Colaboradores . . . 9

2.1. Tabla de requisitos funcionales . . . 12
2.2. Tabla de requisitos funcionales . . . 12
2.3. Tabla de requisitos no funcionales . . . 13
2.4. Tabla de requisitos no funcionales . . . 13
2.5. Mapa de navegación. . . . 14

3.1. Diagrama de Flujo de datos Administrar usuario. . . . 15
3.2. Diagrama de Flujo de datos Cobrar. . . . 16
3.3. Diagrama de Flujo de datos Iniciar sesión. . . . 16
3.4. Diagrama de Flujo de datos Inicio. . . . 17
3.5. Diagrama de Flujo de datos Inventario . . . 17
3.6. Diagrama de Flujo de datos Modificar usuario. . . . 18
3.7. Diagrama de Flujo de datos Modificar producto. . . . 18
3.8. Diagrama de Flujo de datos Registrarse. . . . 19
3.9. Diagrama de Flujo de datos Reporte. . . . 19
3.10. Diagrama de Flujo de datos Tabla administrador. . . . 20
3.11. Diagrama de Flujo de datos Tabla usuario. . . . 20
3.12. Diagrama de Clases . . . 21
3.13. Diagrama de Clases . . . 22
3.14. Diagrama de Clases . . . 23
3.15. Diagrama de Clases . . . 24
3.16. Diagrama de Objetos. . . . 30
3.17. Diagrama de Componentes . . . 32
3.18. Diagrama de Despliegue de SFH. . . . 34
3.19. Diagrama de casos de uso de inicio . . . 36
3.20. Documentación de diagrama de casos de uso de inicio . . . 37
3.21. Diagrama de casos de uso de Registrase . . . 38
3.22. Documentación de diagrama de casos de uso de Registrase . . . 39
3.23. Diagrama de casos de uso de inicio de sesión . . . 41
3.24. Documentación de diagrama de casos de uso de inicio de sesión . . . 42
3.25. Diagrama de casos de uso de tabla administrador . . . 43
3.26. Documentación de diagrama de casos de uso de tabla administrador . . . 44
3.27. Diagrama de casos de uso de reporte diario . . . 46
3.28. Documentación de diagrama de casos de uso de reporte diario . . . 47
3.29. Diagrama de casos de uso de cobrar . . . 48
3.30. Documentación de diagrama de casos de uso de cobrar . . . 49

3.31. Diagrama de casos de uso de agregar producto . . . 51
3.32. Documentación de diagrama de casos de agregar producto . . . 52
3.33. Diagrama de casos de uso de modificar producto . . . 54
3.34. Documentación de diagrama de casos de modificar producto . . . 55
3.35. Diagrama de Estado Administrador . . . 57
3.36. Diagrama de Estado Cobrar . . . 59
3.37. Diagrama de Estado Iniciar sesión . . . 60
3.38. Diagrama de Estado Inicio . . . 61
3.39. Diagrama de Estado Inventario . . . 62
3.40. Diagrama de Estado Modificar producto . . . 64
3.41. Diagrama de Estado Modificar usuario . . . 66
3.42. Diagrama de Estado Registrarse. . . . 67
3.43. Diagrama de Estado Reporte diario. . . . 68
3.44. Diagrama de Estado Tabla administrador. . . . 69
3.45. Diagrama de Estado Tabla usuario. . . . 70
3.46. Diagrama de Secuencia Administrar usuario . . . 72
3.47. Diagrama de Secuencia Cobrar . . . 74
3.48. Diagrama de Secuencia Inicio de sesión . . . 75
3.49. Diagrama de Secuencia Inicio . . . 76
3.50. Diagrama de Secuencia Inventario . . . 77
3.51. Diagrama de Secuencia Modificar producto . . . 80
3.52. Diagrama de Secuencia Modificar usuario . . . 82
3.53. Diagrama de Secuencia Registrarse . . . 84
3.54. Diagrama de Secuencia Reporte . . . 86
3.55. Diagrama de Secuencia Tabla administrador. . . . 88
3.56. Diagrama de Colaboración Administrador. . . . 90
3.57. Diagrama de Colaboración Usuario. . . . 91
3.58. Diagrama de Actividades Administrar usuario . . . 92
3.59. Diagrama de Actividades Agregar producto . . . 94
3.60. Diagrama de Actividades cobrar . . . 96
3.61. Diagrama de Actividades Inicio sesión . . . 98
3.62. Diagrama de actividades de iniciar sesion . . . 99
3.63. Diagrama de Actividades Inventario . . . 100
3.64. Diagrama de Actividades Modoficar producto . . . 102
3.65. Diagrama de actividades de modificar usuario . . . 104
3.66. Diagrama de Actividades Registro . . . 106
3.67. Diagrama de Actividades Reporte . . . 107
3.68. Diagrama de Actividades Tabla administrador . . . 108
3.69. Diagrama de Actividades Tabla usuario . . . 110

4.1. Interface de inicio . . . 113
4.2. Interface de ingreso al usuario . . . 114
4.3. Interface de registro de ingreso del usuario . . . 114
4.4. Interface de ventas del usuario . . . 115
4.5. Interface de ventas del administrador . . . 115
4.6. Interface de inventario del administrador . . . 116
4.7. Interface de modificación de productos . . . 116
4.8. Interface de agregación de productos . . . 117
4.9. Interface de cobro . . . 117
4.10. Interface de reporte . . . 118

4.11. Interface de modificar usuario 118
4.12. Interface de lista de usuarios 119
4.13. Estándar de codificación 120
4.14. Diagrama Entidad-Relación del proyecto de telélefonos celulares. 122
4.15. Diagrma de modelo Relacional del proyecto de telélefonos celulares 122
4.16. Diagrama físico de la creación de base de datos SFH. 123
4.17. Tabla Teléfonos 124
4.18. Tabla Usuarios 124
4.19. Tabla Reporte 125

6.1. Gráfica del usu de PSP 249
6.2. Evidencia la aportación de Hermilo 250
6.3. Evidencia la aportación de Hermilo 251
6.4. Evidencia la aportación de Floriceli 252
6.5. Evidencia la aportación de Floriceli 253
6.6. Evidencia la aportación de Stephanie 254
6.7. Evidencia la aportación de Stephanie 255
6.8. Evidencia de la colaboración del equipo 256

Capítulo 1

Introducción

En la actualidad existen muchas tiendas que realizan venta de teléfonos, pero no todas cuentan con un buen sistema de control de ventas, esto conlleva a que no tengan un buen control de sus productos y que tengan perdidas a su paso, ya que si no se tiene un inventario de ellos puede que haya perdidas que no se vean.

Gracias a estos tipos de sistemas ha habido un gran **avance** y ha ocasionado que en estos tiempos haya un mejor manejo en el control de ventas en las tiendas, esto ayuda a que no necesita estar contratando contadores para que lleven el control, ya que este sistema automáticamente genera un reporte de las ventas en fechas determinadas, lo cual genera un gasto menos para el propietario, este tipo de sistema será multiusuario, esto quiere decir que podrán acceder varias personas en diferentes interfaces, uno será para el administrador y otro para los usuarios de ventas, para el administrador habrá más funcionalidades que lo ayudaran a controlar y organizar los datos, en cambio el usuario solo podrá realizar ventas en el sistema, por eso se realizará un sistema de control de ventas que ayudará a los usuarios a realizar las tareas más rápido y eficaz, así también este sistema contendrá varias funciones que serán de ayuda para el administrador y el usuario, contendrá un apartado donde mostrará un manual de usuario que ayudará a comprender y utilizar mejor el sistema, de esta manera ayudará a realizar las ventas en poco tiempo.

1.1. Personal involucrado

	Nombre	Roles	Descripción
	Lirio Ruiz Guerra	DBA	Grado Académico • Maestría en Ciencias de la Computación • Ingeniería en Sistemas Computacionales Líneas de investigación Actuales • Tecnología educativa • Procesos de software
	Rolando Pedro Gabriel	Usuario final	Grado Académico • Maestría en Tecnologías de Cómputo Aplicado • Licenciatura en informática Línea de Investigación actual • Tecnologías de la información y comunicación
	Everardo de Jesus Pacheco Antonio	Súper Administrador	Grado Académico Maestría en Tecnología Educativa • Licenciatura en Informática Líneas de Investigación Actuales • Tecnología Educativa • Tecnologías de Información y Comunicación

Figura 1.1: Colaboradores

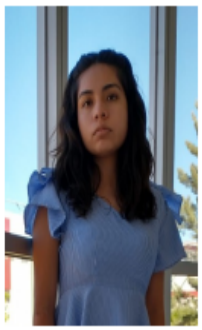
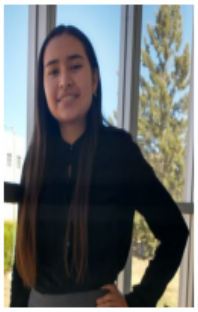
	Stephanie Jacqueline Lagunas González	Jefe	Grado Académico • Estudiante de la Licenciatura en Informática
	Floriceli González Ruiz	Administrador	Grado Académico • Estudiante de la Licenciatura en Informática
	Hermilo Almaraz Vargas	Desarrollador	Grado Académico • Estudiante de la Licenciatura en Informática

Figura 1.2: Colaboradores

Capítulo 2

Especificación de requisitos

2.1. Alcance (Ámbito del proyecto)

Este sistema de ventas de teléfonos, ayuda al usuario a realizar los registros del los equipos existente, para llevar un control adecuado diario de los equipos vendidos, ya que actualmente las tiendas no cuentan con un sistema adecuado para el registro de sus productos, ya que base de estos se ha tenido caso de perdidas de información, así como perdida de ganancia al llevar un mal registro de sus equipos vendidos, esto se puede evitar contando con un adecuado sistema de venta que garantice la seguridad de su información. Este sistema permitirá llevar un mejor control de sus registros de ventas, ganancias, y perdidas. Esto le permitirá al usuario detectar los posibles problemas que tendrá en un futuro. Evitando de esta manera perdidas monetarias y descontrol de su inventario, este sistema de venta estará diseñado de manera que el usuario pueda interactuar fácilmente y obtenga buenos resultados al utilizarlo. El sistema contará con diversas funcionalidades:

- Validación y registro de usuario.

  Se implantará la validación de usuario con la finalidad de mantener seguro los datos del usuario, evitando de esta manera que otras personas no autorizadas puedan acceder al sistema haciendo un mal uso del sistema de ventas.

- Registros de los datos de los dispositivos.

  Él podrá realizar sus registros diarios de las ventas y adquisición de dispositivos nuevos, además podrá registras las características de cada dispositivo como marca, modelo, precio y color del producto.

- Modificación

  En cualquier momento el usuario podrá acceder a sus registros para poder hacer las modificaciones necesarias siempre y cuando se requiera.

- Eliminar registros

- Búsqueda

Este sistema tendrá el objetivo de ser una gran herramienta que pueda ayudar al usuario llevar una mejor organización de sus operaciones en su negocio y también está claro que esta también debe

cumplir las expectativas de cliente en cuanto al diseño del software, ya que el diseño se efectuará de forma que pueda ser más agradable y entendible para el usuario, este sistema contara con diferentes herramientas, como el registro de ganancias en donde podrá ver si su negocio está resultando así como también contara con página en donde puedan consultar los modelos de teléfono que ha vendido y así el usuario podrá ver que compras son más recomendables les hará a sus proveedores, también se tomara en cuenta que se harán las necesidades que debe cubrir el software, este software también se hará un registro del usuario que ocupara el software y se podrán más usuarios si fuera necesario.

2.2. Viabilidad y factibilidad

Para la realización de este sistema primero tenemos que identificar los pasos que nos llevará a realizar un buen desarrollo del proyecto, debido a que se deben de ir identificando los problemas futuros que surgirán en el transcurso de la realización del proyecto. En este estudio tendremos que invertir mucho tiempo para hacer pruebas y determinar si puede llegar a realizar de una manera y buscar diferentes soluciones, cosas que se ajusten a nuestro presupuesto sin tener dificultades en el futuro. Una cosa que hemos mencionado antes, es que nuestro sistema tendrá implementado el ingreso al usuario, en cuestión técnica se utilizará una función para comparar los datos del usuario con los que está ingresando. El control de ventas es otra herramienta que estará incluida en nuestro sistema, técnicamente solo se encargará de llevar el control de los productos que entran y salen del almacén para poder llevar un registro de cada una de ellas, también contará con una sección que nos permitirá la actualización, en esta solo se tendrá que hacer una investigación previa de la información que se le agregará de cada uno de los dispositivos, contará con una sección de consulta(búsqueda), en esta sección nos permitirá observar las características específicas de un dispositivo. Recabando todas las secciones mencionadas anteriormente se llegó a la conclusión de que se tendrá que invertir la mayor parte del tiempo en la codificación y el diseño de nuestra interfaz, en cuestión económica no se tendrá que invertir demasiado, debido a que únicamente se estará utilizando internet y equipo de cómputo, mencionamos en este último punto porque el equipo que estaremos utilizando sufrirá desgaste y disminuirá su costo, las secciones que incluiremos no generaran costo. De acuerdo con el análisis mencionado en los párrafos anteriores podemos ver que nuestro sistema si lo podemos llevar a cabo ya que contamos con el presupuesto y las herramientas necesarias para desarrollarlo, así como el conocimiento necesario para elaborarlo sin ningún contratiempo. Nos llevará mucho tiempo en realizarlo, pero organizaremos nuestros tiempos para avanzar de una manera segura y viable, realizaremos búsquedas de información de los dispositivos móviles que tienen más demanda en el mercado para poder tener la información que se va a requerir para guardarla en nuestra base de datos.

2.3. Requisitos

2.3.1. Requisitos funcionales

Número de requisito:	RF1
Nombre del requisito:	Autentificación de usuario
Descripción:	Esta función verificará la autenticidad de los usuarios al introducir su información y determinar el tipo de usuario.
Fuente de requisito:	Tabla: Nombre y Contraseña
Prioridad de requisito:	Esencial

Número de requisito:	RF2
Nombre del requisito:	Modificar datos
Descripción:	Esta función permitirá modificar cualquier registro de la base de datos.
Fuente de requisito:	Tabla: Administrador
Prioridad de requisito:	Alta

Número de requisito:	RF3
Nombre del requisito:	Inventario diario
Descripción:	En esta función se podrá determinar el inventario diario de los productos.
Fuente de requisito:	Tabla: Producto y cantidad
Prioridad de requisito:	Alta

Figura 2.1: Tabla de requisitos funcionales

Número de requisito:	RF4
Nombre del requisito:	Insertar datos
Descripción:	Esta función permitirá insertar cualquier registro de la base de datos.
Fuente de requisito:	Tabla: Administrador
Prioridad de requisito:	Alta

Número de requisito:	RF5
Nombre del requisito:	Eliminar datos
Descripción:	Esta función permitirá eliminar cualquier registro de la base de datos.
Fuente de requisito:	Tabla: Administrador
Prioridad de requisito:	Alta

Número de requisito:	RF6
Nombre del requisito:	Cobrar ventas
Descripción:	Esta función nos permitirá cobrar los productos vendidos a los clientes.
Fuente de requisito:	Tabla: Administrador usuario
Prioridad de requisito:	Alta

Figura 2.2: Tabla de requisitos funcionales

2.3.2. Requisitos no funcionales

Número de requisito:	RNF1
Nombre:	Confidencialidad
Descripción:	La información otorgada por los usuarios se manipulará únicamente con fines corporativos y de manera limpia.
Prioridad:	Alta

Número de requisito:	RNF2
Nombre:	Robustez
Descripción:	El software debe ser capaz de manejar toda la información recolectada a través del tiempo con fluidez.
Prioridad:	Media

Número de requisito:	RNF3
Nombre:	Restricción de Contenido
Descripción:	El acceso a cada página del aplicativo está determinado por el rol del usuario.
Prioridad:	Alta

Figura 2.3: Tabla de requisitos no funcionales

Número de requisito:	RNF4
Nombre:	Usabilidad
Descripción:	Este sistema debe ser fácil de usar y debe tener buena accesibilidad para el usuario y el administrador.
Prioridad:	Media

Número de requisito:	RNF5
Nombre:	Seguridad
Descripción:	El ingreso al sistema estará restringido bajo contraseñas cifradas y usuarios definidos.
Prioridad:	Alta

Figura 2.4: Tabla de requisitos no funcionales

2.3.3. Mapa de navegación

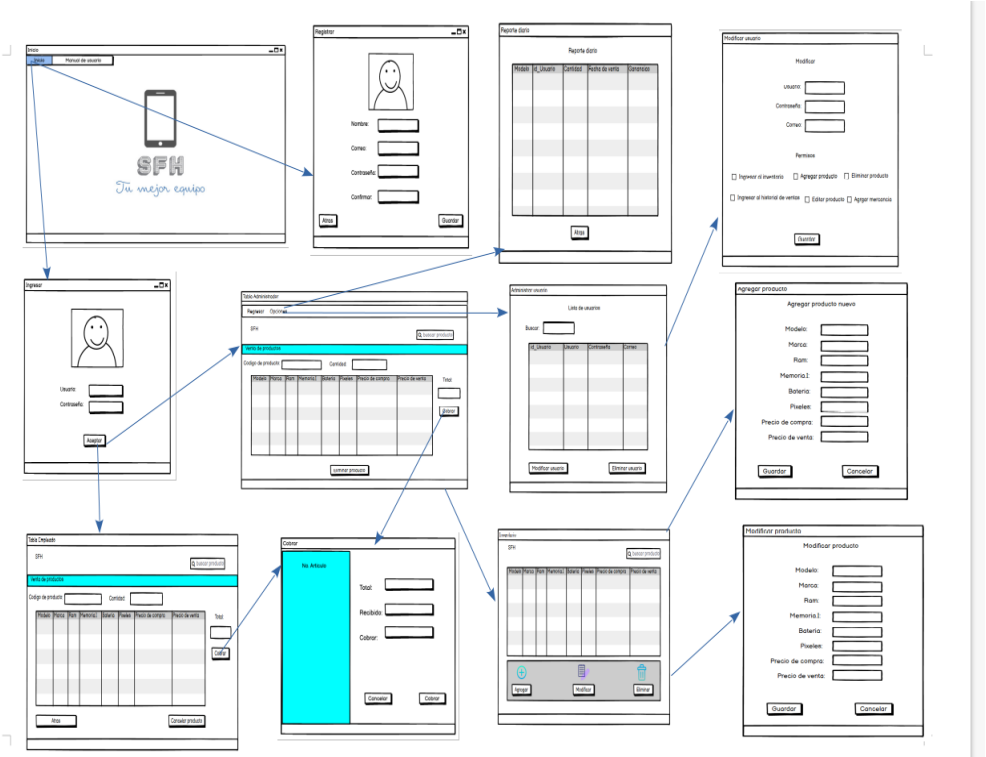


Figura 2.5: Mapa de navegación.

Capítulo 3

Análisis

3.1. Introducción

En el análisis de requisitos se detalla aquellos componentes del sistema que vamos a implementar mediante software. El objetivo es comprender cuales son los datos que se van a manejar, cual va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cual es el rendimiento que se espera lograr. En este capítulo se presentan los distintos diagramas que fueron implementados mediante la utilización de la herramienta PlanUML. Se especifica el flujo de ejecución que tendrá el sistema mediante la representación gráfica que fueron implementado con la ayuda de la herramienta.

3.2. Diagrama de estructura de UML

3.2.1. Diagrama de Flujo de datos

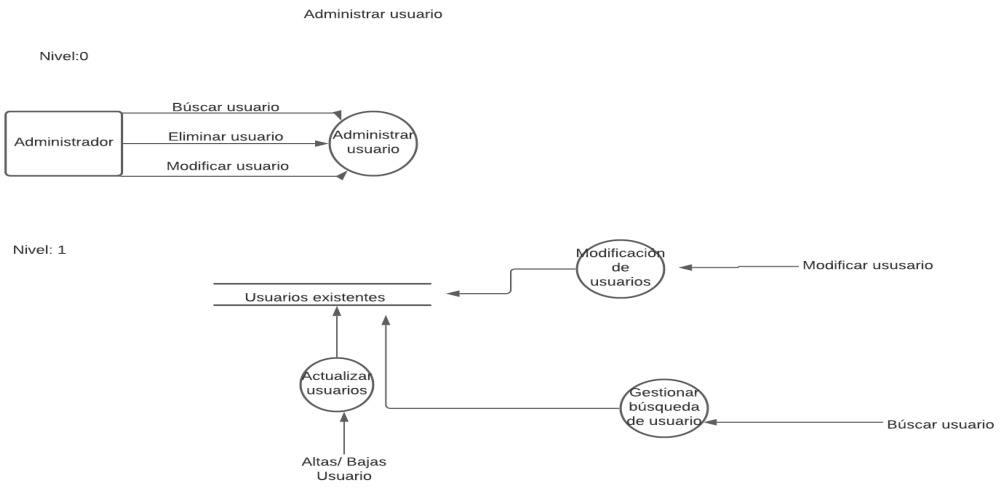


Figura 3.1: Diagrama de Flujo de datos Administrar usuario.

Se especifica de manera grafica la los pasos que debe seguir el administrador para poder administrar a los usuarios.

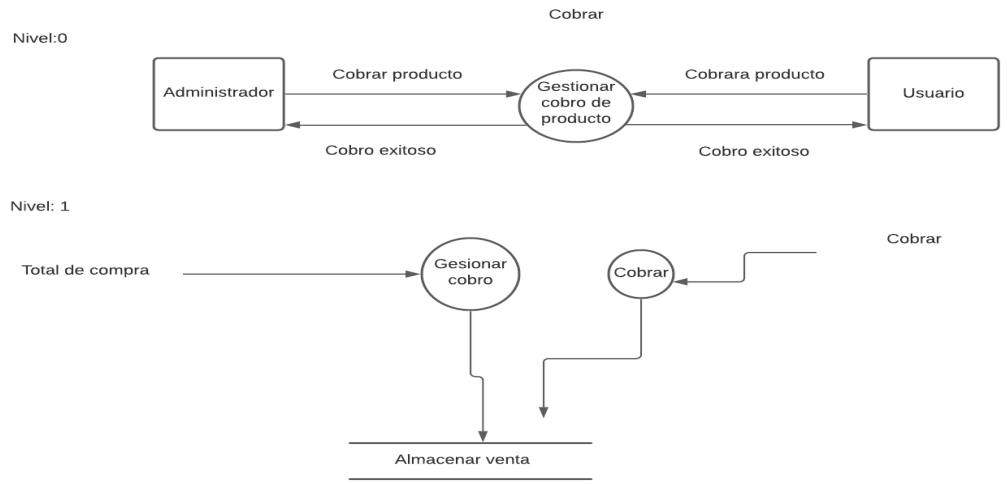


Figura 3.2: Diagrama de Flujo de datos Cobrar.

Se detalla de manera general el flujo que debe seguir el administrador, así como el usuario para poder realizar un cobro de un producto móvil.

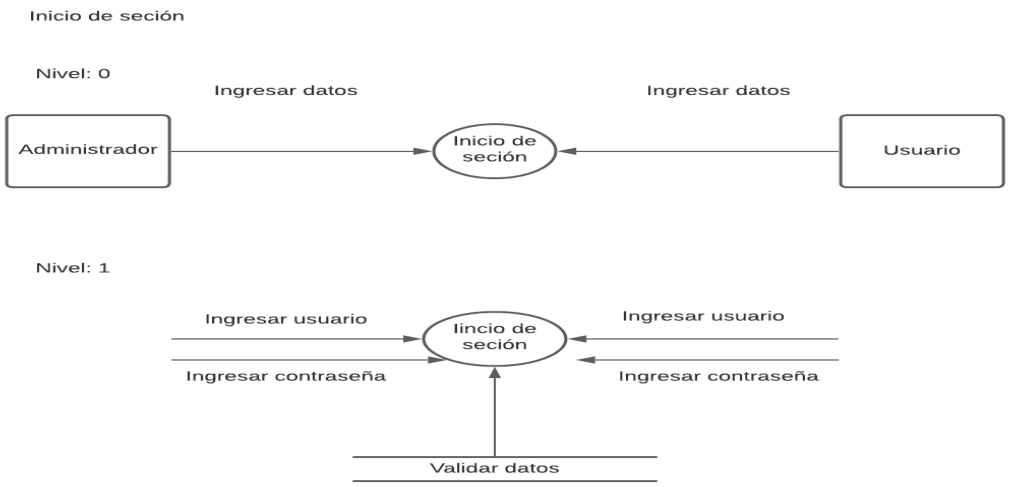


Figura 3.3: Diagrama de Flujo de datos Iniciar sesión.

En el diagrama de Inicio de sesión se puede observar el flujo que debe seguir el usuario como el administrador para poder iniciar sesión.

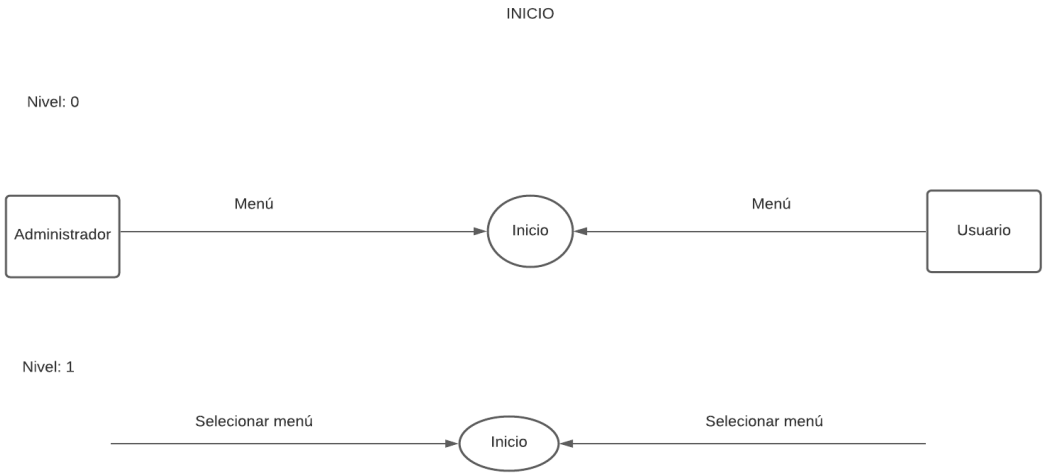


Figura 3.4: Diagrama de Flujo de datos Inicio.

Administrado o usuario selecciona una de las opciones de la venta inicio.

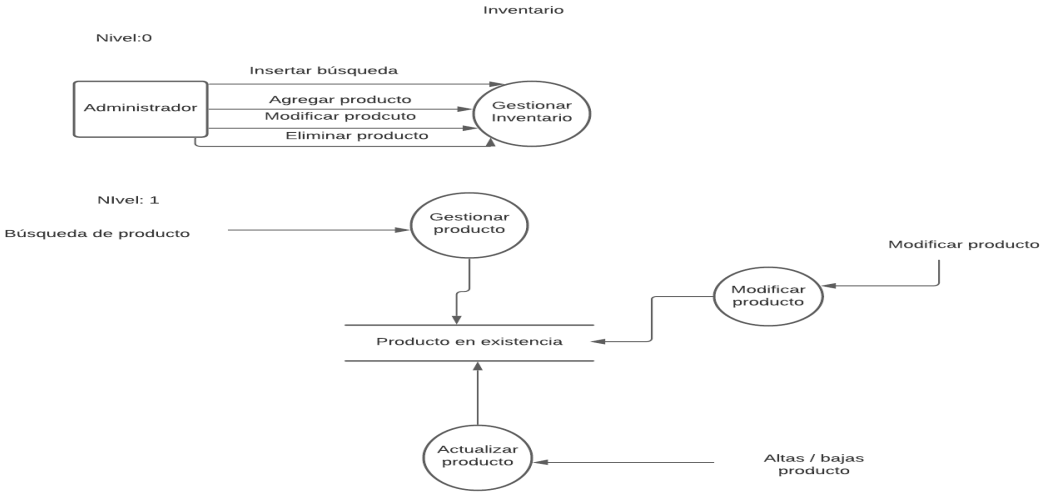


Figura 3.5: Diagrama de Flujo de datos Inventario

La grafica inventario detalla de manera general las acciones que el administrador puede controlar y como es que debe seguir el flujo de navegación.

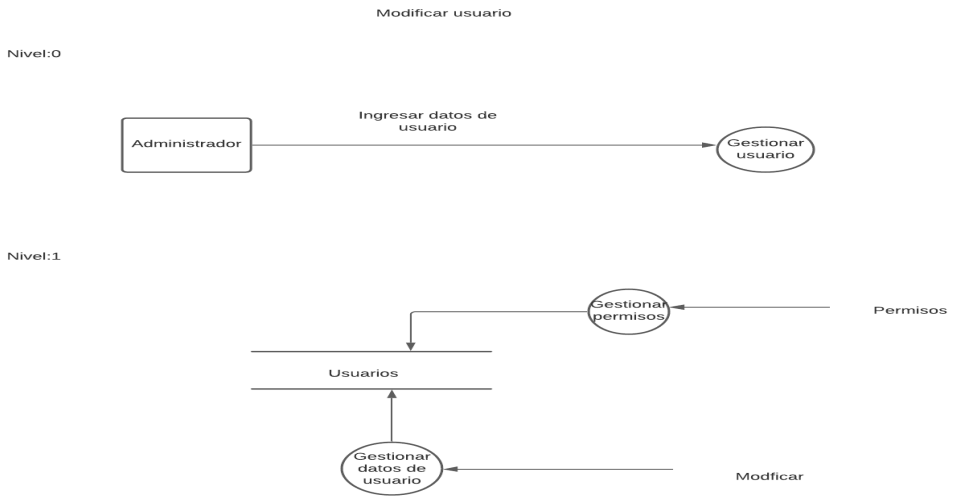


Figura 3.6: Diagrama de Flujo de datos Modificar usuario.

El diagrama inventario especifica el flujo que debe seguir el administrador para poder modificar un usuario.

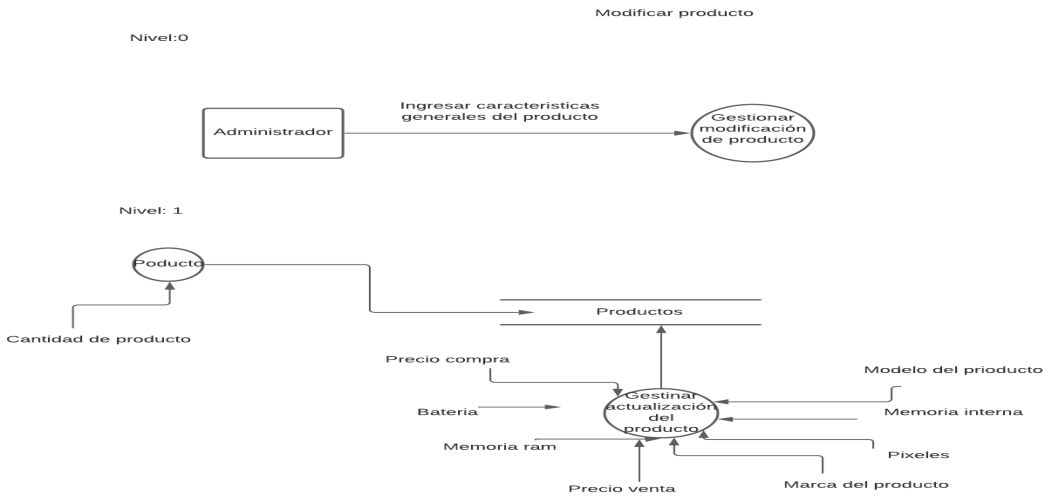


Figura 3.7: Diagrama de Flujo de datos Modificar producto.

El flujo o la forma en que el administrador o usuario deben seguir para poder modificar un producto se especifica en este diagrama de manera general.

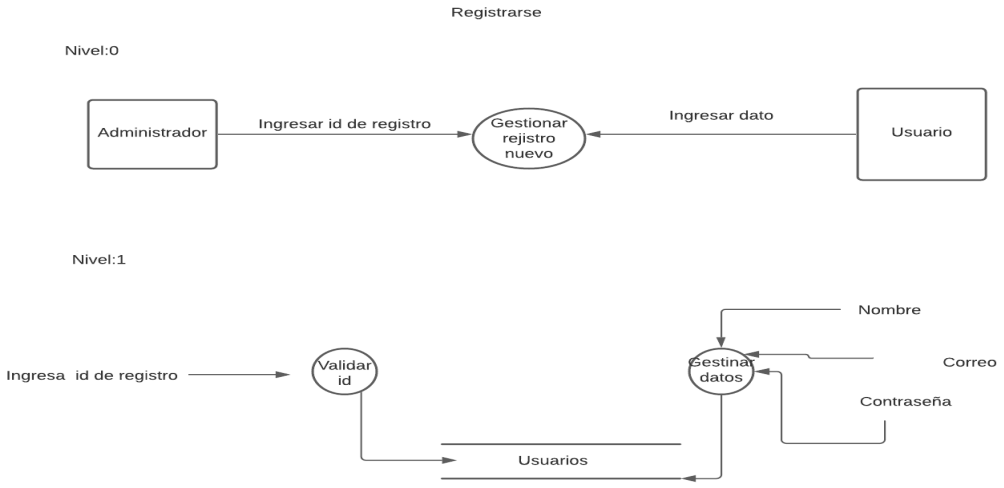


Figura 3.8: Diagrama de Flujo de datos Registrarse.

Se detalla de manera grafica la forma en la se debe registrar un usuario.

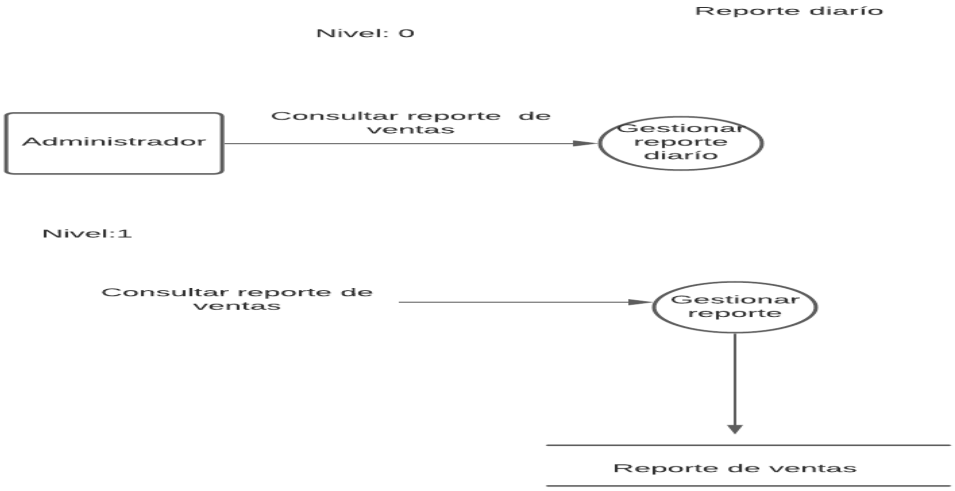


Figura 3.9: Diagrama de Flujo de datos Reporte.

Se detalla el flujo a seguir para poder gestionar un reporte de ventas de productos.

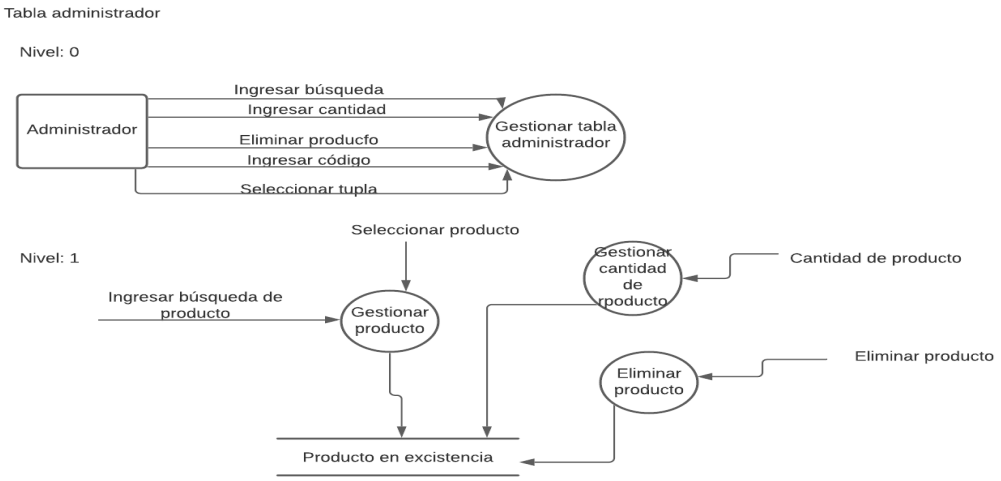


Figura 3.10: Diagrama de Flujo de datos Tabla administrador.

En este diagrama se puede observar de manera general como es que el administrador gestiona la tabla administradora para realizar una venta de un producto móvil.

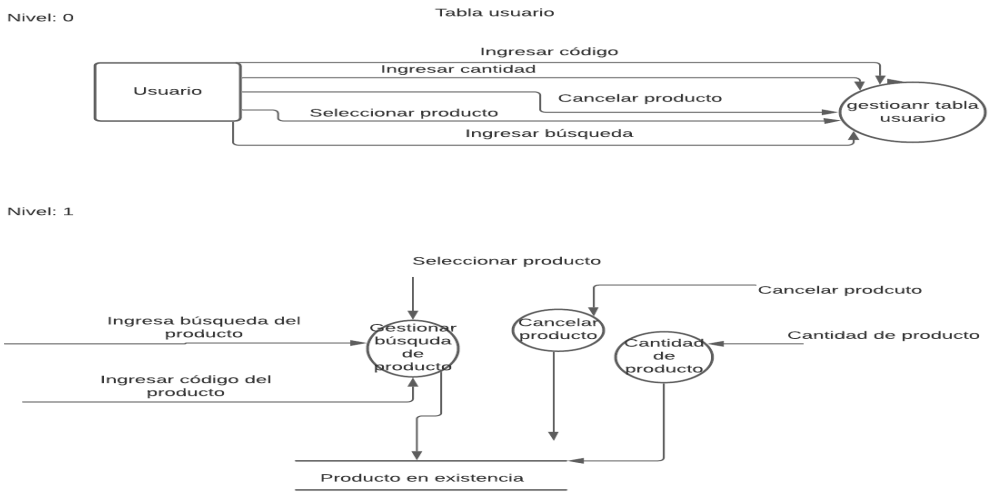


Figura 3.11: Diagrama de Flujo de datos Tabla usuario.

En este diagrama se puede observar de manera general como es que el usuario gestiona la tabla usuario para realizar una venta de un producto móvil.

3.2.2. Diagrama de clases

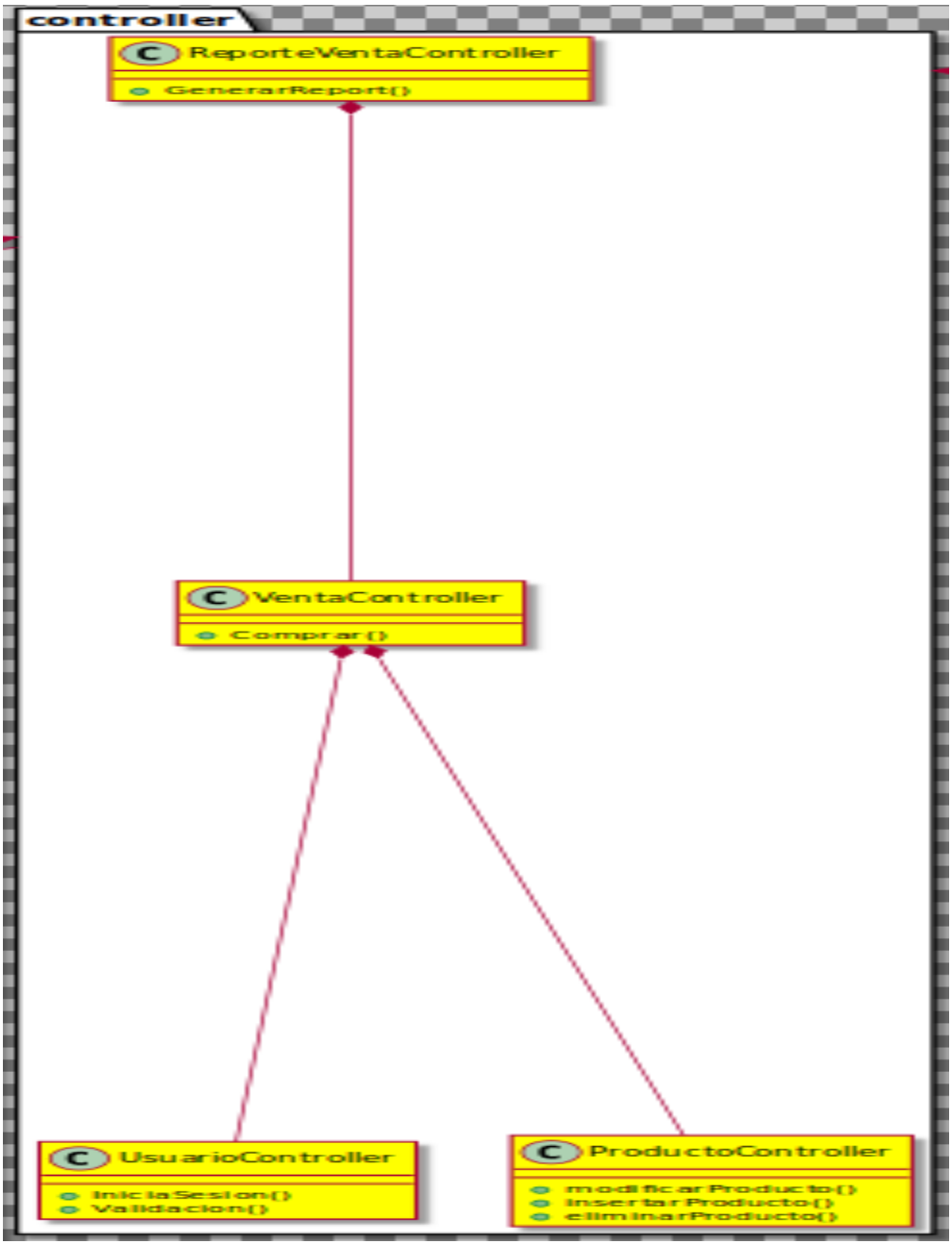


Figura 3.12: Diagrama de Clases

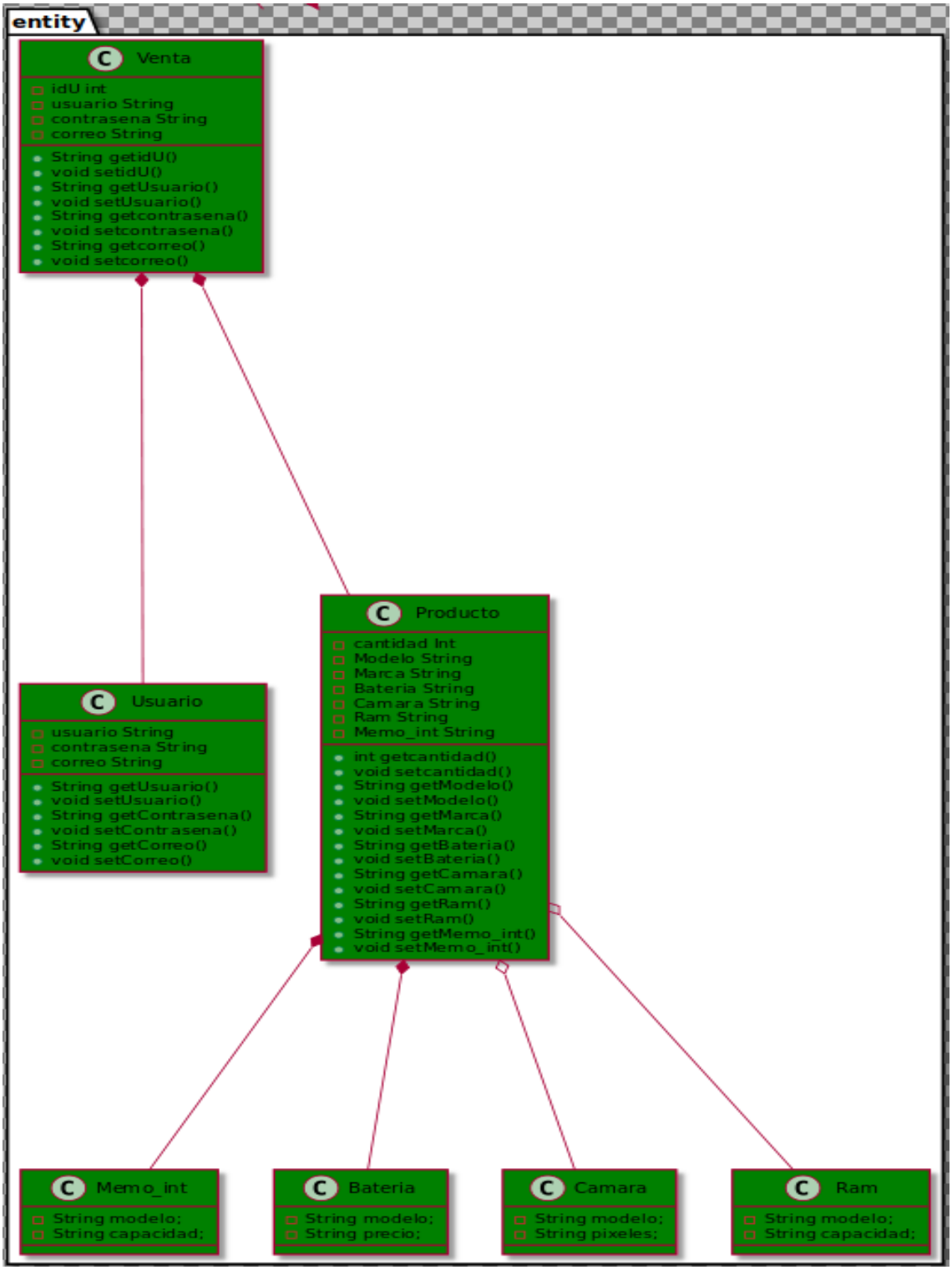


Figura 3.13: Diagrama de Clases

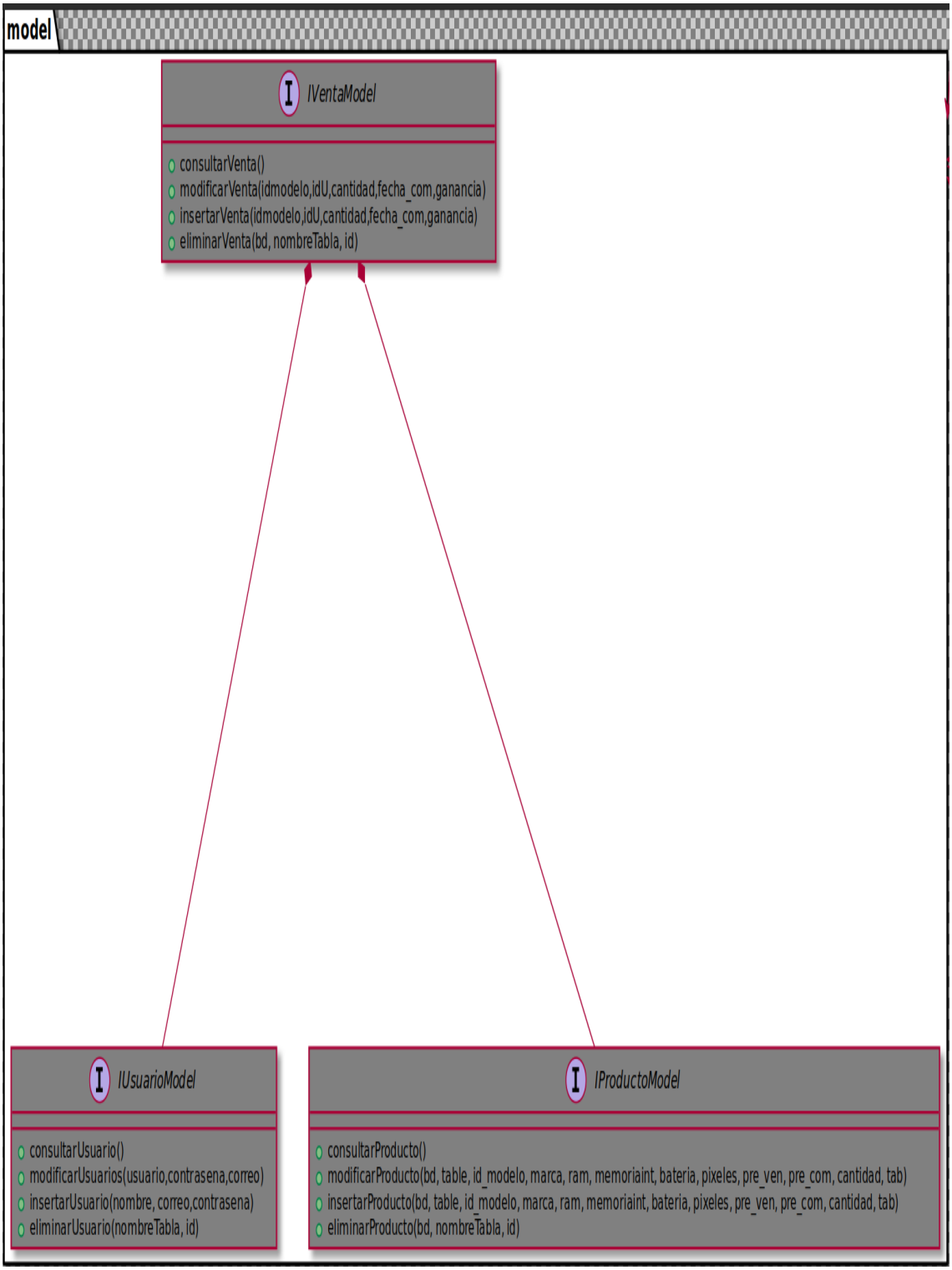


Figura 3.14: Diagrama de Clases

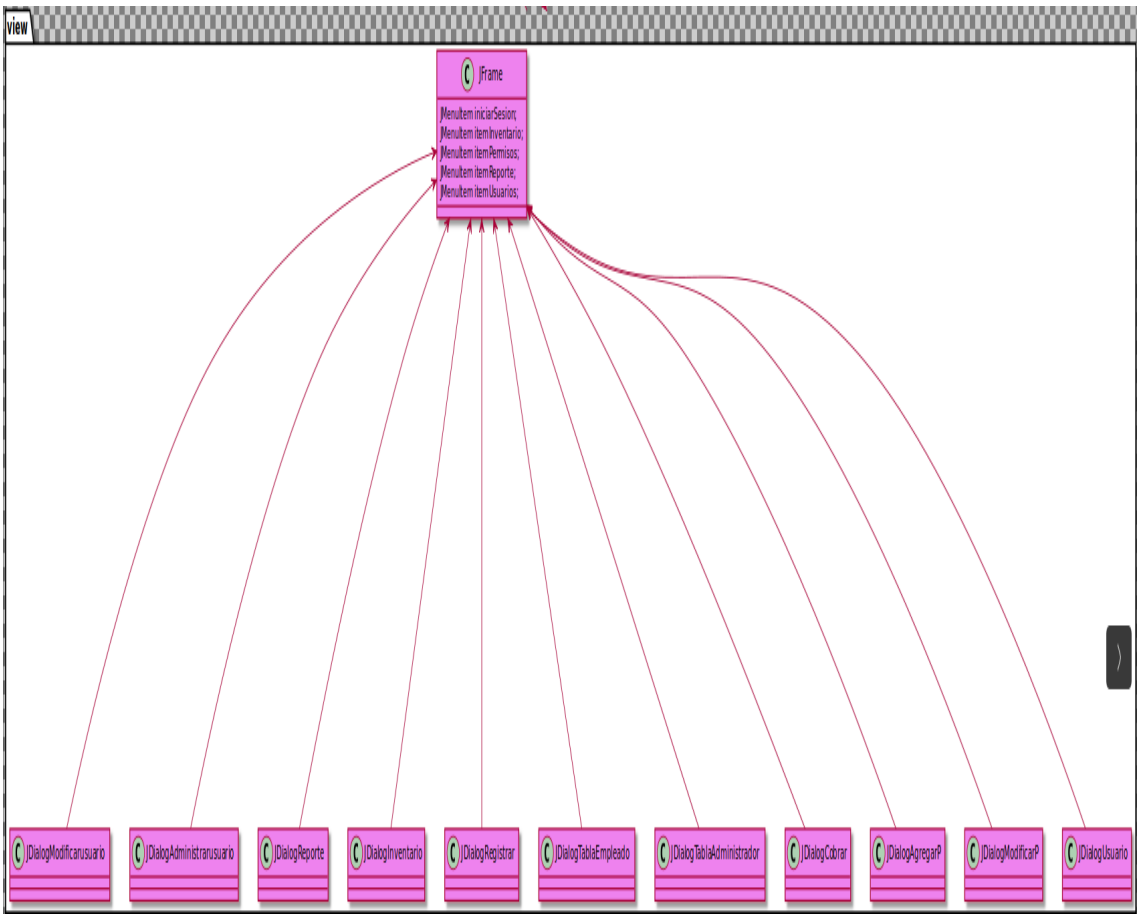


Figura 3.15: Diagrama de Clases

Este Diagrama de Clases nos permite trazan claramente la estructura de nuestro sistema de ventas al modelar sus clases, atributos, operaciones y relaciones entre objetos ya que es muy esencial para la creacion de nuestro sistema.

```
Codigo de Diagrama de Clases

@startuml
class bd.Conexion #orange{
+String pass
+String user
+String host

+Connection conectaServer()
+Connection conectaBase(nombrebd)
+ResultSet ejecutaQuery(nombrebd, sql)

}
```

```
class entity.Usuario #green{
-usuario String
-contrasena String
-correo String


+String getUsuario()
+void setUsuario()
+String getContrasena()
+void setContrasena()
+String getCorreo()
+void setCorreo()

}
class entity.Venta #green{
-idU int
-usuario String
-contrasena String
-correo String

+String getidU()
+void setidU()
+String getUsuario()
+void setUsuario()
+String getcontrasena()
+void setcontrasena()
+String getcorreo()
+void setcorreo()


}
class entity.Producto #green{
-cantidad Int
-Modelo String
-Marca String
-Bateria String
-Camara String
-Ram String
-Memo_int String
+int getcantidad()
+void setcantidad()
+String getModelo()
+void setModelo()
+String getMarca()
+void setMarca()
+String getBateria()
+void setBateria()
+String getCamara()
+void setCamara()
```

```
+String getRam()
+void setRam()
+String getMemo_int()
+void setMemo_int()

}

class entity.Memo_int #green{
-String modelo;
-String capacidad;
}

class entity.Bateria #green{
-String modelo;
-String precio;
}

class entity.Camara #green{
-String modelo;
-String pixeles;
}

class entity.Ram #green{
-String modelo;
-String capacidad;
}


interface model.IUsuarioModel #gray{


+consultarUsuario()
+modificarUsuarios(usuario,contrasena,correo)
+insertarUsuario(nombre, correo,contrasena)
+eliminarUsuario(nombreTabla, id)
}
interface model.IVentaModel #gray{
+consultarVenta()
+modificarVenta(idmodelo,idU,cantidad,fecha_com,ganancia)
+insertarVenta(idmodelo,idU,cantidad,fecha_com,ganancia)
+eliminarVenta(bd, nombreTabla, id)
}
interface model.IProductoModel #gray{
+consultarProducto()
+modificarProducto(bd, table, id_modelo, marca, ram, memoriaint, bateria, pixeles, pre_ven, pre_com,
+insertarProducto(bd, table, id_modelo, marca, ram, memoriaint, bateria, pixeles, pre_ven, pre_com, c
+eliminarProducto(bd, nombreTabla, id)
}
class controller.ReporteVentaController #yellow{
```

```
+GenerarReport()

}
class controller.UsuarioController #yellow{
+IniciaSesion()
+Validacion()

}
class controller.VentaController #yellow{
+Comprar()

}
class controller.ProductoController #yellow{
+modificarProducto()
+insertarProducto()
+eliminarProducto()
}

class view.JFrame #violet{
JMenuItem iniciarSesion;
JMenuItem itemInventario;
JMenuItem itemPermisos;
JMenuItem itemReporte;
JMenuItem itemUsuarios;
}


class view.JDialogInventario #violet{
}

class view.JDialogRegistrar #violet{
}


class view.JDialogTablaEmpleado #violet{
}


class view.JDialogTablaAdministrador #violet{
}

class view.JDialogCobrar #violet{
}


class view.JDialogAgregarP #violet{
}
```

```
class view.JDialogModificarP #violet{
}


class view.JDialogUsuario #violet{
}


class view.JDialogModificarusuario #violet{
}


class view.JDialogAdministrarusuario #violet{
}


class view.JDialogReporte #violet{
}

bd <--- model
bd ---> model

controller <--- model
controller ---> model


controller <--- view
controller ---> view

model <--- entity
model ---> entity


entity.Producto *---- entity.Memo_int
entity.Producto *---- entity.Bateria
entity.Producto o---- entity.Camara
entity.Producto o---- entity.Ram
entity.Venta *---- entity.Producto
entity.Venta *---- entity.Usuario

view.JFrame <---- view.JDialogInventario
view.JFrame <---- view.JDialogRegistrar
view.JFrame <---- view.JDialogTablaEmpleado
view.JFrame <---- view.JDialogTablaAdministrador
view.JFrame <---- view.JDialogCobrar
view.JFrame <---- view.JDialogAgregarP
view.JFrame <---- view.JDialogModificarP
view.JFrame <---- view.JDialogUsuario
view.JFrame <---- view.JDialogModificarusuario
```

```
view.JFrame <---- view.JDialogAdministrarusuario
view.JFrame <---- view.JDialogReporte


model.IVentaModel *---- model.IProductoModel
model.IVentaModel *---- model.IUsuarioModel

controller.VentaController *---- controller.ProductoController
controller.VentaController *---- controller.UsuarioController
controller.ReporteVentaController *----- controller.VentaController
@enduml
```

3.2.3. Diagrama de Objetos

Figura 3.16: Diagrama de Objetos.
Este diagrama de objetos nos muestra una instancia de nuestros diagramas de clases y se enfoca en los atributos y sus relaciones que tienen los objetos en nuestro sistema de ventas y así como también nos describen la estructura estática de nuestro sistema.

```
@startuml

object Usuario #green{
usuario = "maria"
contrasena = 123
numero =9512344238
}
object Venta #green{
idU = 1
usuario = "raul"
contrasena = 321
numero = 9510435604
}
object Producto #green{
cantidad = 1
Modelo = "G30"
Marca = "LG"
Bateria = 4200
Camara = 42
Ram = 6
Memo_int = 16
}
object Memo_int #green{
Modelo = "G30"
capacidad = 64
}
object Bateria #green{
Modelo = "G30"
precio = 3000
}
object Camara #green{
Modelo = "G30"
pixeles = 42
}
object Ram #green{
Modelo = "G30"
capacidad = 8
}
Producto *---- Memo_int
Producto *---- Bateria
Producto o---- Camara
Producto o---- Ram
Venta *---- Producto
Venta *---- Usuario

@enduml
```

3.2.4. Diagrama de Componentes

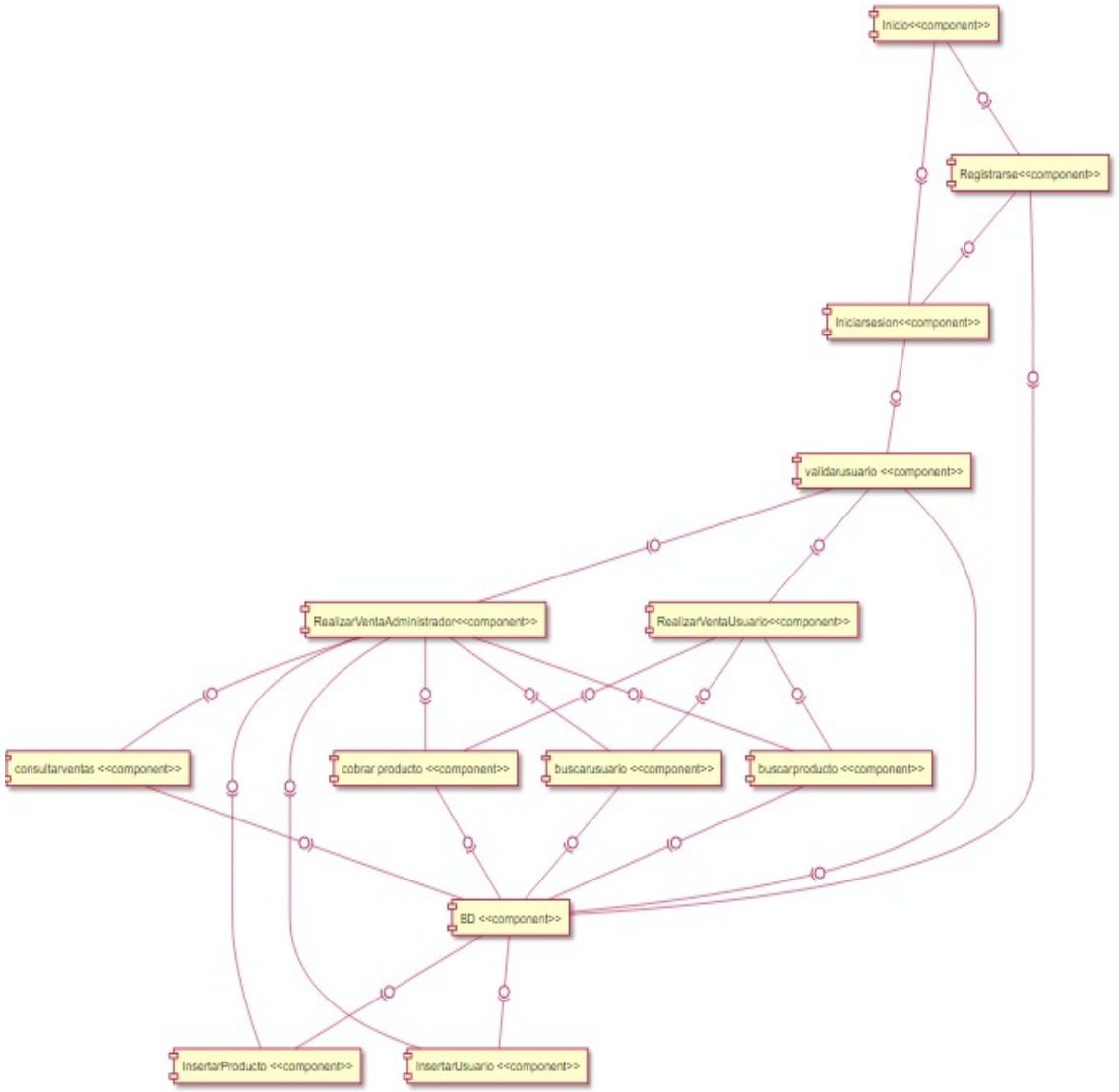


Figura 3.17: Diagrama de Componentes
Este diagrama nos proporciona una visión orientada a la ejecución de nuestro sistema ya que con ella se puede ver las organización de nuestros componentes , también en este diagrama se puede observar las relaciones que tiene cada componente en nuestro sistema de ventas.

Codigo de Diagrama de Componentes

```
    @startuml
[BD <<component>>] as b
[Inicio<<component>>] as i
[Iniciarsesion<<component>>] as is
[Registrarse<<component>>] as r
[RealizarVentaAdministrador<<component>>] as rva
[RealizarVentaUsuario<<component>>] as rvu
[InsertarProducto <<component>>] as p
[InsertarUsuario <<component>>] as u
[buscarproducto <<component>>] as bp
[buscarusuario <<component>>] as bu
[consultarventas <<component>>] as cv
[validarusuario <<component>>] as buu
[cobrar producto <<component>>] as cp
i --0)- is
i --0)- r
r --0)- is
is --0)-  buu
buu --0)- rva
buu --0)- rvu
rva --0)- p
rva --0)- u
rvu --0)- bp
rvu --0)- bu
rva --0)- bp
rva --0)- bu
rva --0)- cv
rva --0)- cp
rvu --0)- cp
b --0)-  p
b --0)-  u
bp --0)- b
bu --0)- b
cv --0)- b
buu --0)- b
cp --0)- b
r --0)- b
@enduml
```

3.2.5. Diagrama de Despliegue

Figura 3.18: Diagrama de Despliegue de SFH.
El Diagrama de despliegue que creamos nos muestran la arquitectura de ejecución de nuestro sistema de ventas y en este también se puede observar los enlaces de comunicación que tiene cada uno de los nodos de nuestro sistema ya que también nos ayudo a comprender como se desplegará el sistema de ventas físicamente.

Codigo de Diagrama de Despliegue

```
@startuml

node Base_de_datos as bd
node Framer_Inicio  as fi
node JDialog_Registrarse as jr
node JDialog_iniciar_sesion as ji
node JDialog_Administrador  as ja
node JDialog_Usuario as ju
node JDialog_cobrar as jc
node JDialog_Inventario as i
node JDialog_Reporte as r
node JDialog_Administrar_usuario as au
node JDialog_Agregar_producto as ap
node JDialog_Modificar_producto as mp
node JDialog_Agregar_usuario as jau
node JDialog_Modificar_usuario as jmu

fi -- jr
fi -- ji
jr -- ji
ji -- ja
ji -- ju
ju -- jc
ja -- au
ja -- ap
ja -- i
i -- ap
i -- mp
i -- r
au -- jau
au -- jmu
ji -- bd:JDBC
ja -- jc
ap -- bd:JDBC
mp -- bd:JDBC
jau -- bd:JDBC
jmu -- bd:JDBC
jc -- bd:JDBC



@enduml
```

3.3. Diagramas de comportamiento

3.3.1. Diagramas de casos de uso

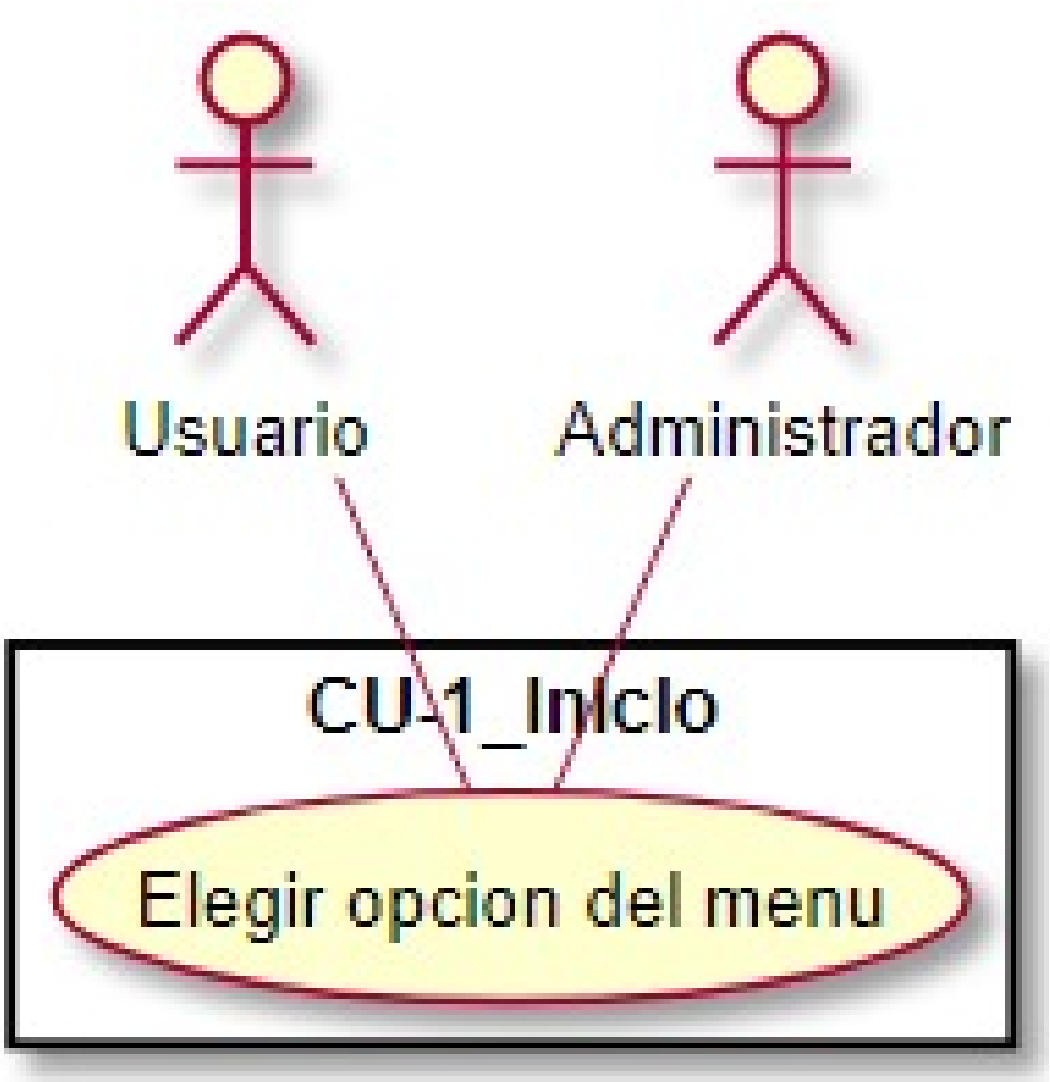


Figura 3.19: Diagrama de casos de uso de inicio

En el diagrama de caso de uso de iniciar sesión se muestra el comportamiento y la estructura de esta pantalla en la cual se puede observar que tenemos dos actores que es el administrador, el usuario y también se pueden observar nuestro caso de se uso que es elegir opción del menú.

ID:	CU 1		
Nombre:	Inicial		
Creado por:	L.G.SJ	Actualizado de:	G.R.F
Fecha creación:	10/10/2021	Fecha última versión	15/10/2021
Actores:	Administrados, Usuario, base de datos		
Descripción:	Es la pantalla principal al ingresar.		
Disparador:	Selecciona la opción iniciar sección de la pantalla principal. Selecciona la opción registrar de la pantalla principal.		
Precondiciones:	N/A		
Poscondiciones:	Se inicio sección y se despliega la pantalla iniciar sección. Se registra y se muestra la pantalla registrar.		
Flujo normal:	Se selecciona inicio sección y se despliega la pantalla iniciar sección. Se selecciona registrar y se muestra la pantalla registrar.		
Flujos alternos:	Al seleccionar se va a la pantalla de registro o inicio.		
X includes:			
Frecuencia de uso:	Muy frecuente.		
Requisitos especiales:	N/A		
Supuestos:			
Issues y notas			

Figura 3.20: Documentación de diagrama de casos de uso de inicio

```
Codigo de Diagrama de diagrama

@startuml
actor "Usuario" as u
actor "Administrador" as a
rectangle CU-1_Inicio {
usecase "Elegir opcion del menu" as EO
}
u - EO
a - EO
@enduml
```

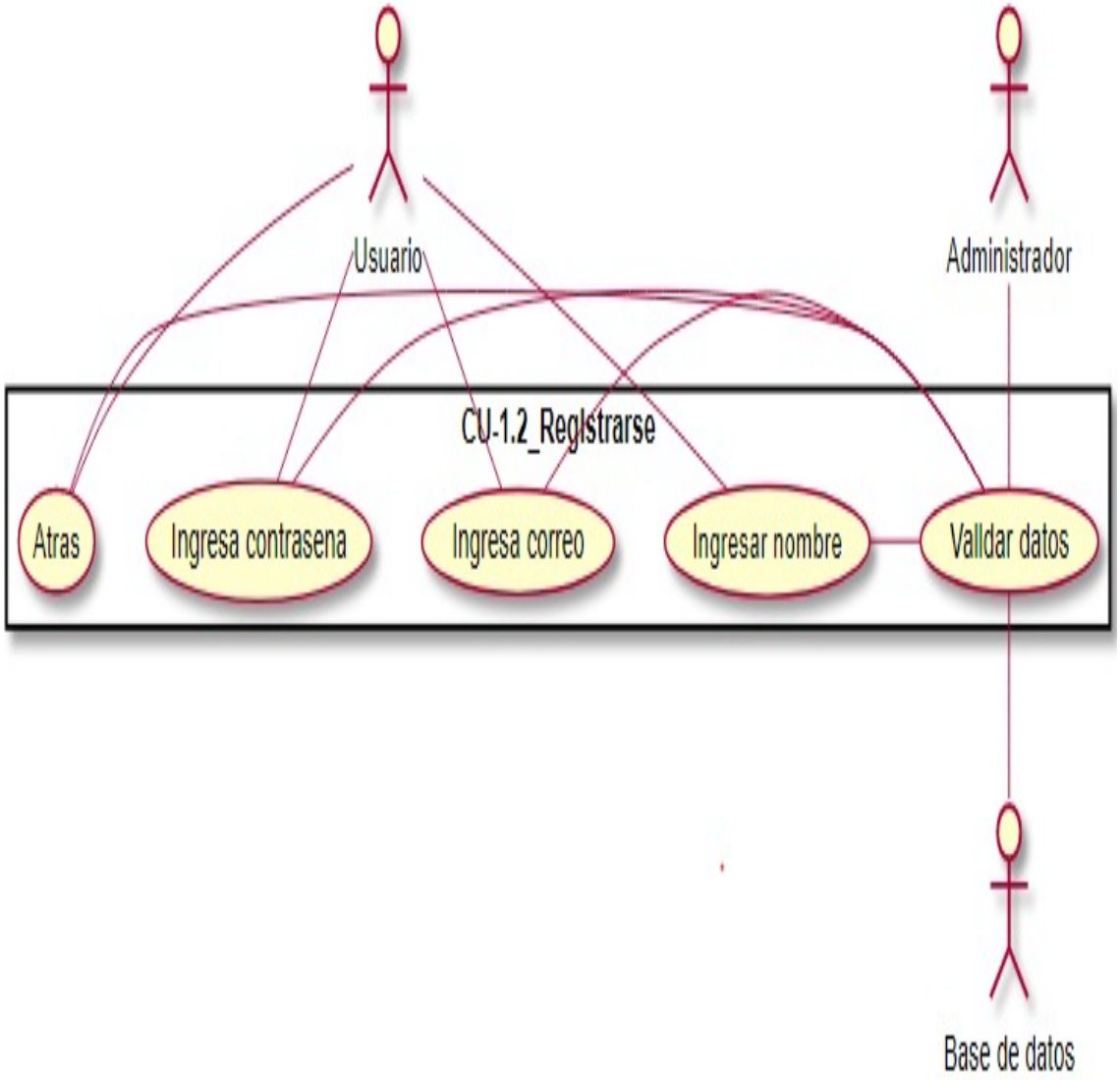


Figura 3.21: Diagrama de casos de uso de Registrase

ID:	CU 1.2		
Nombre:	Registrar		
Creado por:	G.R.F	Actualizado de:	L.G.S.J
Fecha creación:	8/10/2021	Fecha última versión	11/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Es la pantalla donde se pueden registrar usuarios nuevos.		
Disparador:	Si se le da en el botón guardar y se comprueba que ingresa la misma contraseña, se va a la pantalla de iniciar sesión.		
Precondiciones:	N/A		
Poscondiciones:	Se guarda y se despliega la pantalla iniciar sección		
Flujo normal:	Se inserta el nombre, apellido, contraseña y correo y se guarda y se despliega la pantalla iniciar sesión.		
Flujos alternos:	Al terminar de agregar los datos se va a la pantalla iniciar sesión.		
X includes:			
Frecuencia de uso:	Frecuente regular.		
Requisitos especiales:	N/A		
Supuestos:			
Issues y notas			

Figura 3.22: Documentación de diagrama de casos de uso de Registrase

El diagrama de caso de uso de la pantalla registrarse nos muestra el comportamiento de nuestra pantalla registrarse, en la que tenemos dos actores que son el administrador y el usuario en el cual el administrador solo hace la validación con la base de datos y el usuario tiene cinco casos de uso que son ingresar nombre, ingresar correo, ingresar contraseña y atrás y si los campos son correctos y el la validación del administrado es correcta entonces se hace una inserción en la base de datos.

```
Codigo de Diagrama de casos de uso de Registrase

@startuml
actor "Usuario" as u
actor "Administrador" as a
rectangle CU-1.2_Registrarse {
```

```
usecase "Ingresar nombre" as IN
usecase "Ingresa correo" as IC
usecase "Ingresa contrasena" as CI
usecase "Atras" as AT
usecase "Validar datos" as VD
}
u - IN
u - IC
u - CI
u - AT
a - VD
IN - VD
IC - VD
CI - VD
AT - VD

actor "Base de datos" as BD
VD - BD
@enduml
```

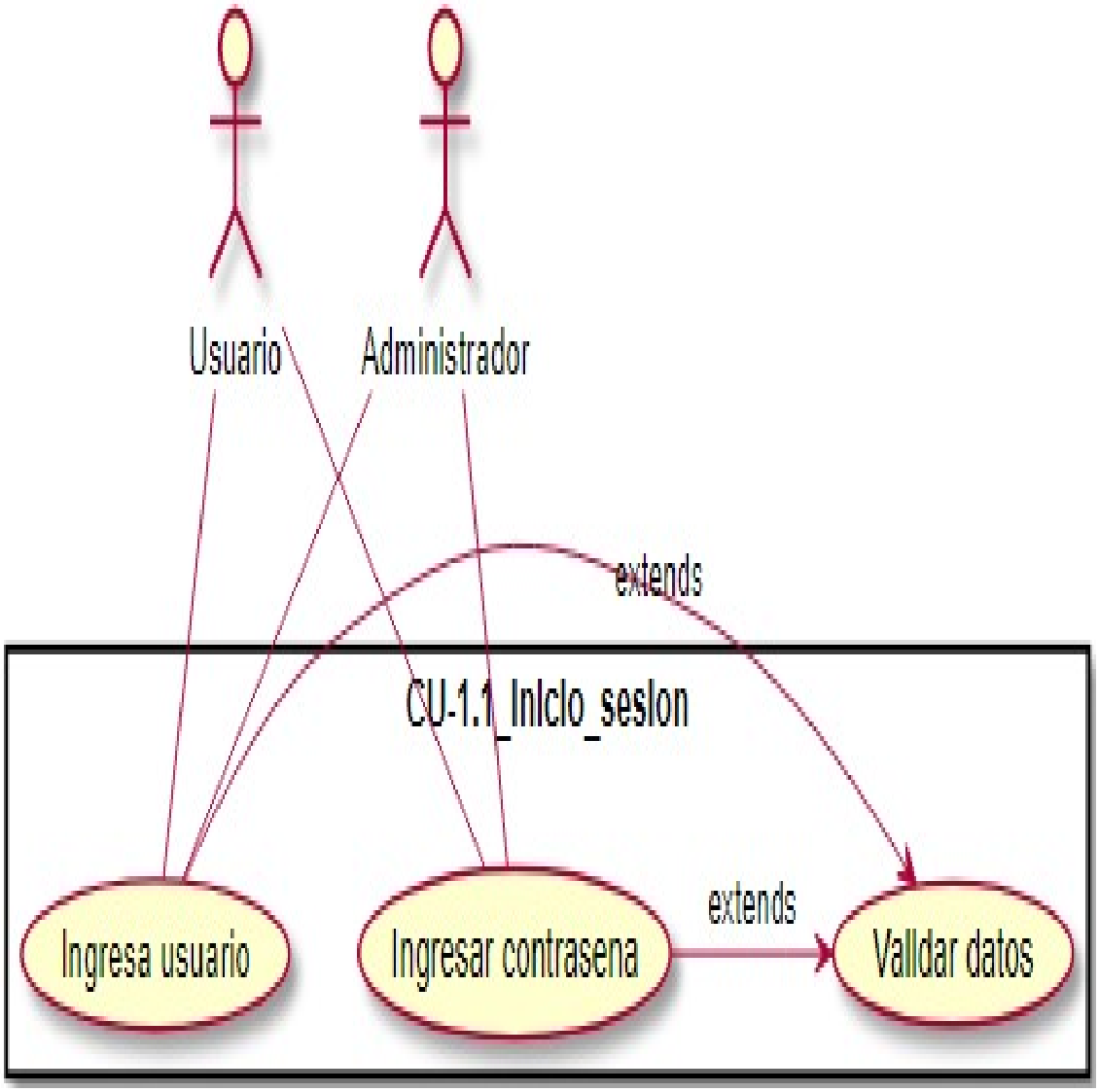


Figura 3.23: Diagrama de casos de uso de inicio de sesión

ID	CU 1.1		
Nombre:	Iniciar sesión		
Creado por:	G.R.F	Actualizar	L.G.S.J
Fecha de creación:	8/10/2021	Fecha última versión	11/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Es la pantalla donde se puede registrar usuarios nuevos.		
Disparador:	Si se le da en el botón guardar y se comprueba que ingresa la misma contraseña, se va a la pantalla de inicio de sesión.		
Precondiciones:	Se guarda y se despliega la pantalla inicio de sesión.		
Flujo normal:	Se inserta el nombre, apellido, contraseña, correo y se guarda y se despliega la pantalla inicio de sesión.		
Flujos alternos:	Al terminar de agregar los datos se va a la pantalla inicio de sesión.		
X includes:			
Frecuencia de uso:	Frecuente.		
Requisitos especiales:	N/A		
Supuestos:			
Issu y notas:			

Figura 3.24: Documentación de diagrama de casos de uso de inicio de sesión

El diagrama de caso de uso de iniciar sesión nos muestra el comportamiento de nuestra pantalla iniciar sesión, en la cual se puede observar que tenemos dos actores que son el administrador y el usuario los cuales interactúan en los dos casos de uso que tenemos que son ingresar usuario y ingresar contraseña y estos dos casos de uso se validan en nuestra base de datos.

```
Codigo de Diagrama de casos de uso de inicio de sesión

@startuml
actor "Usuario" as u
actor "Administrador" as a
rectangle CU-1.1_Inicio_sesion {
usecase "Ingresa usuario" as IU
```

```
usecase "Ingresar contrasena" as IC
usecase "Validar datos" as VD
}
u - IU
u - IC
a - IU
a - IC
IU -> (VD) : extends
IC -> (VD) : extends
@enduml
```

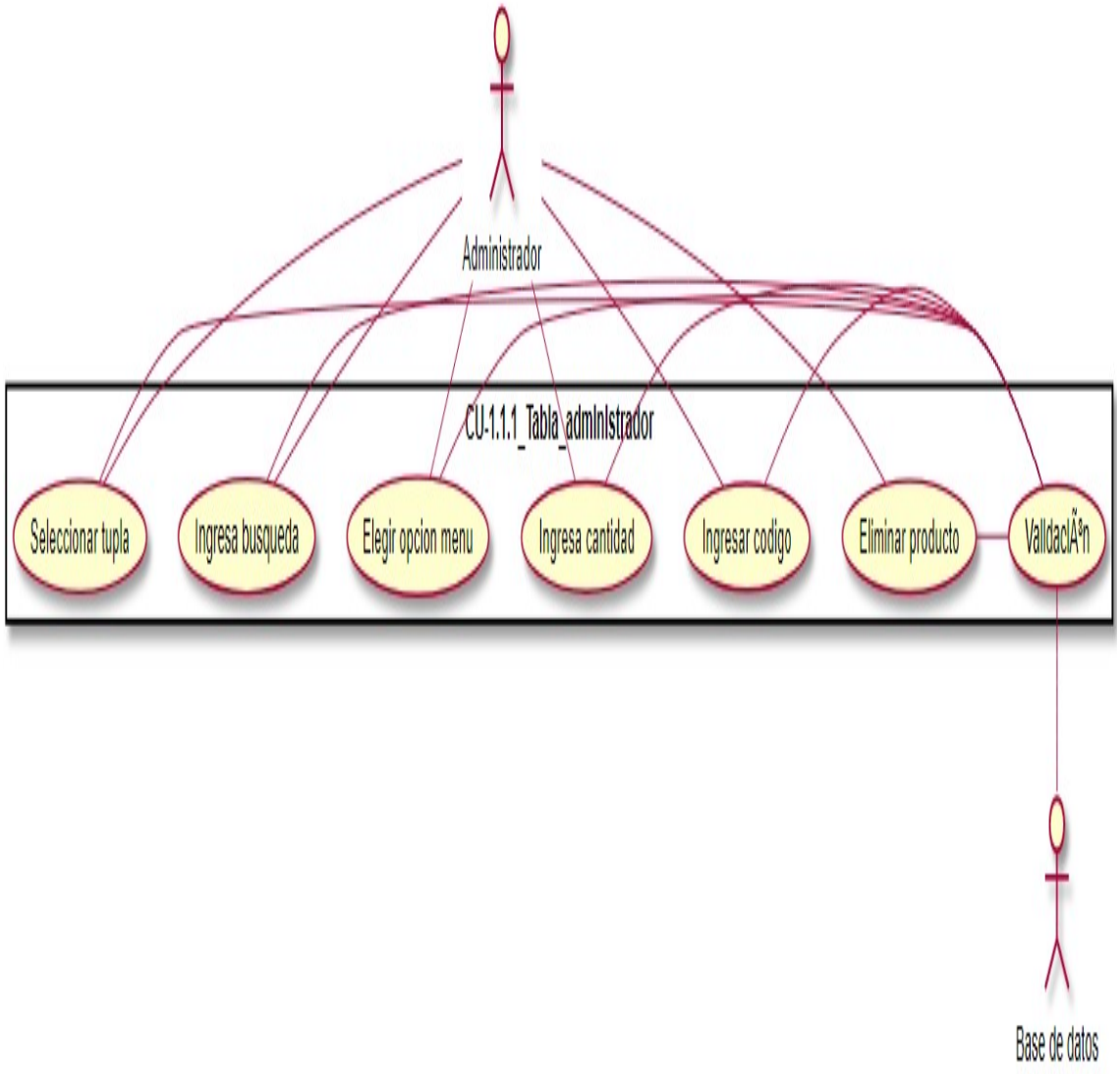


Figura 3.25: Diagrama de casos de uso de tabla administrador

ID:	CU 1.1.1		
Nombre:	Tabla administrador		
Creado por:	A.B.H	Actualizado de:	G.R.F
Fecha creación:	8/10/2021	Fecha última versión	11/16/2021
Actores:	Administrador, base de datos.		
Descripción:	Esta pantalla nos permite buscar, agregar y eliminar productos a nuestra compra, como también ir a diferentes ventanas.		
Disparador:	Si se le da clic en la opción de permisos no manda a la pantalla modificar usuario. Si se le da clic en la opción de usuarios no manda a la pantalla de usuarios. Si se le da clic en la opción de inventario nos manda al pantalla inventario. Si se le da clic en la opción de reporte nos manda a la pantalla de reportes. Si se le da clic en la opción de cobrar nos manda a la pantalla cobrar.		
Precondiciones:	N/A		
Poscondiciones:	Si se seleccionaron productos para comprar y le damos en cobrar no manda a la pantalla cobrar.		
Flujo normal:	Se selecciona una opción del ítem o se ingresa el código y se busca o se selecciona y elimina el que no deseamos o se selecciona y se cobran los productos seleccionados.		
Flujos alternos:	Al darle en cualquier opción del ítem nos manda otra pantalla o al seleccionar y darle en el botón cobrar nos manda a la pantalla cobrar.		
X includes:			
Frecuencia de uso:	Muy frecuente.		
Requisitos especiales:	N/A		
Supuestos:			
Issues y notas			

Figura 3.26: Documentación de diagrama de casos de uso de tabla administrador

En el diagrama de caso de uso de la pantalla administrador, se puede observar la estructura y el comportamiento de nuestra pantalla en la cual solo actuá administrador, el cual tiene seis casos de uso que son eliminar producto, ingresar código, ingresar cantidad, elegir opción del menú, ingresar búsqueda y seleccionar tupla en los cuales se hace una validación con la base de datos al utilizar cada uno.

```
Codigo de Diagrama de casos de uso de tabla administrador

@startuml

actor "Administrador" as a
rectangle CU-1.1.1_Tabla_administrador {
```

```
usecase "Eliminar producto" as EP
usecase "Ingresar codigo" as IC
usecase "Ingresa cantidad" as ICD
usecase "Elegir opcion menu" as EM
usecase "Ingresa busqueda" as IB
usecase "Seleccionar tupla" as ST
usecase "Validación" as VD
}

a - EP
a - IC
a - ICD
a - EM
a - IB
a - ST
EP - VD
IC - VD
ICD - VD
EM - VD
IB - VD
ST - VD
VD - BD
actor "Base de datos" as BD
@enduml
```

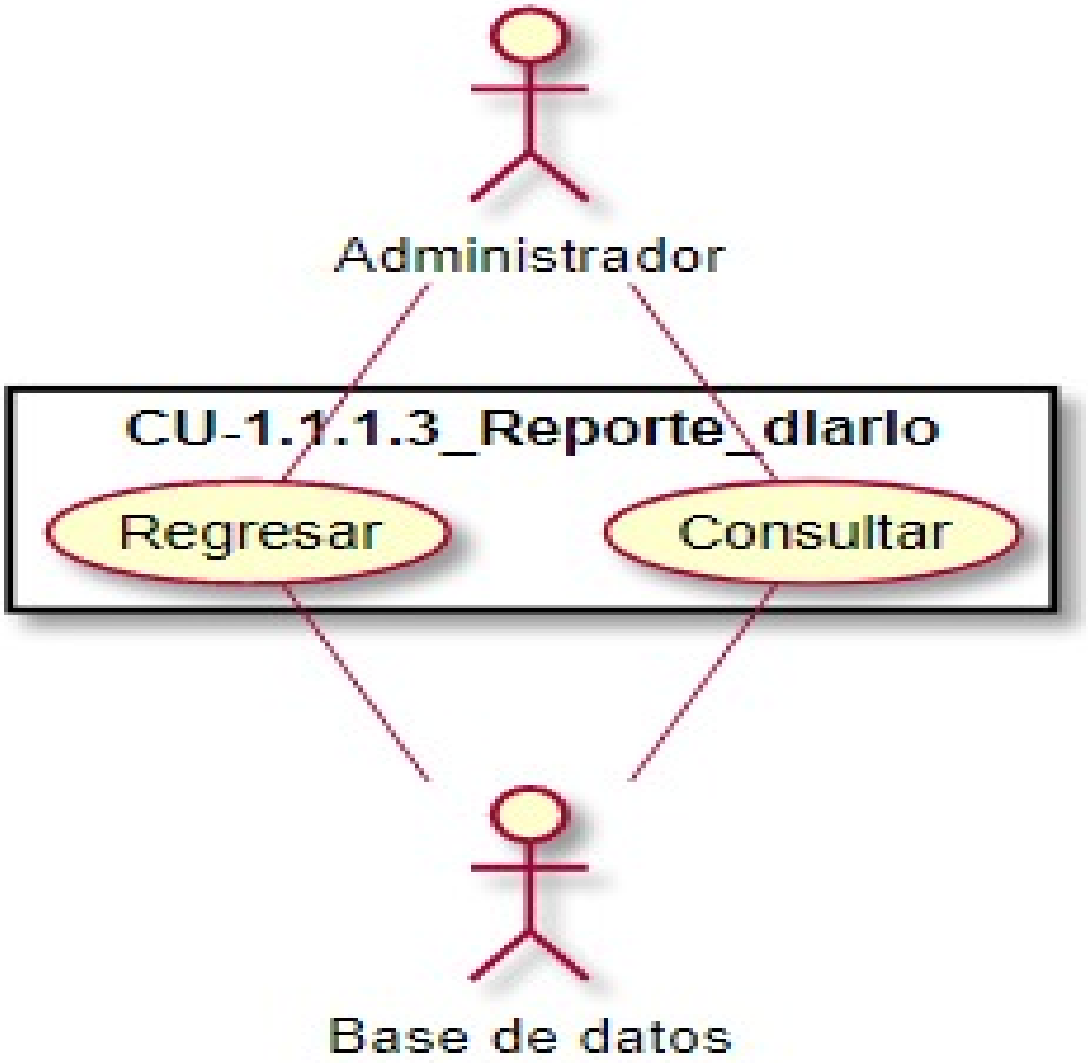


Figura 3.27: Diagrama de casos de uso de reporte diario

ID:	CU 1.1.1.3		
Nombre:	Reporte diario		
Creado por:	G.R.F	Actualizado de:	L.G.S.J
Fecha creación:	08/10/21	Fecha última versión	04/12/21
Actores:	Administrador y base de datos.		
Descripción:	Esta pantalla permite visualizar el reporte diario de las ventas de los productos.		
Disparador:	Se consultan los datos.		
Precondiciones:	N/A		
Poscondiciones:	Se vuelve a reiniciar la pantalla para pedir otra fecha.		
Flujo normal:	Se verifican que la fechas correspondan.		
Flujos alternos:	Al terminarse de ejecutar se regresa a la pantalla del administrador.		
X includes:			
Frecuencia de uso:	Muy alta		
Requisitos especiales:			
Supuestos:			
Issues y notas			

Figura 3.28: Documentación de diagrama de casos de uso de reporte diario

En el diagrama de caso de uso de la pantalla reporte diario, se puede observar la estructura y el comportamiento de nuestra pantalla en la que solo tenemos un actor que es el administrador el cual tiene dos casos de uso que son regresar y consultar el cual hace un llamado a la base de datos para hacer la consulta.

```
Codigo de Diagrama de casos de uso de reporte diario

@startuml

actor "Administrador" as a
rectangle CU-1.1.1.3_Reporte_diario {
usecase "Consultar" as C
```

```
usecase "Regresar" as R

}

a - C
a - R
C - BD
R -BD
actor "Base de datos" as BD
@enduml
```

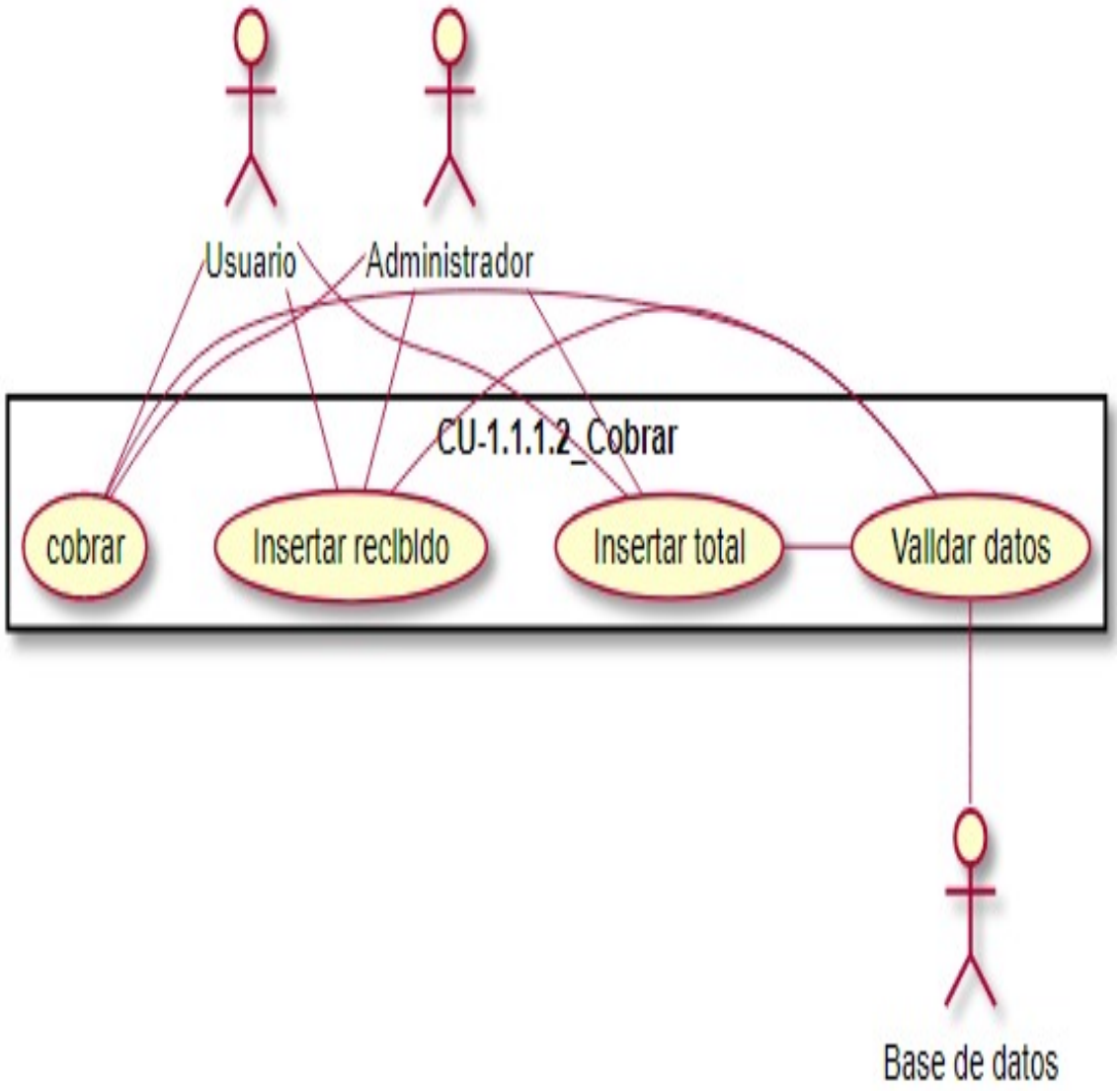


Figura 3.29: Diagrama de casos de uso de cobrar

ID:	CU 1.1.1.2		
Nombre:	Cobrar		
Creado por:	L.G.S.J	Actualizado de:	G.R.F
Fecha creación:	8/08/2021	Fecha última versión	11/152021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Es la pantalla donde se cobraran los productos vendidos.		
Disparador:	Si se le da en el botón de cancelar nos dirige a la pantalla de administrador o empleado.		
Precondiciones:	N/A		
Poscondiciones:	Si la cantidad que ingresa es correcta, se guarda y nos manda a la pantalla de administrador o empleado.		
Flujo normal:	Nos aparece el total, se ingresa el dinero recibido y se le da en el botón guardar.		
Flujos alternos:	Al terminar la venta nos manda a la pantalla de administrador o empleado.		
X includes:			
Frecuencia de uso:	Muy frecuente		
Requisitos especiales:	N/A		
Supuestos:			
Issues y notas			

Figura 3.30: Documentación de diagrama de casos de uso de cobrar

Este diagrama de caso de uso de la pantalla cobra nos muestra el comportamiento y la estructura de esta pantalla, la cual tiene dos actores que son el administrador, el usuario y los cuales tienen tres casos de uso que son insertar total, insertar recibido y cobrar y de todos se hacen una validación en la base de datos para que no haya datos incorrectos.

```
Codigo de Diagrama de casos de uso de cobrar

@startuml
actor "Usuario" as u
actor "Administrador" as a
rectangle CU-1.1.1.2_Cobrar {
usecase "Insertar total" as IT
```

```
usecase "Insertar recibido" as IR
usecase "cobrar" as C
usecase "Validar datos" as VD
}
u - IT
u - IR
u - C
a - IT
a - IR
a - C
IT - VD
IR - VD
C - VD
actor "Base de datos" as BD
VD - BD
@enduml
```

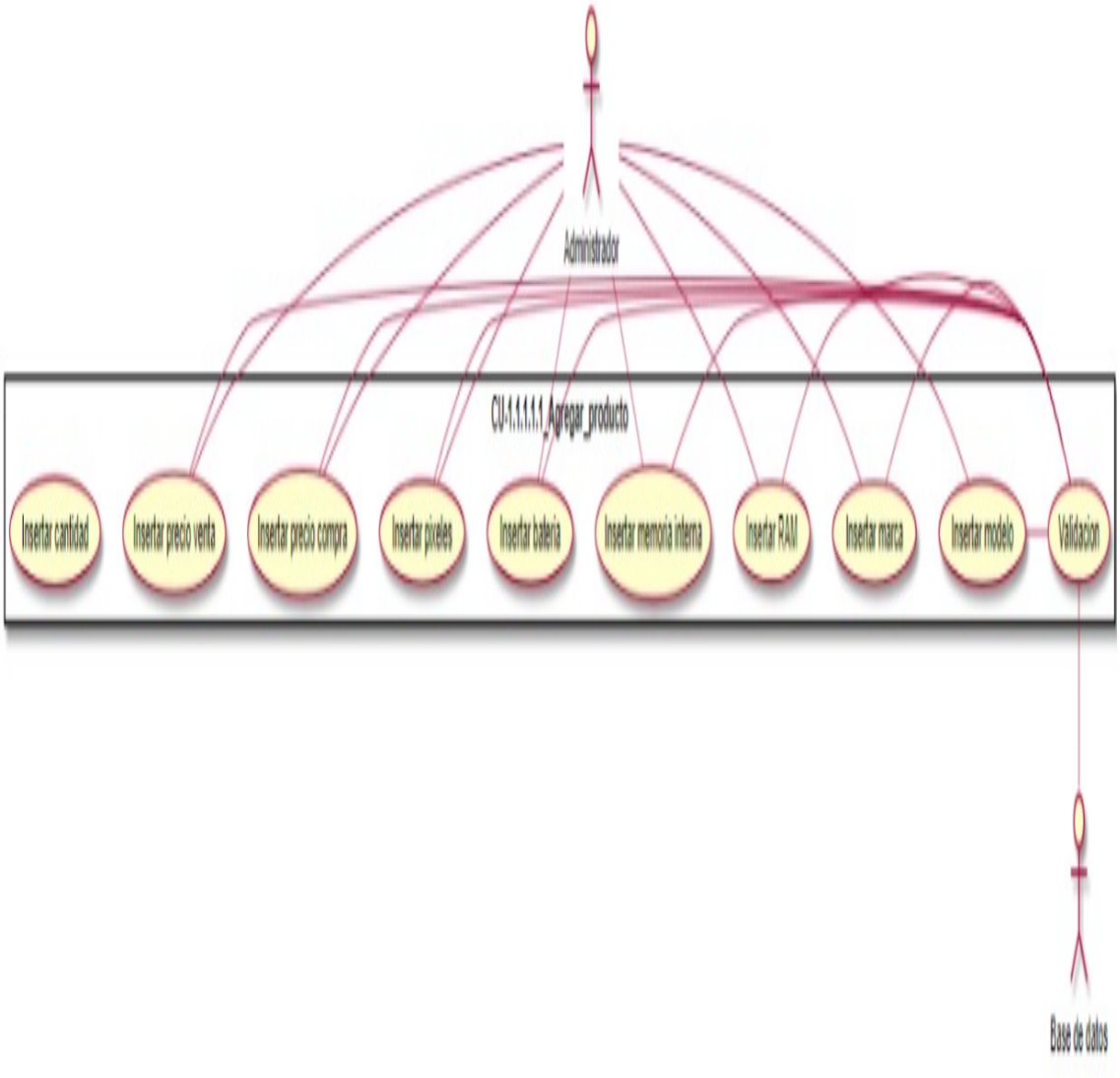


Figura 3.31: Diagrama de casos de uso de agregar producto

ID:	CU 1.1.1.1.1		
Nombre:	Agregar producto		
Creado por:	G.R.F	Actualizado de:	A.B.H
Fecha creación:	8/12/2021	Fecha última versión	16/10/2021
Actores:	Administrador, base de datos.		
Descripción:	Es la pantalla en donde podemos agregar productos nuevos.		
Disparador:	Si le da al botón guardar nos manda a la pantalla de inventario.		
Precondiciones:	N/A		
Poscondiciones:	Si los el usuario rellena bien todos los campos y le da en guardar nos manda a la pantalla de inventario.		
Flujo normal:	Insertamos el modelo, marca, ram, memoria interna ,batería, pixeles, precio compra, precio venta y cantidad y le damos al botón guardar.		
Flujos alternos:	Al guardar los datos nos manda a la pantalla Inventario.		
X includes:			
Frecuencia de uso:	Frecuencia regular.		
Requisitos especiales:	N/A		
Supuestos:			
Issues y notas			

Figura 3.32: Documentación de diagrama de casos de agregar producto

En este diagrama de caso de uso de la pantalla administrador, se puede observar la estructura y el comportamiento de nuestra pantalla agregar producto y este diagrama tiene un actor que es el administrador el cual tiene nueve casos de uso que son insertar modelo, insertar marca, insertar RAM, insertar memoria interna, insertar batería, insertar pixeles, insertar precio compra, insertar precio venta, insertar cantidad y de los cuales se hace una validación antes de ser insertados y después se insertan a nuestra base de datos.

```
Codigo de Diagrama de casos de uso de agregar producto

@startuml
actor "Administrador" as a
rectangle CU-1.1.1.1.1_Agregar_producto {
```

```
usecase "Insertar modelo" as IM
usecase "Insertar marca" as IMA
usecase "Insertar RAM" as IR
usecase "Insertar memoria interna" as IMI
usecase "Insertar bateria" as IB
usecase "Insertar pixeles" as IP
usecase "Insertar precio compra" as IPC
usecase "Insertar precio venta" as IPV
usecase "Insertar cantidad" as IC
usecase "Validacion" as VD
}
a - IM
a - IMA
a - IR
a - IMI
a - IB
a - IP
a - IPC
a - IPV
IM - VD
IMA - VD
IR - VD
IMI - VD
IB - VD
IP - VD
IPC - VD
IPV - VD
VD - BD
actor "Base de datos" as BD
@enduml
```

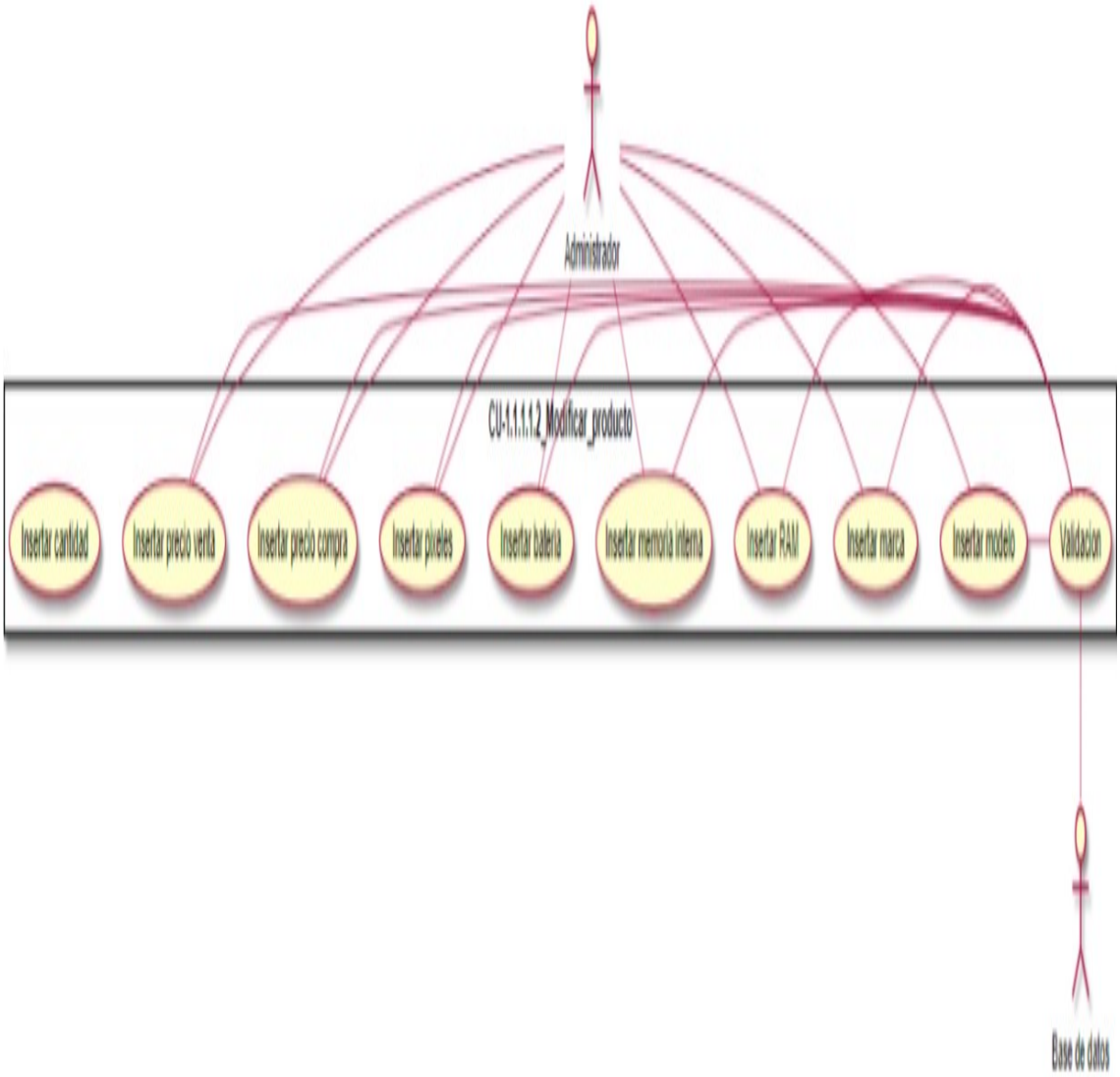


Figura 3.33: Diagrama de casos de uso de modificar producto

ID:	CU 1.1.1.1.2		
Nombre:	Modificar producto		
Creado por:	H.A.V	Actualizado de:	L.G.S.J
Fecha creación:	10/10/21	Fecha última versión	04/12/21
Actores:	Usuario.		
Descripción:	Es la pantalla que permite modificar los datos de los teléfonos.		
Disparador:	Se selecciona insertar modelo, se selecciona insertar marca, se selecciona insertar RAM, se selecciona insertar memoria interna, se selecciona insertar batería, se selecciona insertar pixeles, se selecciona insertar precio de venta y precio de compra, se selecciona insertar cantidad.		
Precondiciones:	N/A		
Poscondiciones:	Se reinician los valores y se vuelve a mostrar la pantalla.		
Flujo normal:	Verificar que los campos estén completamente y adecuadamente rellenados.		
Flujos alternos:	Al terminar de modificar productos, regresar a la pantalla de usuario.		
X includes:			
Frecuencia de uso:	Muy alta		
Requisitos especiales:			
Supuestos:			
Issues y notas			

Figura 3.34: Documentación de diagrama de casos de modificar producto

En el diagrama de caso de uso de la pantalla modificar producto, se puede observar la estructura y el comportamiento de nuestra pantalla modificar producto y este diagrama tiene un actor que es el administrador el cual tiene nueve casos de uso que son insertar modelo, insertar marca, insertar RAM, insertar memoria interna, insertar batería, insertar pixeles, insertar precio compra, insertar precio venta, insertar cantidad y de estos se hace una validación y después una modificación en nuestra base de datos.

```
Codigo de Diagrama de casos de uso de modificar producto

@startuml
actor "Administrador" as a
rectangle CU-1.1.1.1.2_Modificar_producto {
```

```
usecase "Insertar modelo" as IM
usecase "Insertar marca" as IMA
usecase "Insertar RAM" as IR
usecase "Insertar memoria interna" as IMI
usecase "Insertar bateria" as IB
usecase "Insertar pixeles" as IP
usecase "Insertar precio compra" as IPC
usecase "Insertar precio venta" as IPV
usecase "Insertar cantidad" as IC
usecase "Validacion" as VD
}
a - IM
a - IMA
a - IR
a - IMI
a - IB
a - IP
a - IPC
a - IPV
IM - VD
IMA - VD
IR - VD
IMI - VD
IB - VD
IP - VD
IPC - VD
IPV - VD
VD - BD
actor "Base de datos" as BD
@enduml
```

3.3.2. Diagrama de Estado

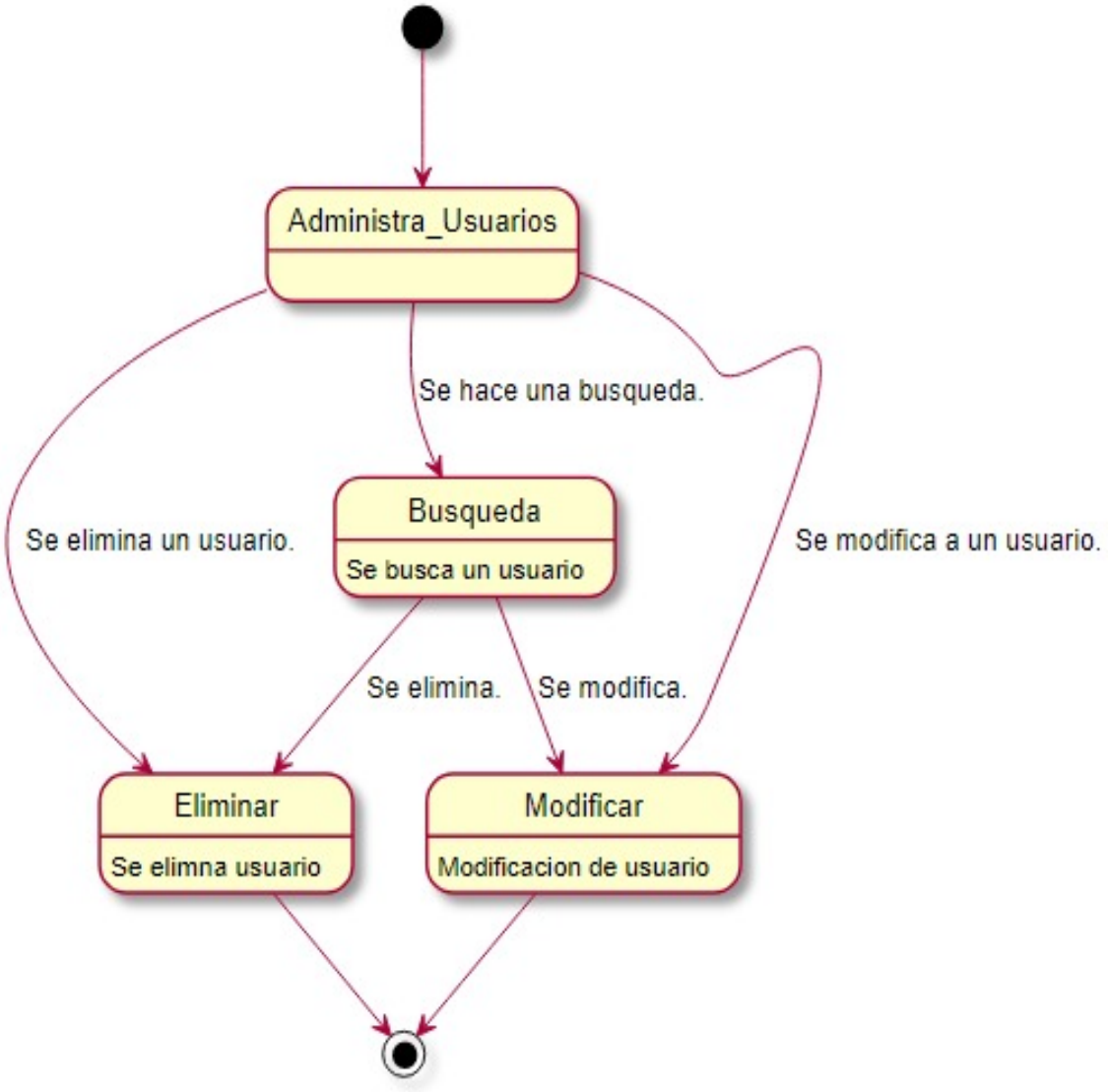


Figura 3.35: Diagrama de Estado Administrador

Tiene un estado inicial que es administrar usuario, si se realiza una busqueda llega al estado de busqueda, si se elimina un usuario llega al estado eliminado, y si se modifica un usuario se llega al estado de modificación.

```
@startuml
[*] --> Administra_Usuarios
```

```
Busqueda:Se busca un usuario
Eliminar: Se elimna usuario
Modificar: Modificacion de usuario
Administra_Usuarios --> Busqueda: Se hace una busqueda.
Administra_Usuarios --> Eliminar: Se elimina un usuario.
Administra_Usuarios --> Modificar: Se modifica a un usuario.
Busqueda --> Eliminar: Se elimina.
Busqueda --> Modificar: Se modifica.

Eliminar --> [*]
Modificar --> [*]
@enduml
```

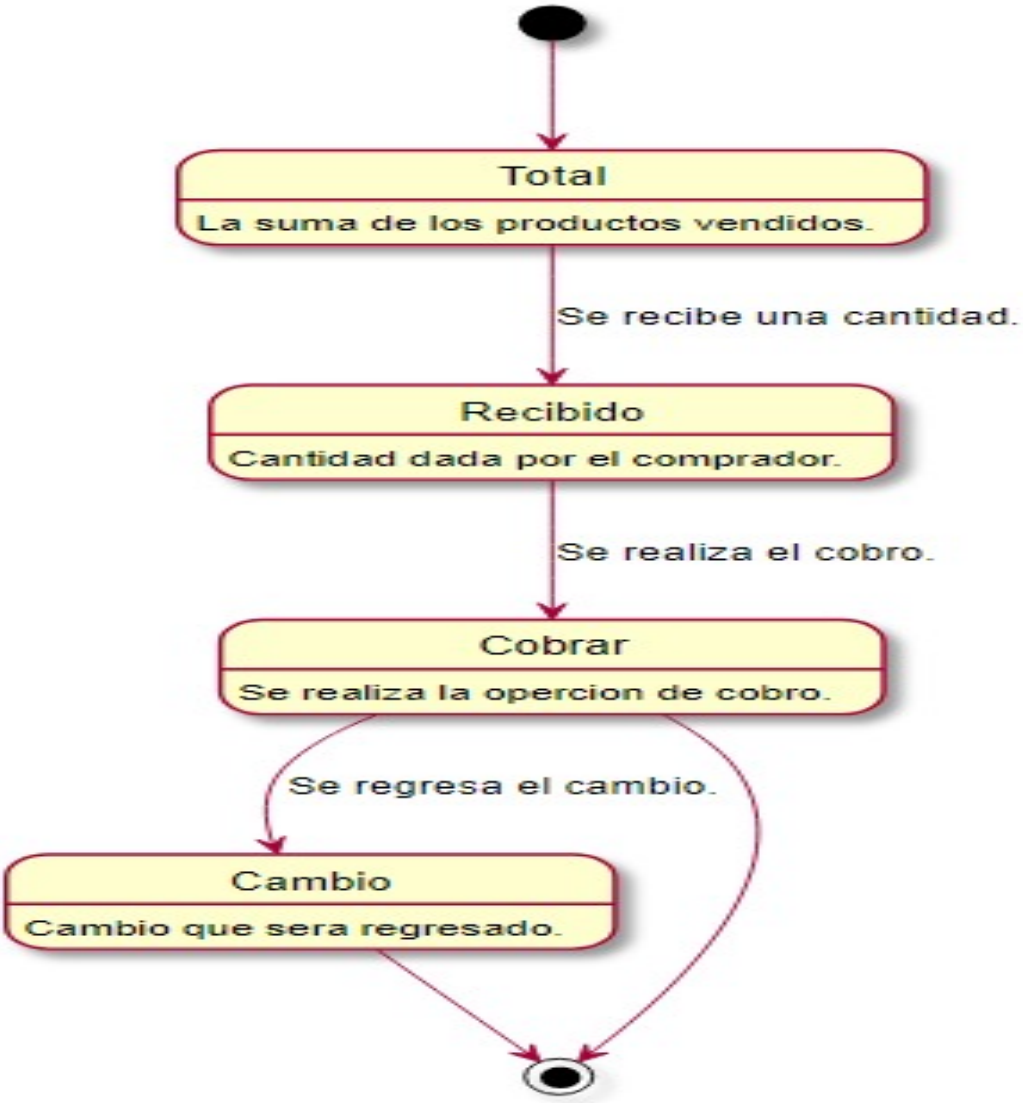


Figura 3.36: Diagrama de Estado Cobrar

Es ente diagama el estado inicial es tener un total, posteriormente se recibe una atidad de dinero, pasa a estado recibido, despues se realiza el cobro y pasa a estado cobrado, si la cantidad dada es mayor, pasa al estado de cambio, y si es exacto, se termina el cobro.

```
@startuml
[*] --> Total
Total: La suma de los productos vendidos.
Recibido: Cantidad dada por el comprador.
Cobrar: Se realiza  la opercion de cobro.
Cambio: Cambio que sera regresado.
Total --> Recibido: Se recibe una cantidad.
```

```
Recibido --> Cobrar: Se realiza el cobro.
Cobrar --> Cambio: Se regresa el cambio.
Cobrar --> [*]
Cambio --> [*]
@enduml
```

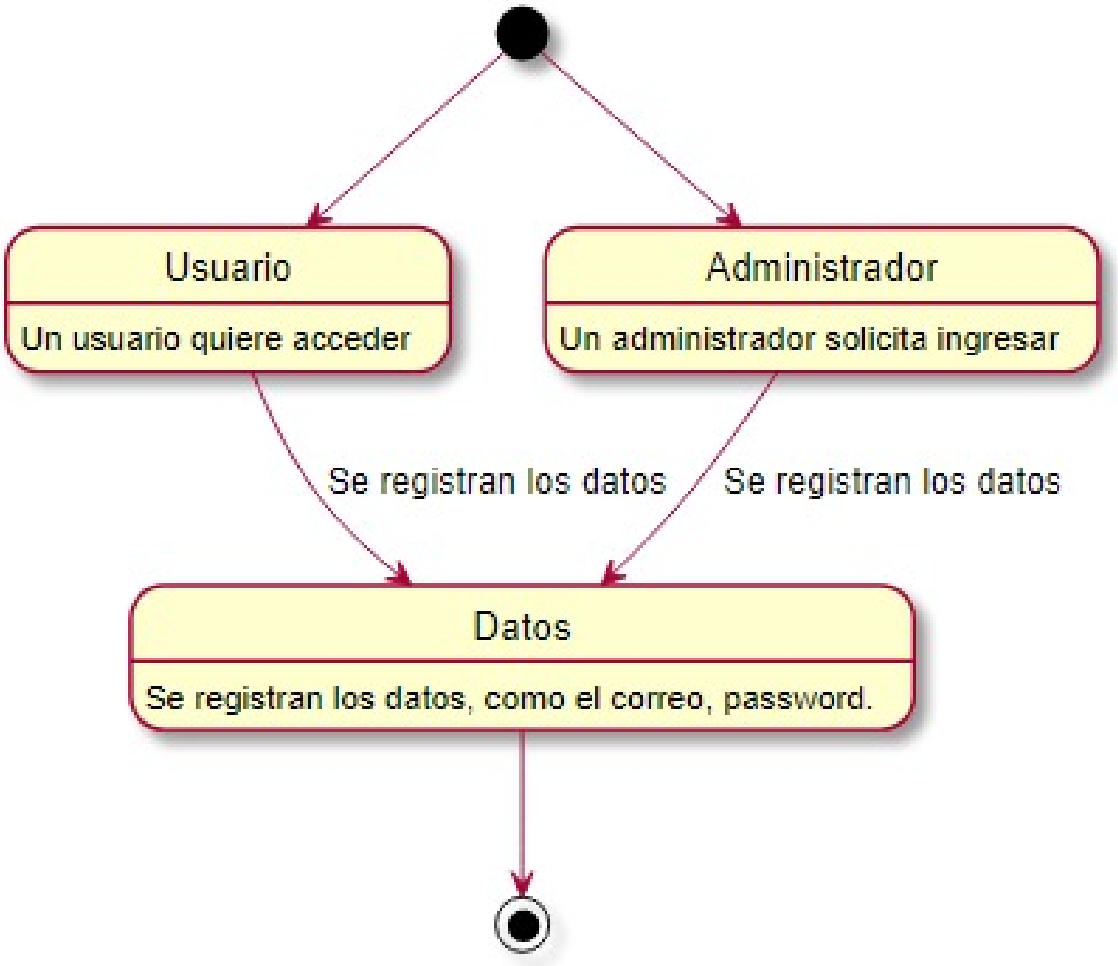


Figura 3.37: Diagrama de Estado Iniciar sesión

El estado inicial tiene dos variantes, una es el usuario y la otra el administrador, el cual dependiendo de la persona accederá registrando sus datos, al registrar sus datos, pasará al estado final, el cual es que pudo acceder.

```
@startuml
[*] --> Usuario
[*] --> Administrador
```

```
Usuario :Un usuario quiere acceder
Datos : Se registran los datos, como el correo, password.
Administrador : Un administrador solicita ingresar

Usuario --> Datos: Se registran los datos
Administrador --> Datos :Se registran los datos
Datos --> [*]
@enduml
```

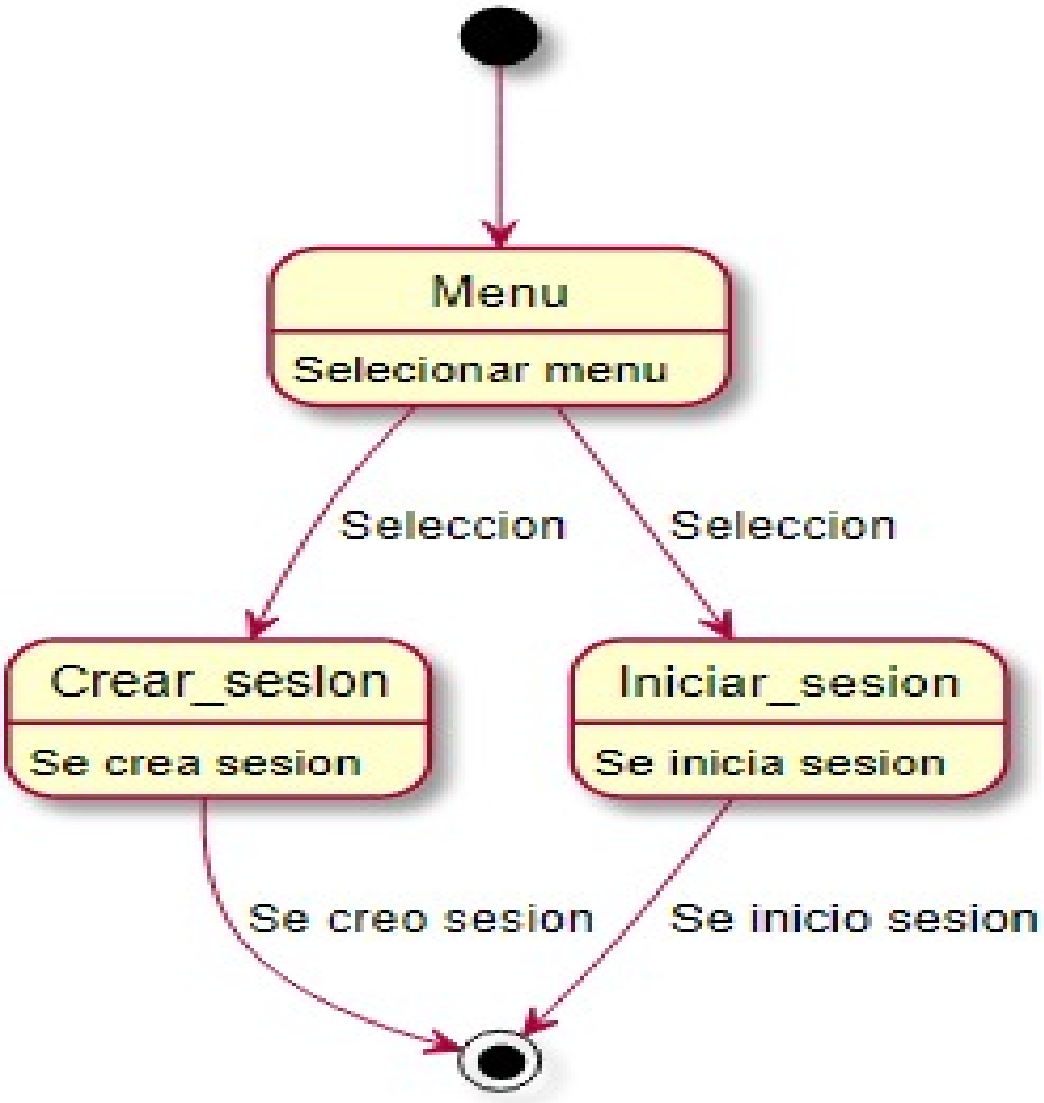


Figura 3.38: Diagrama de Estado Inicio

El diagrama cuenta con un estado inicail llamdo menu, el cual es nuestro inicio de la aplicación,

este puede irse a dos estados, los cuales son que cree una sesión o inicie sesión, al realizar uno, pasara al estado acceder al sistema.

```
@startuml
 [*] --> Menu
    Menu : Selecionar menu
    Crear_sesion :Se crea sesion
    Iniciar_sesion : Se inicia sesion


    Menu -->  Crear_sesion : Seleccion
    Menu -->  Iniciar_sesion : Seleccion
    Crear_sesion -->[*] : Se creo sesion
    Iniciar_sesion --> [*] : Se inicio sesion
@enduml
```

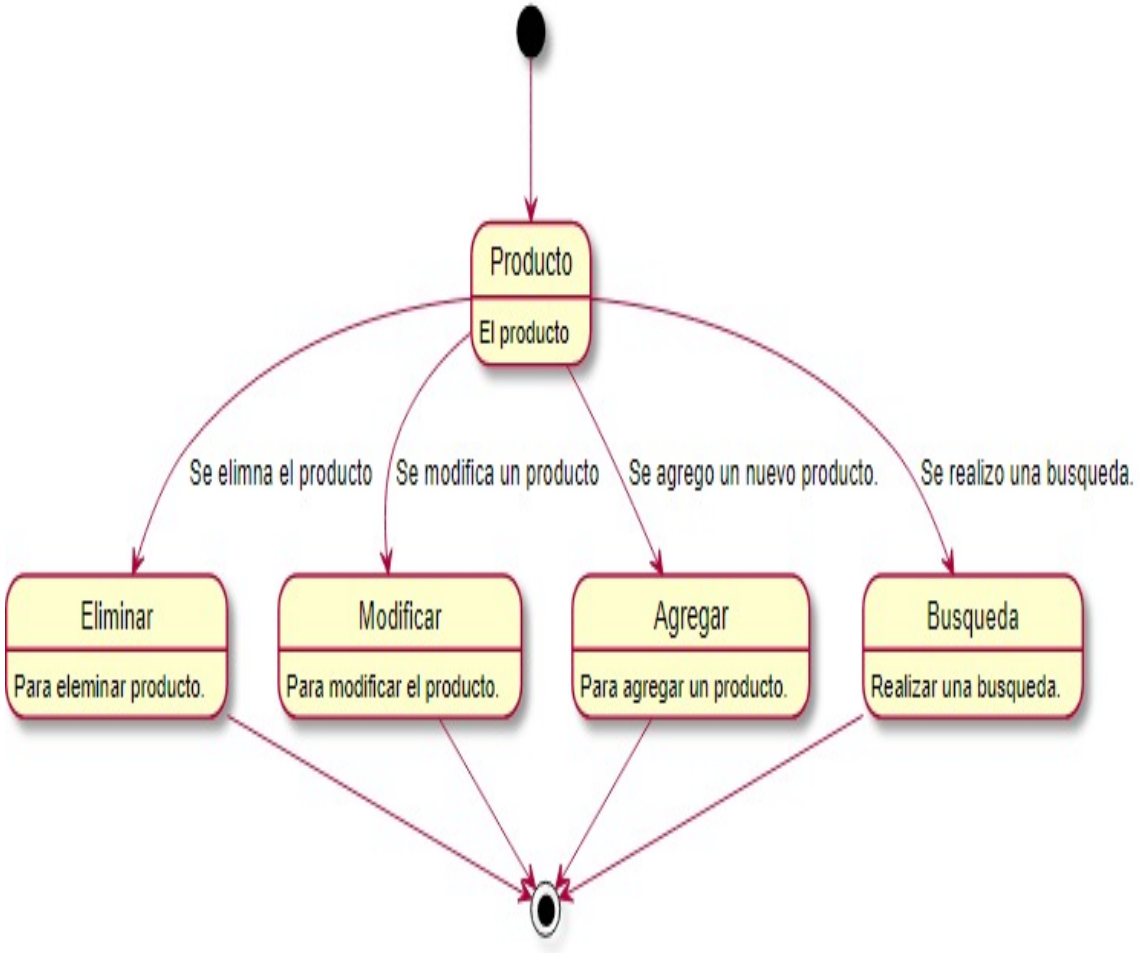


Figura 3.39: Diagrama de Estado Inventario

En este diagrama al tener un producto seleccionado, puede pasar a varios estados, los cuales son, eliminar, modificar, agregar y busqueda, el esatdo final depende de la selección.

```
@startuml
[*] --> Producto
Producto: El producto
Eliminar: Para eleminar producto.
Modificar: Para modificar el producto.
Agregar: Para agregar un producto.
Busqueda: Realizar una busqueda.
Producto --> Eliminar : Se elimna el producto
Producto --> Modificar :Se modifica un producto
Producto --> Agregar : Se agrego un nuevo producto.
Producto --> Busqueda : Se realizo una busqueda.
Eliminar --> [*]
Modificar --> [*]
Agregar --> [*]
Busqueda --> [*]
@enduml
```

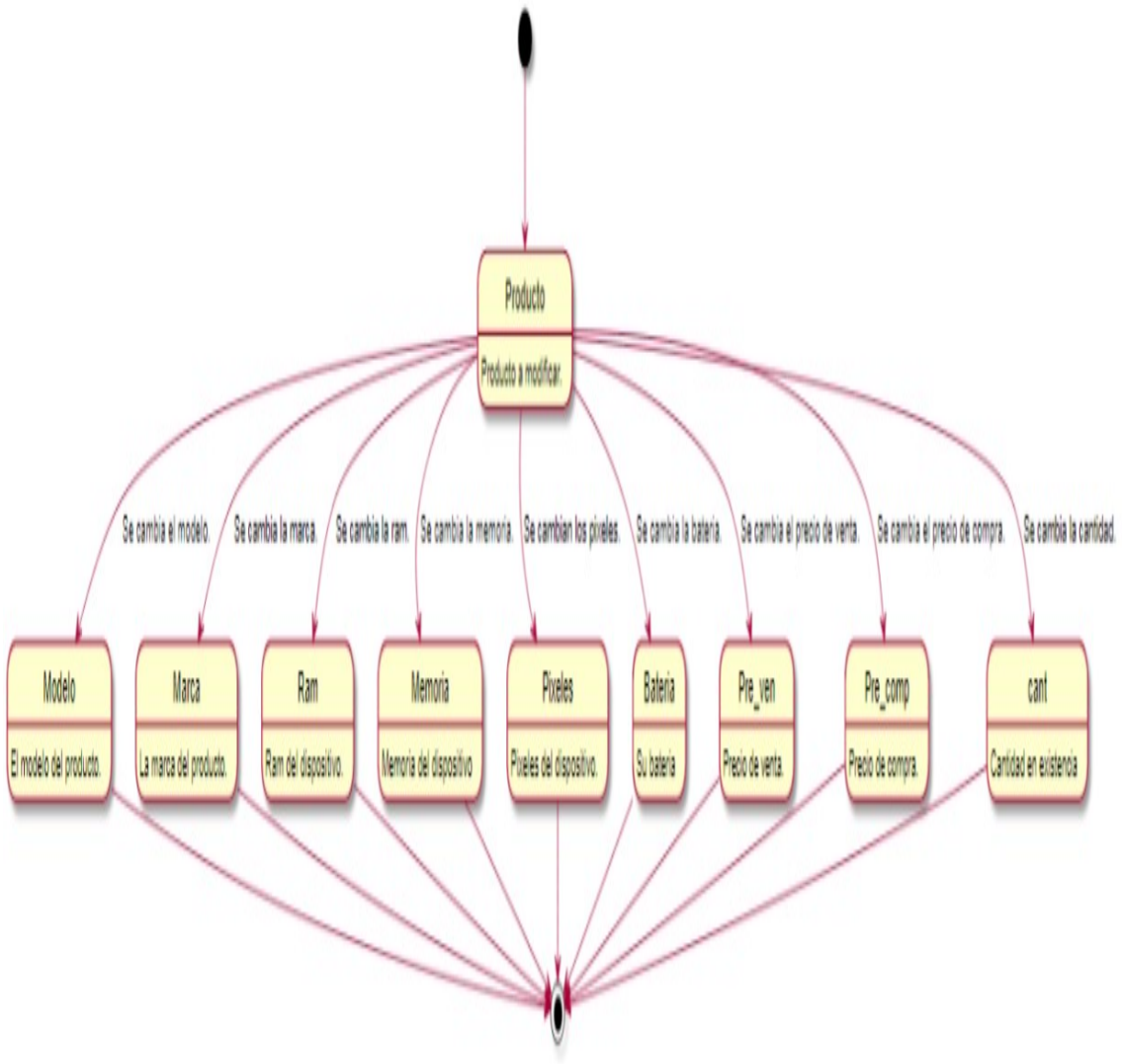


Figura 3.40: Diagrama de Estado Modificar producto

```
@startuml
[*] --> Producto

Producto :Producto a modificar.
Modelo: El modelo del producto.
Marca: La marca del producto.
Ram: Ram del dispositivo.
Memoria: Memoria del dispositivo
Pixeles: Pixeles del dispositivo.
Bateria: Su bateria
Pre_ven: Precio de venta.
```

```
Pre_comp: Precio de compra.
cant: Cantidad en existencia

Producto --> Modelo: Se cambia el modelo.
Producto --> Marca: Se cambia la marca.
Producto --> Ram: Se cambia la ram.
Producto --> Memoria:  Se cambia la memoria.
Producto --> Pixeles: Se cambian los pixeles.
Producto --> Bateria: Se cambia la bateria.
Producto --> Pre_ven: Se cambia el precio de venta.
Producto --> Pre_comp: Se cambia el precio de compra.
Producto --> cant:Se cambia la cantidad.
Modelo --> [*]
Marca --> [*]
Ram --> [*]
Memoria--> [*]
Pixeles--> [*]
Bateria--> [*]
Pre_ven--> [*]
Pre_comp--> [*]
cant--> [*]
@enduml
```

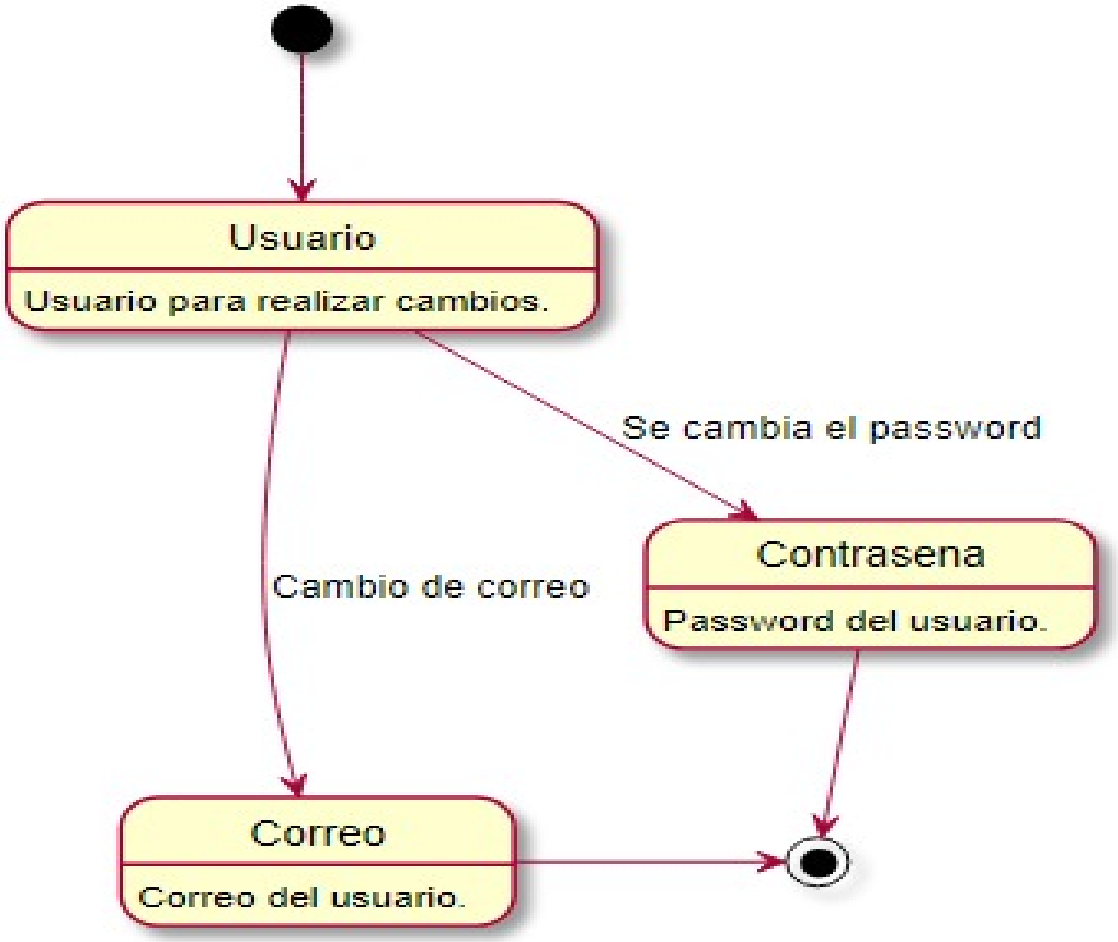


Figura 3.41: Diagrama de Estado Modificar usuario

Su estado inicial parte de el usuario, hay dos estados por consiguientes, el de cambio de contaseña y cambio de correo.

```
@startuml
[*] --> Usuario
Usuario: Usuario para realizar cambios.
Contrasena: Password del usuario.
Correo: Correo del usuario.
Usuario --> Contrasena: Se cambia el password
Usuario --> Correo: Cambio de correo
Contrasena--> [*]
Correo-> [*]
@enduml
```

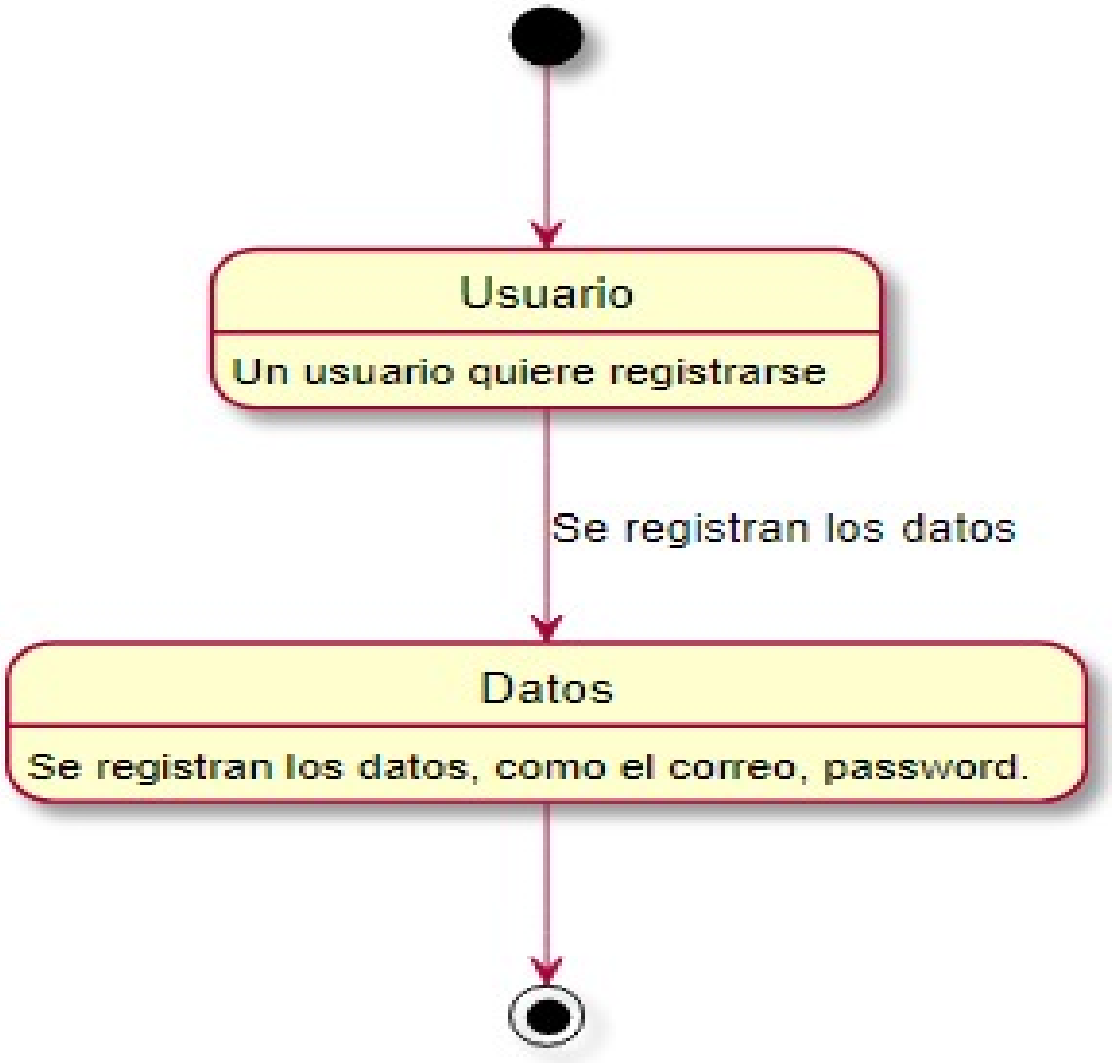


Figura 3.42: Diagrama de Estado Registrarse.

Contiene un esatdo de inicio, el cual comienza por el usuario que quiere registrarse, se obtiene sus datos y se va al estado de registrado por final.

```
@startuml
[*] --> Usuario
Usuario :Un usuario quiere registrarse
Datos : Se registran los datos, como el correo, password.
Usuario --> Datos: Se registran los datos
Datos --> [*]
@enduml
```

Figura 3.43: Diagrama de Estado Reporte diario.

Su estado inicial comienza por la parte de ventas, al realializarse una venta se genera un reporte, después de generarse pasa al esatdo de regreso a la parte de ventas.

```
@startuml
[*] --> Ventas

Consultar: Se consulta el reporte
Regresar: Se regresa a la ventana de ventas

Ventas --> Consultar :Se genera el reporte
Consultar --> Regresar: Se regresa.
Regresar --> [*]
```

```
@enduml
```

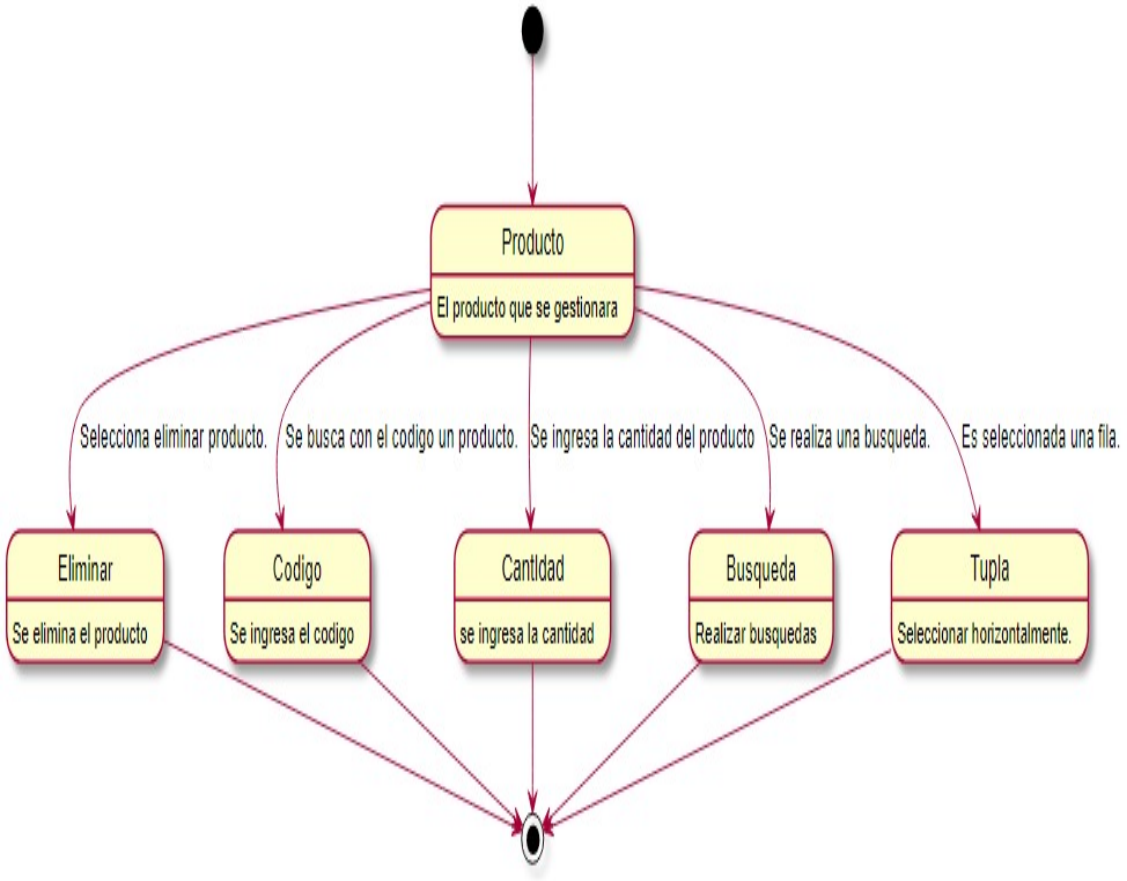


Figura 3.44: Diagrama de Estado Tabla administrador.

```
@startuml
[*] --> Producto
Producto: El producto que se gestionara
Eliminar: Se elimina el producto
Codigo: Se ingresa el codigo
Cantidad: se ingresa la cantidad
Busqueda: Realizar busquedas
Tupla: Seleccionar horizontalmente.
Producto -->Eliminar: Selecciona eliminar producto.
Producto -->Codigo: Se busca con el codigo un producto.
Producto -->Cantidad: Se ingresa la cantidad del producto
Producto -->Busqueda: Se realiza una busqueda.
Producto --> Tupla: Es seleccionada una fila.
Eliminar --> [*]
Codigo--> [*]
```

```
Cantidad--> [*]
Busqueda --> [*]
Tupla --> [*]
@enduml
```

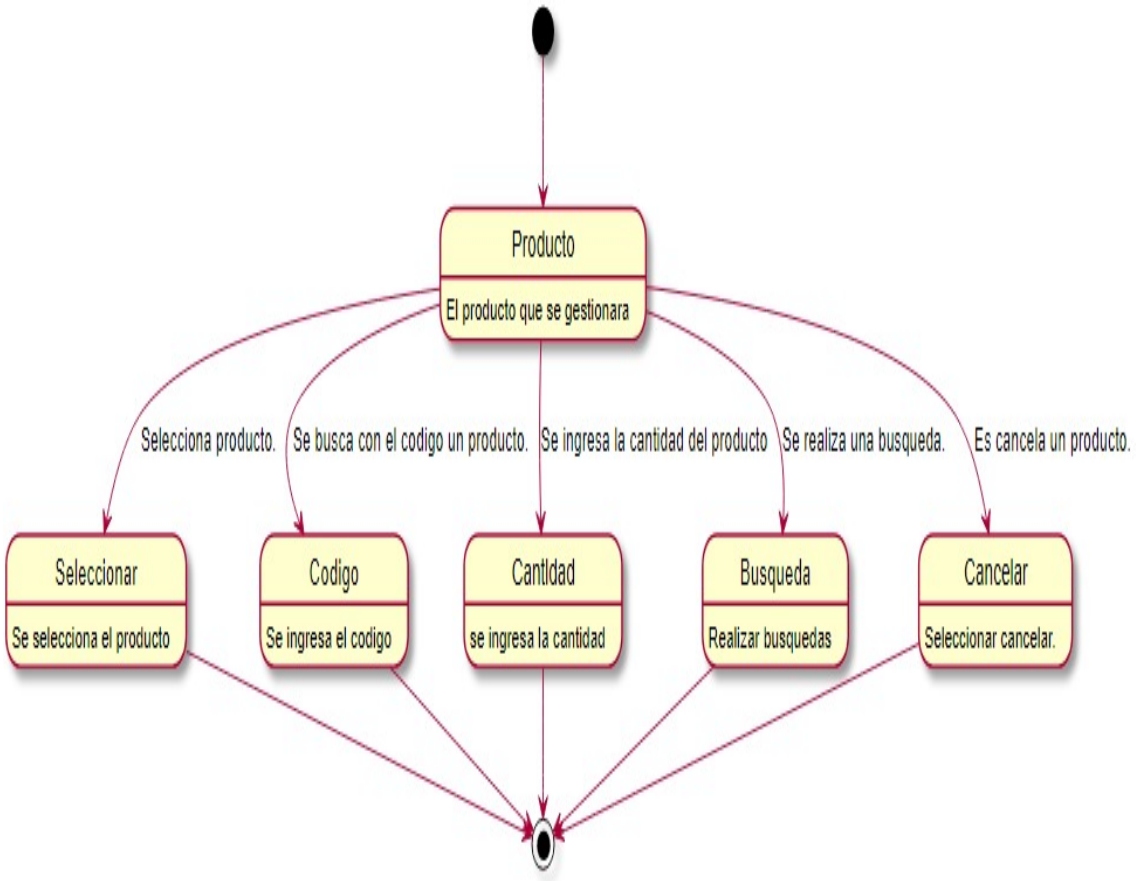


Figura 3.45: Diagrama de Estado Tabla usuario.

```
@startuml
[*] --> Producto
Producto: El producto que se gestionara
Seleccionar: Se selecciona el producto
Codigo: Se ingresa el codigo
Cantidad: se ingresa la cantidad
Busqueda: Realizar busquedas
Cancelar: Seleccionar cancelar.
Producto -->Seleccionar: Selecciona producto.
Producto -->Codigo: Se busca con el codigo un producto.
Producto -->Cantidad: Se ingresa la cantidad del producto
Producto -->Busqueda: Se realiza una busqueda.
Producto -->Cancelar : Es cancela un producto.
```

```
Seleccionar --> [*]
Codigo--> [*]
Cantidad--> [*]
Busqueda --> [*]
Cancelar --> [*]
@enduml
```

3.3.3. Diagrama de Secuencia

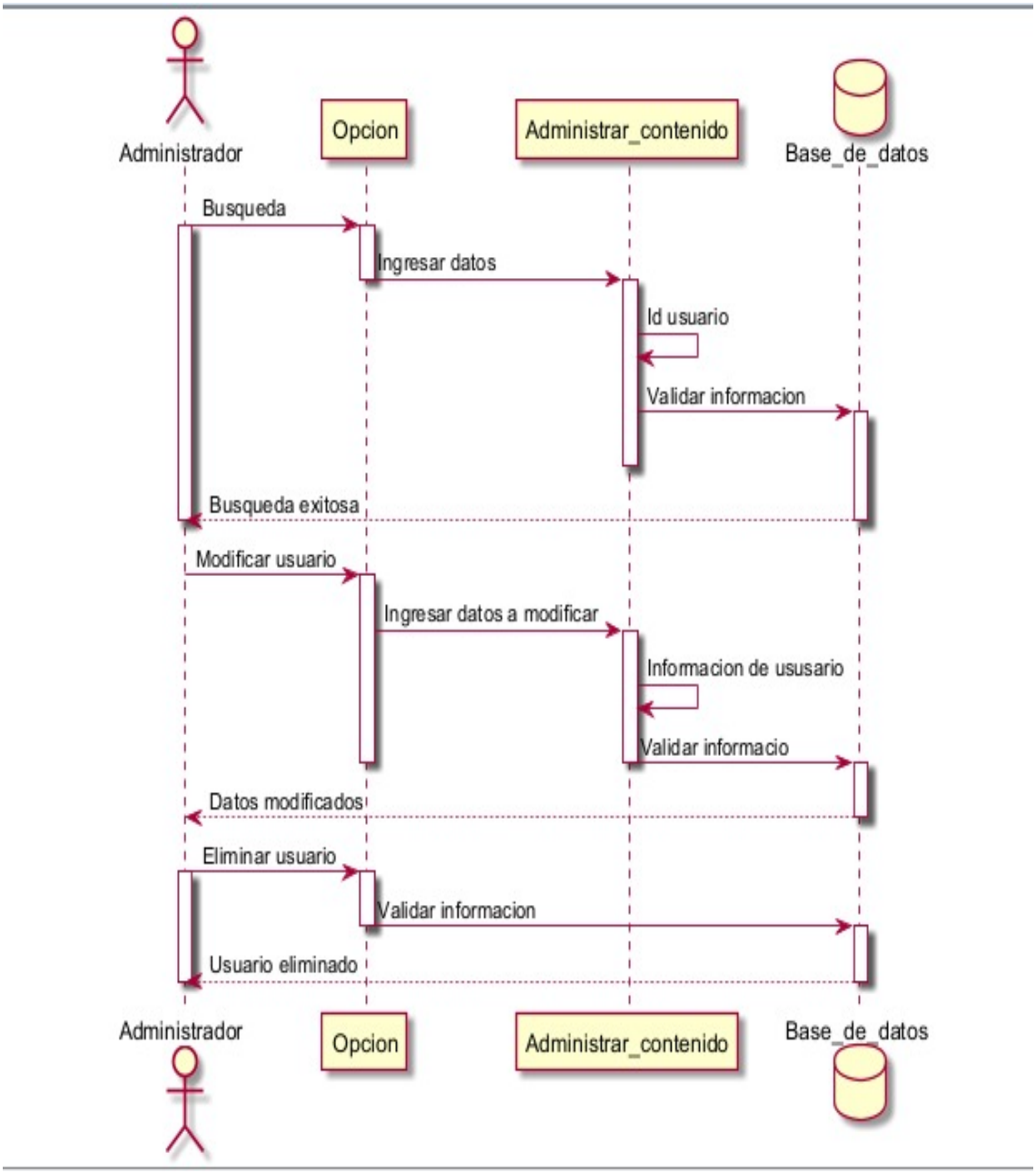


Figura 3.46: Diagrama de Secuencia Administrar usuario

En este diagrama se representa el flujo que debe seguir el administrador para poder administrar a cierto usuario. Se selecciona e insertando los datos en el orden como se muestra en el diagrama. Finalmente se validan los datos en la base de datos.

```
@startuml
Actor Administrador
Administrador->Opcion:Busqueda
Activate Opcion
Opcion->Administrar_contenido:Ingresar datos
deactivate Opcion
Activate Administrar_contenido
Administrar_contenido->Administrar_contenido:Id usuario
database Base_de_datos
Administrar_contenido->Base_de_datos:Validar informacion
Activate Base_de_datos
deactivate Administrar_contenido
 Base_de_datos-->Administrador:Busqueda exitosa
deactivate Base_de_datos


Administrador->Opcion:Modificar usuario
Activate Opcion
Opcion->Administrar_contenido:Ingresar datos a modificar
Activate Administrar_contenido
Administrar_contenido->Administrar_contenido:Informacion de ususario
Administrar_contenido->Base_de_datos:Validar informacio
deactivate Opcion
deactivate Administrar_contenido
Activate Base_de_datos
Base_de_datos--> Administrador:Datos modificados
deactivate Base_de_datos

Administrador->Opcion:Eliminar usuario
Activate Opcion
Opcion->Base_de_datos:Validar informacion
deactivate Opcion
Activate Base_de_datos

Base_de_datos-->Administrador:Usuario eliminado
deactivate Base_de_datos
@enduml
```

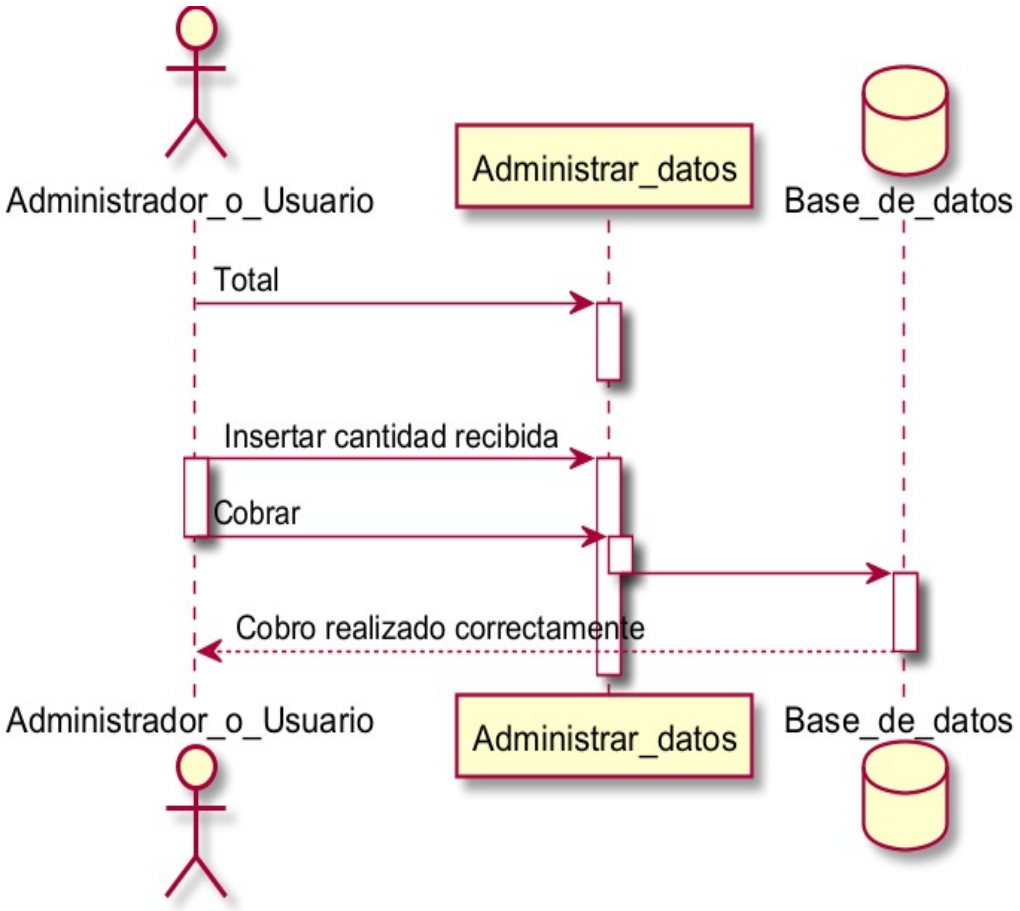


Figura 3.47: Diagrama de Secuencia Cobrar

El flujo del diagrama cobrar se puede observar claramente la forma en la que el administrador, así como el usuario pueden realizar la venta de cierto producto. Una vez que se allá recibido el dinero se realiza la venta, pero antes de eso se valida la cantidad recibida.

```
@startuml
Actor Administrador_o_Usuario
Administrador_o_Usuario-> Administrar_datos:Total
Activate Administrar_datos
deactivate Administrar_datos
Administrador_o_Usuario-> Administrar_datos: Insertar cantidad recibida
Activate Administrar_datos
deactivate Administrar_datos
Administrador_o_Usuario-> Administrar_datos: Cobrar
Activate Administrar_datos
database Base_de_datos
Administrar_datos-> Base_de_datos
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos--> Administrador_o_Usuario: Cobro realizado correctamente
```

```
deactivate Base_de_datos
@enduml
```

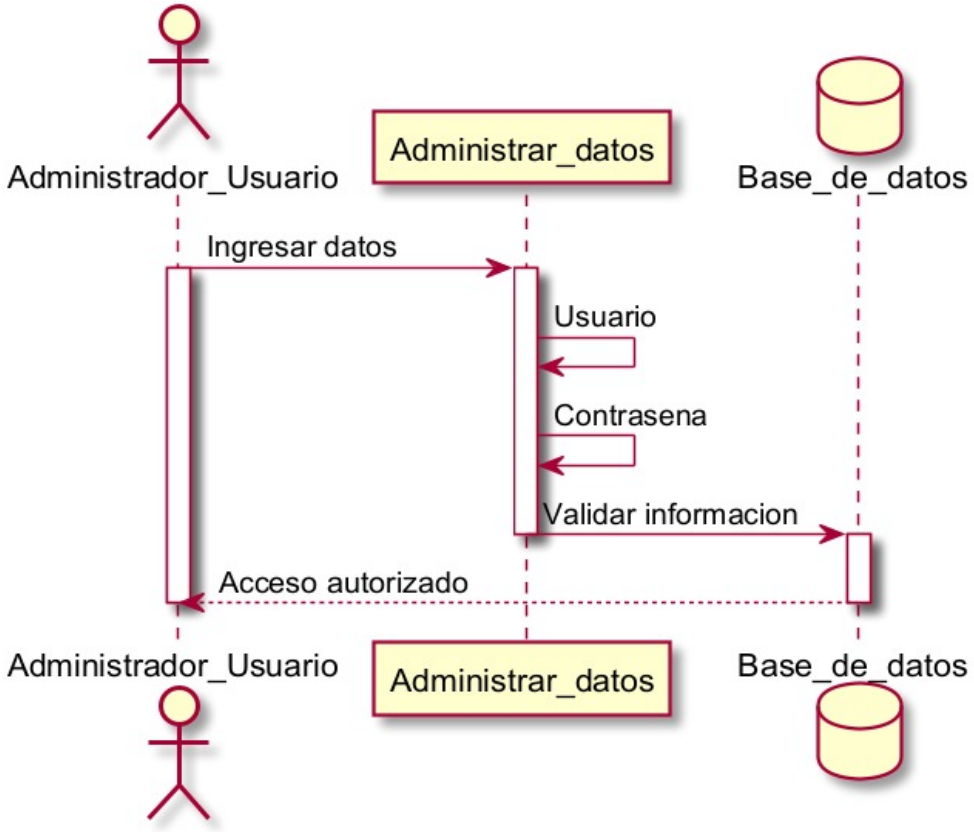


Figura 3.48: Diagrama de Secuencia Inicio de sesión

En el diagrama inicio de sesión el flujo que se sigue es el siguiente se ingrese los datos los cuales son: usuario, contraseña y final mente se ingresan los datos a la base de datos para ser validados y permitir el acceso.

```
@startuml
Actor Administrador_Usuario
 Administrador_Usuario -> Administrar_datos:Ingresar datos
Activate Administrar_datos
Administrar_datos ->Administrar_datos:Usuario
Administrar_datos ->Administrar_datos:Contrasena
database Base_de_datos
Administrar_datos -> Base_de_datos:Validar informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador_Usuario:Acceso autorizado
deactivate Base_de_datos
@enduml
```

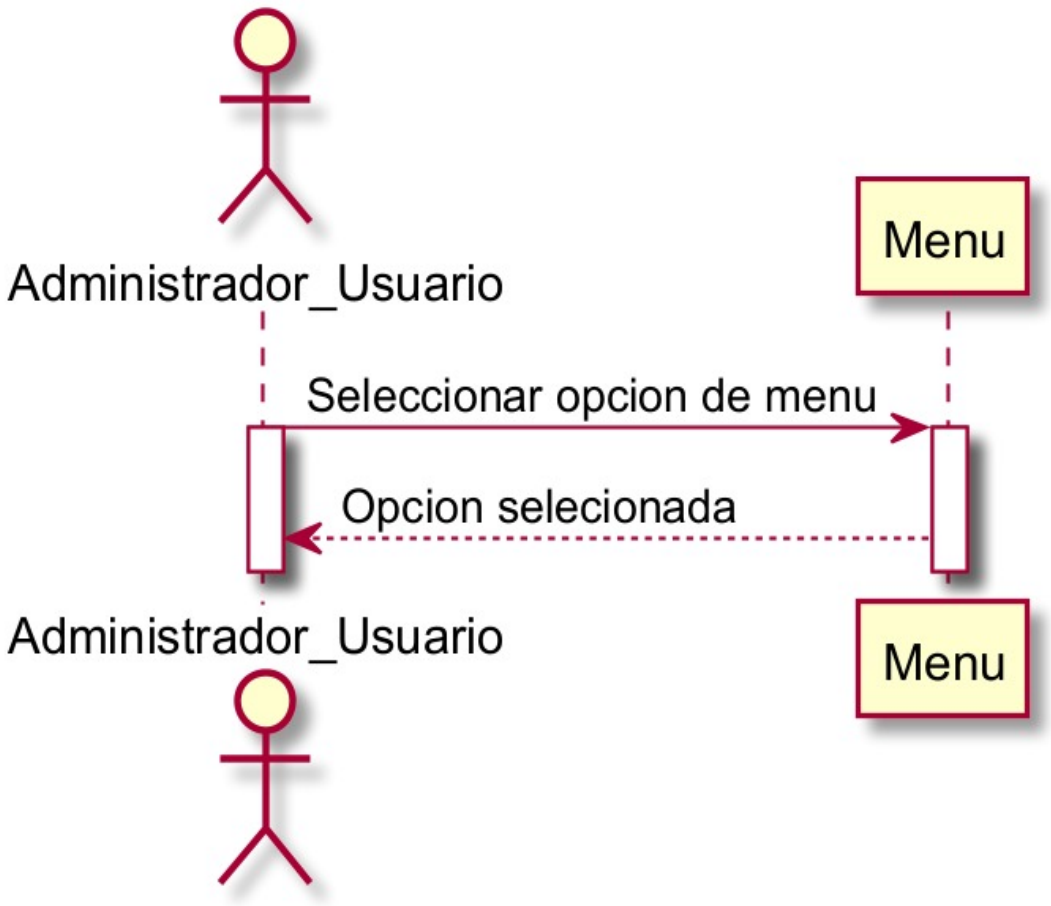


Figura 3.49: Diagrama de Secuencia Inicio

Como se puede apreciar con el diagrama de inicio, se selecciona una opción del menú, el cual despliega otra ventana.

```
@startuml
Actor Administrador_Usuario
Administrador_Usuario -> Menu: Seleccionar opcion de menu
Activate Menu
Administrador_Usuario <-- Menu: Opcion selecionada
@enduml
```

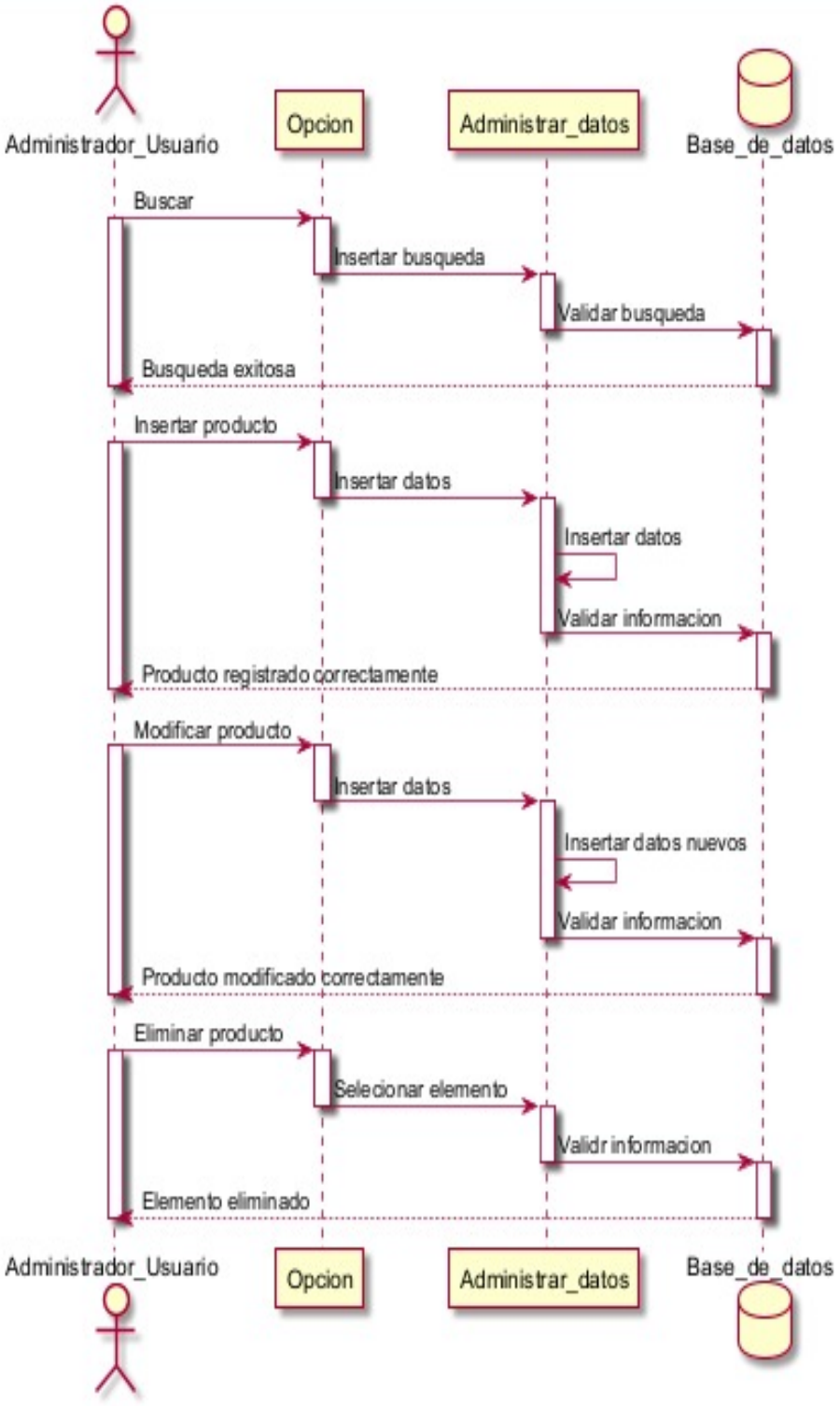


Figura 3.50: Diagrama de Secuencia Inventario

El flujo del diagrama inventario es el siguiente se selecciona una la opción de realizar búsqueda, una vez ingresado el producto a buscar se hace la búsqueda en la base de datos. Para ingresar un producto nuevo se ingresan las características del producto una vez que se realiza lo anterior se envía a la base de datos. Modificar producto con respecto a esta opción se selecciona el producto una vez seleccionado el producto se modifica sus características y finalmente se almacena en la base datos Finalmente, con la opción de eliminar producto solo se selecciona y se elimina desde la base de datos.

```
@startuml
Actor Administrador
Administrador->Opcion:Buscar
Activate Opcion
Opcion->Administrar_datos:Insertar busqueda
deactivate Opcion
Activate Administrar_datos
database Base_de_datos
Administrar_datos ->Base_de_datos:Validar busqueda
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador:Busqueda exitosa
deactivate Base_de_datos

Administrador->Opcion:Insertar producto
Activate Opcion
Opcion->Administrar_datos:Insertar datos
deactivate Opcion
Activate Administrar_datos
Administrar_datos->Administrar_datos:Insertar datos
Administrar_datos ->Base_de_datos:Validar informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador:Producto registrado correctamente
deactivate Base_de_datos

Administrador->Opcion:Modificar producto
Activate Opcion
Opcion->Administrar_datos:Insertar datos
deactivate Opcion
Activate Administrar_datos
Administrar_datos->Administrar_datos:Insertar datos nuevos
Administrar_datos ->Base_de_datos:Validar informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador:Producto modificado correctamente
deactivate Base_de_datos

Administrador->Opcion:Eliminar producto
Activate Opcion
Opcion->Administrar_datos:Selecionar elemento
deactivate Opcion
Activate Administrar_datos
```

```
Administrar_datos->Base_de_datos:Validr informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador:Elemento eliminado
deactivate Base_de_datos
@enduml
```

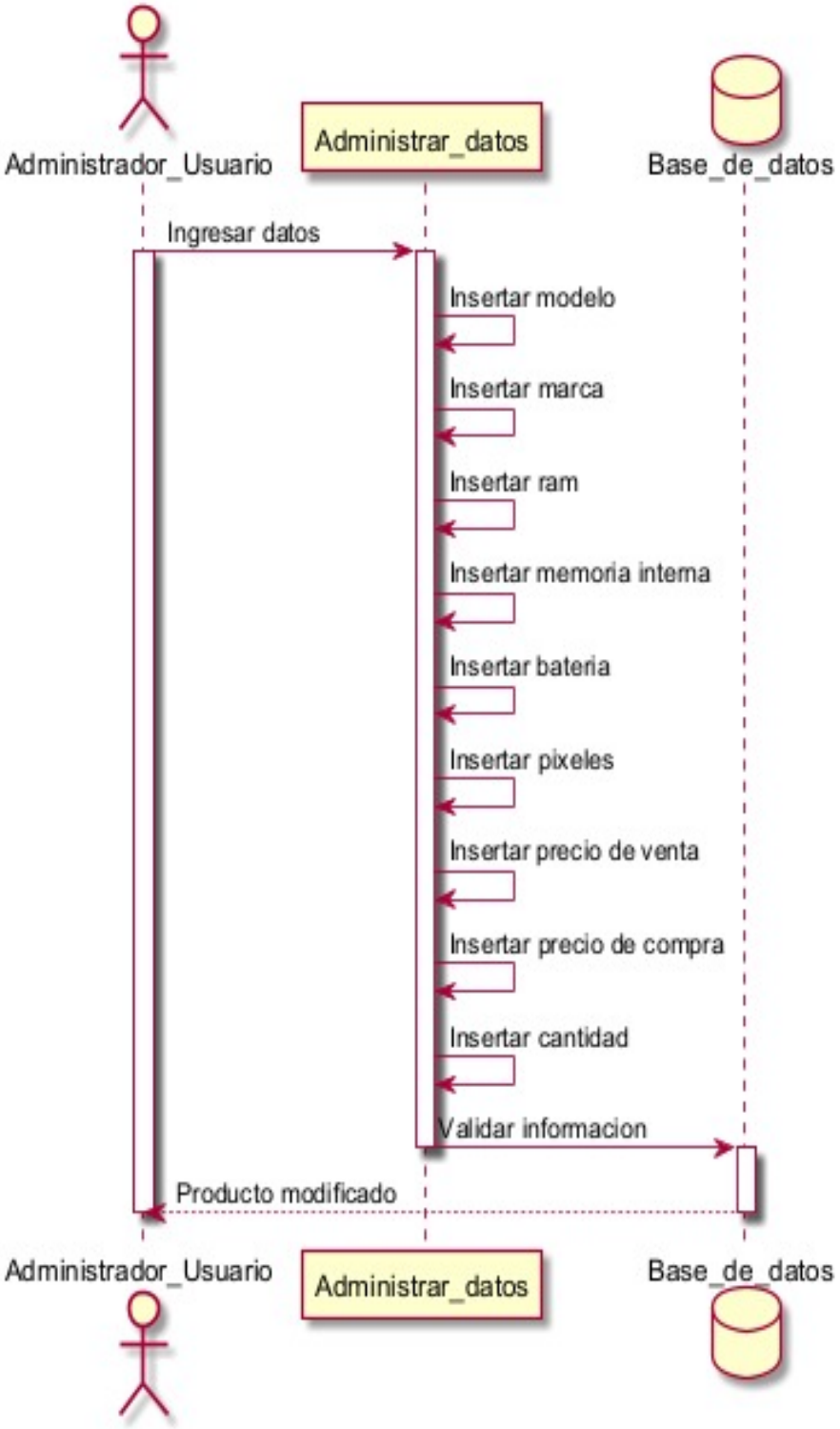


Figura 3.51: Diagrama de Secuencia Modificar producto

Se selecciona el producto a modificar posteriormente se ingresan las características nuevas del producto como punto final se validan el producto y se envía a la base de datos.

```
@startuml
Actor Administrador
Administrador->Administrar_datos:Ingresar datos
Activate Administrar_datos
Administrar_datos->Administrar_datos:Insertar modelo
Administrar_datos->Administrar_datos:Insertar marca
Administrar_datos->Administrar_datos:Insertar ram
Administrar_datos->Administrar_datos:Insertar memoria interna
Administrar_datos->Administrar_datos:Insertar bateria
Administrar_datos->Administrar_datos:Insertar pixeles
Administrar_datos->Administrar_datos:Insertar precio de venta
Administrar_datos->Administrar_datos:Insertar precio de compra
Administrar_datos->Administrar_datos:Insertar cantidad
database Base_de_datos
Administrar_datos->Base_de_datos:Validar informacion
deactivate Administrar_datos
Activate Base_de_datos
Base_de_datos-->Administrador:Producto modificado
deactivate Base_de_datos

@enduml
```

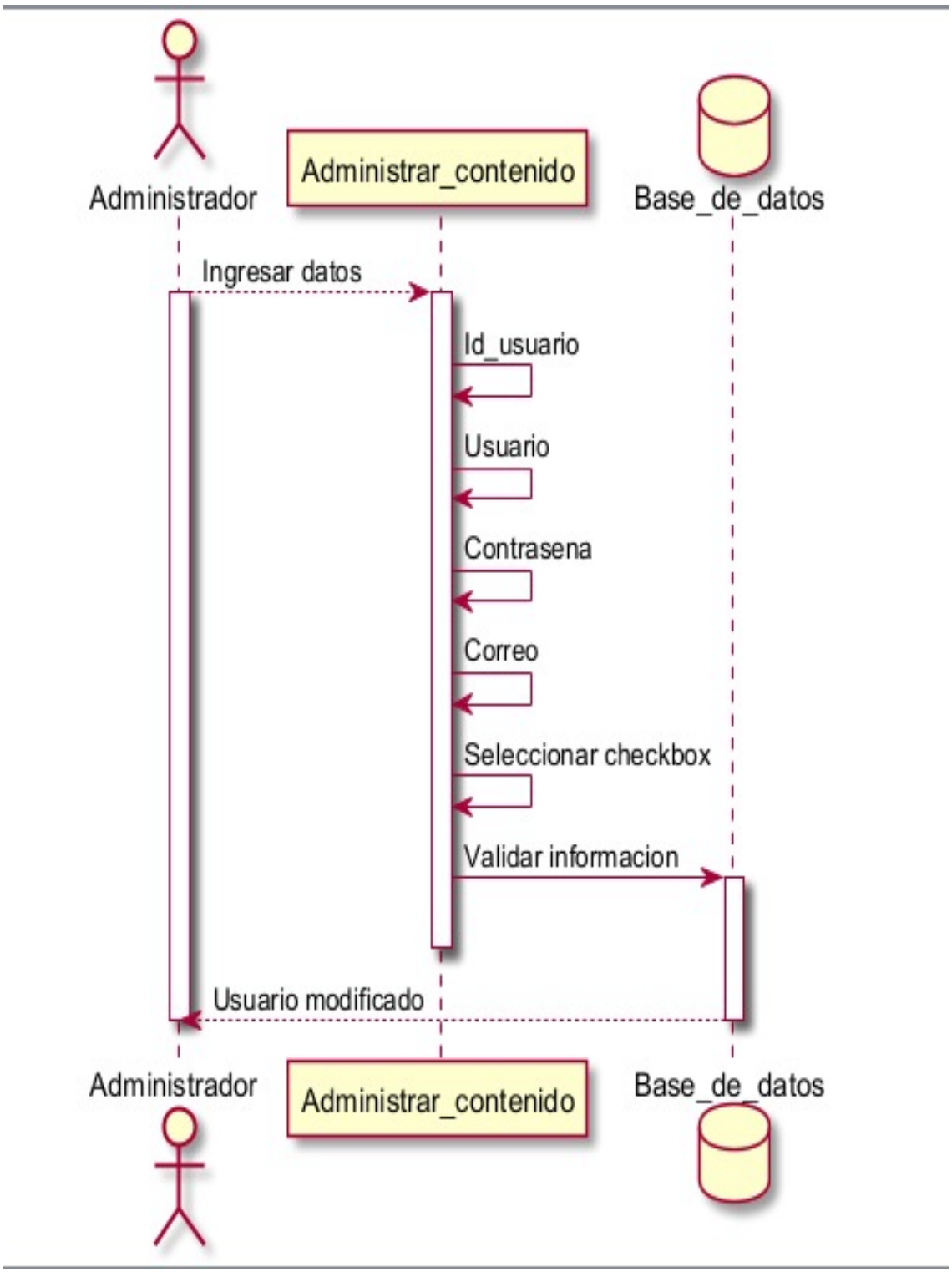


Figura 3.52: Diagrama de Secuencia Modificar usuario

Con respecto a este diagrama de modificar usuario se modifican los datos del usuario aunado a esto se seleccionan los permisos que tendrá el usuario, finalmente se envían la información a la base de datos.

```
@startuml
Actor Usuario
Usuario-->Administrar_contenido:Ingresar datos
Activate Administrar_contenido
Administrar_contenido ->Administrar_contenido:Id_usuario
Administrar_contenido ->Administrar_contenido:Usuario
Administrar_contenido ->Administrar_contenido:Contrasena
Administrar_contenido ->Administrar_contenido:Correo
Administrar_contenido ->Administrar_contenido:Seleccionar checkbox
database Base_de_datos
Administrar_contenido ->Base_de_datos:Validar informacion
Activate Base_de_datos
deactivate Administrar_contenido
Base_de_datos-->Usuario:Usuario modificado
deactivate Base_de_datos
@enduml
```

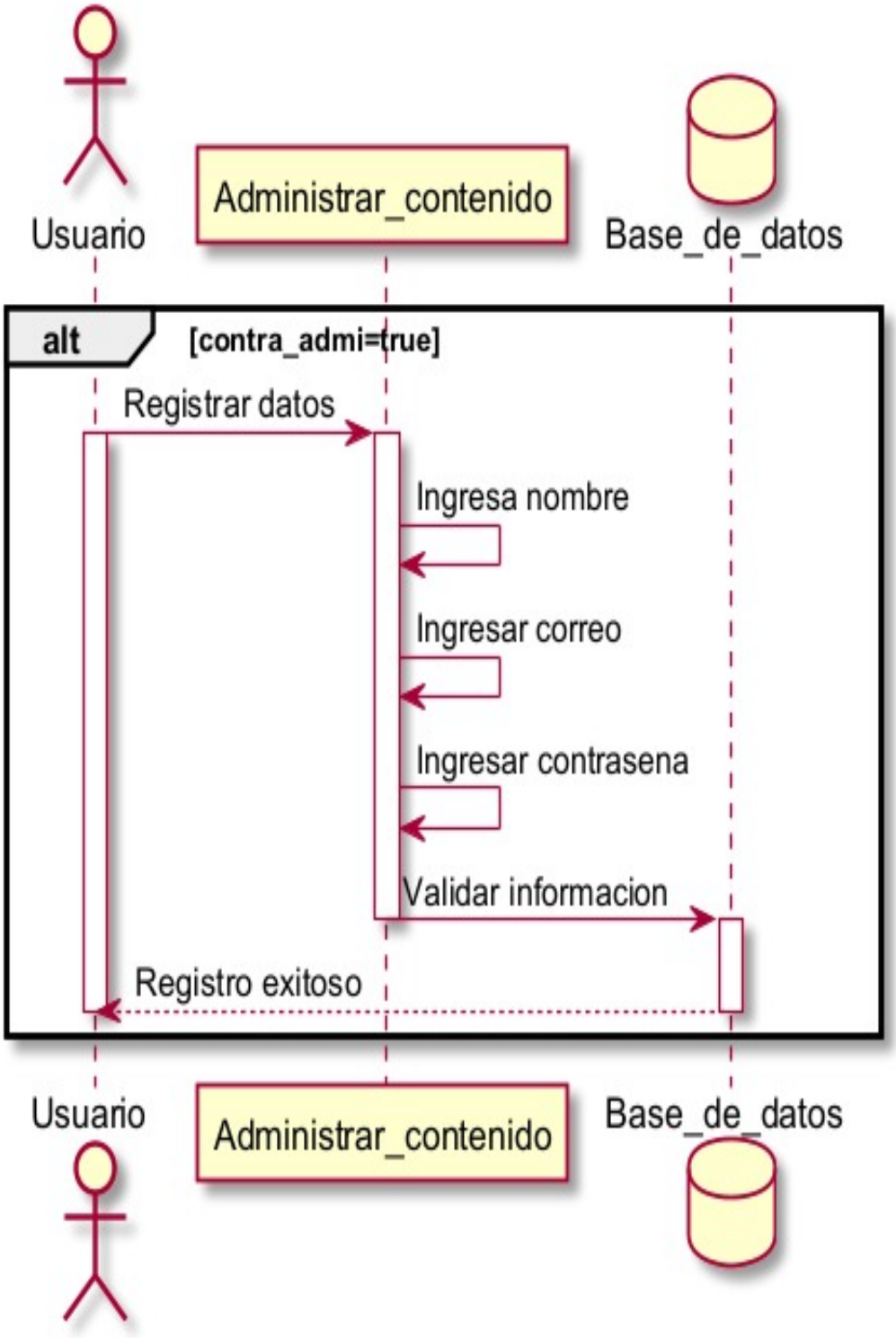


Figura 3.53: Diagrama de Secuencia Registrarse

Primero se valida el código del administrador una vez que el código sea correcto, se registra el usuario con sus respectivos datos ya introducidos.

```
@startuml
Actor Usuario
alt contra_admi=true
Usuario -> Administrar_contenido:Registrar datos
Activate Administrar_contenido
Administrar_contenido ->Administrar_contenido:Ingresa nombre
Administrar_contenido ->Administrar_contenido:Ingresar correo
Administrar_contenido ->Administrar_contenido:Ingresar contrasena
database Base_de_datos
Administrar_contenido -> Base_de_datos:Validar informacion
deactivate Administrar_contenido
Activate  Base_de_datos
Base_de_datos-->Usuario:Registro exitoso
deactivate  Base_de_datos

end
@enduml
```

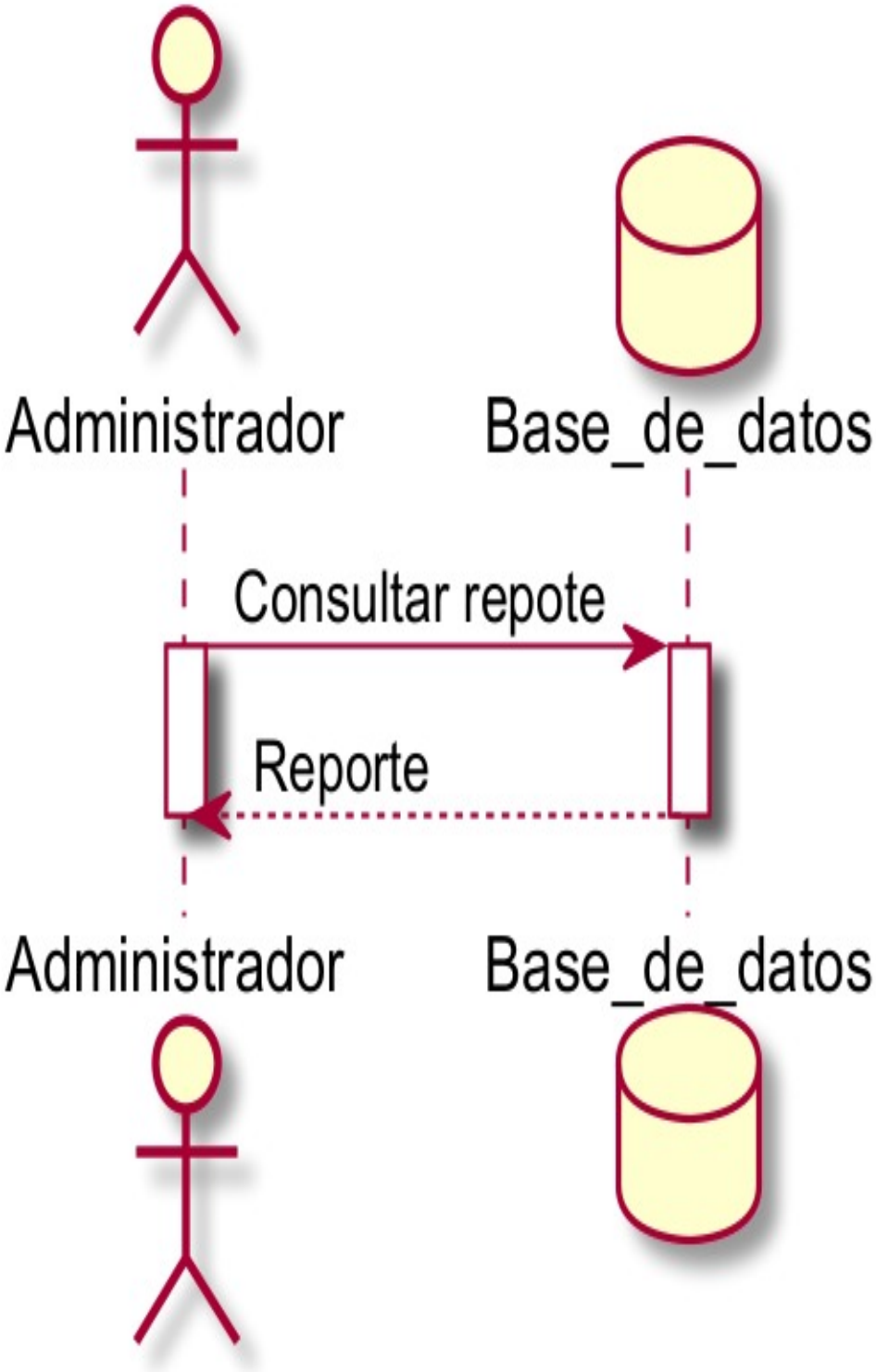


Figura 3.54: Diagrama de Secuencia Reporte

Se realiza una conexión a la base de datos para obtener el registro de los productos vendidos.

```
@startuml
Actor Administrador
database Base_de_datos
Administrador->Base_de_datos:Consultar repote
Activate Base_de_datos
Base_de_datos-->Administrador:Reporte
deactivate Base_de_datos

@enduml
```

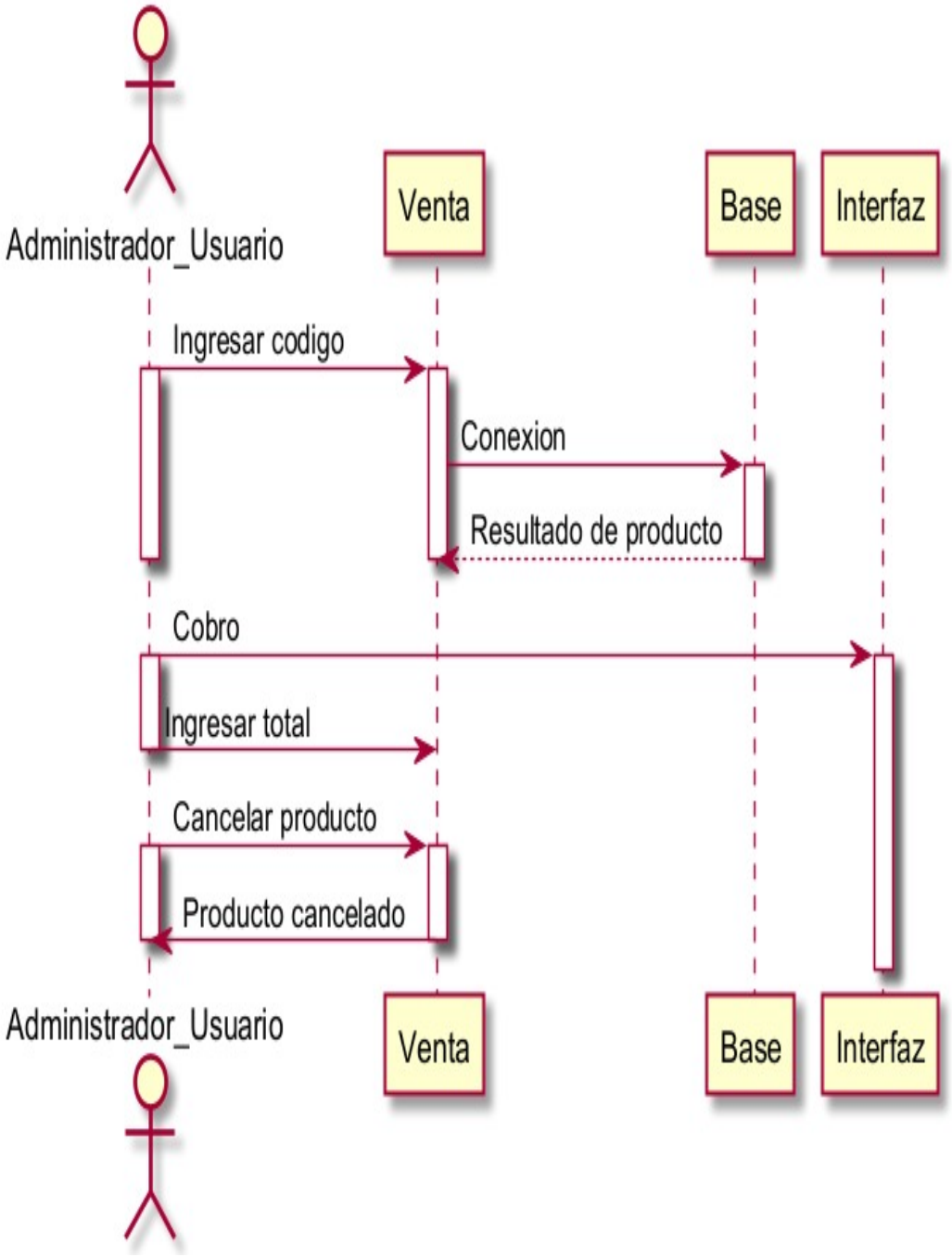


Figura 3.55: Diagrama de Secuencia Tabla administrador.

Se ingresa el código del producto posteriormente se realiza la conexión a la base de datos para obtener el resultado. En cobro se realiza la venta, pero para realizarla se despliega otra interfaz. Finalmente, en la parte de cancelar producto. Solo se cancela el producto.

```
@startuml
 Actor Administrador
 Administrador ->Opcion: Opcion
Activate Opcion
Administrador ->Opcion:Busqueda
Activate Opcion
deactivate Opcion
Administrador ->Opcion:Insertar producto
Activate Opcion
deactivate Opcion
Administrador ->Opcion:Modificar produto
Activate Opcion
deactivate Opcion
Administrador ->Opcion:Eliminar producto
Activate Opcion
deactivate Opcion

Opcion -> Administrar_datos: Insertar datos
deactivate  Opcion
Activate Administrar_datos
database Base_de_datos
Administrar_datos -> Base_de_datos:Validar datos
deactivate Administrar_datos
Activate  Base_de_datos
Base_de_datos --> Administrador: Actividad exitosa
deactivate Base_de_datos

@enduml
```

3.3.4. Diagrama de Colaboración

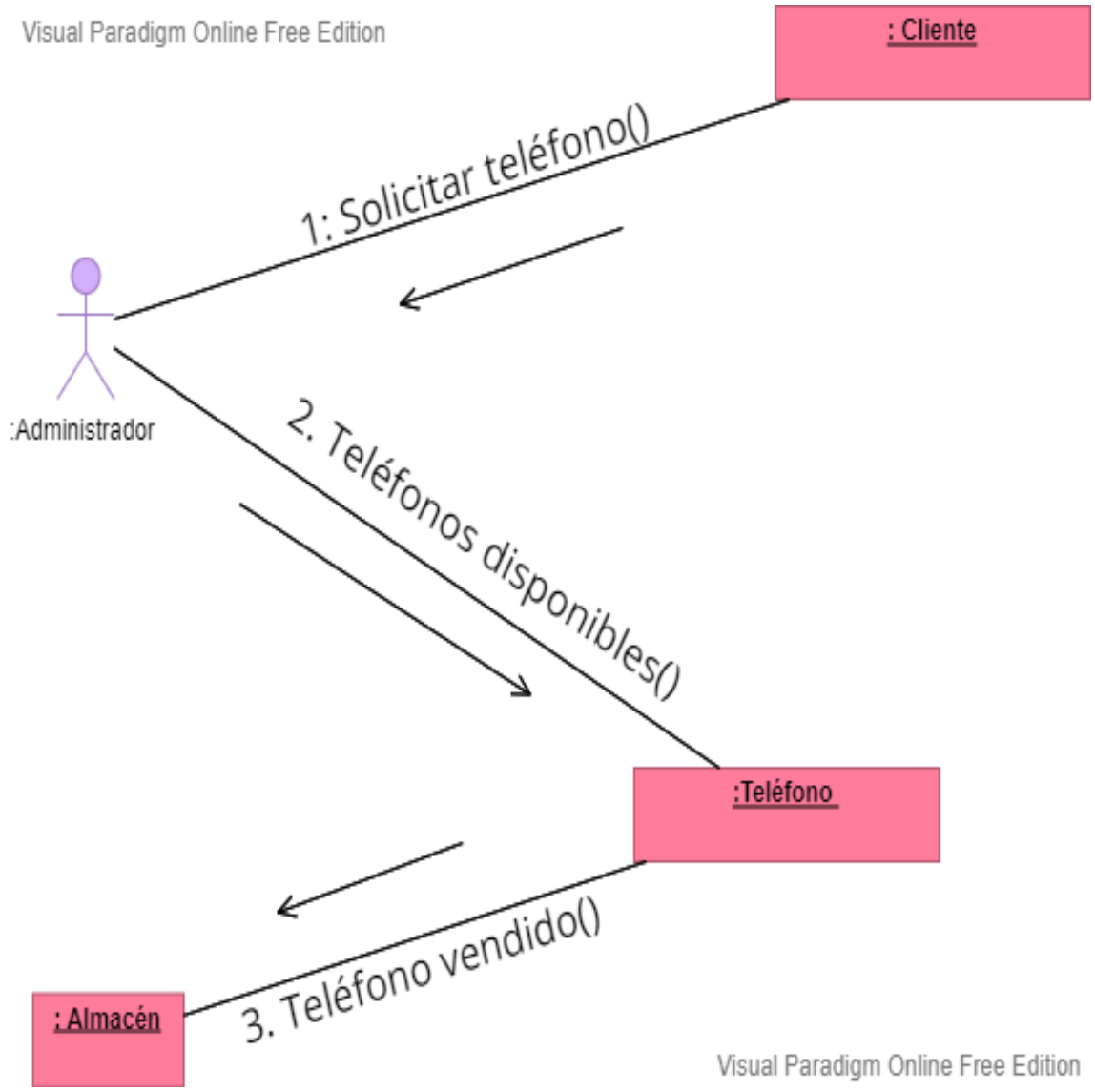


Figura 3.56: Diagrama de Colaboración Administrador.

En este diagrama el cliente solicita el teléfono, después el administrador verifica la cantidad de teléfonos disponibles y por ultimo el teléfono es vendido.

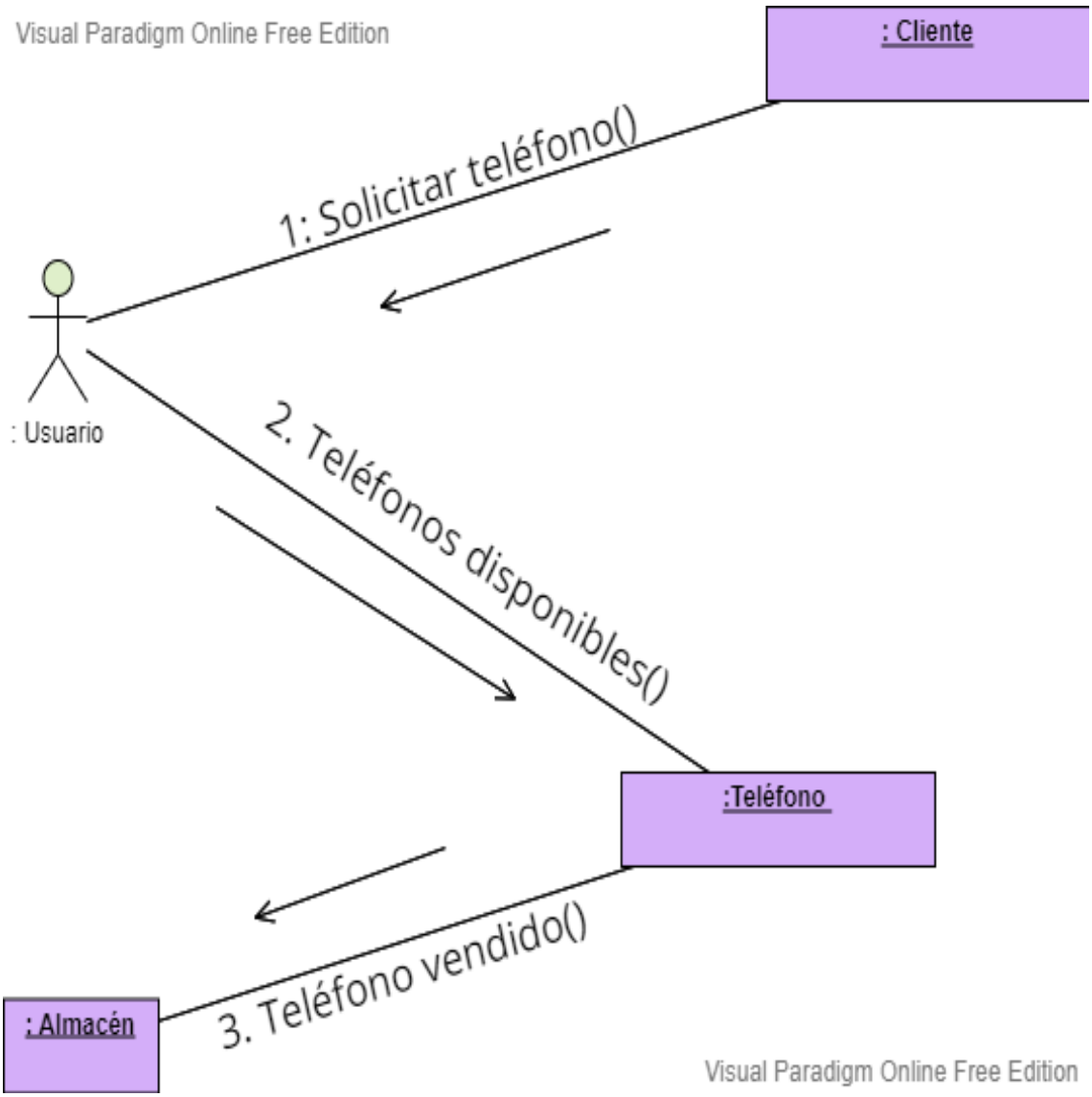


Figura 3.57: Diagrama de Colaboración Usuario.

En este diagrama el cliente solicita el teléfono, después el usuario verifica la cantidad de teléfonos disponibles y por ultimo el teléfono es vendido.

3.3.5. Diagrama de Actividades

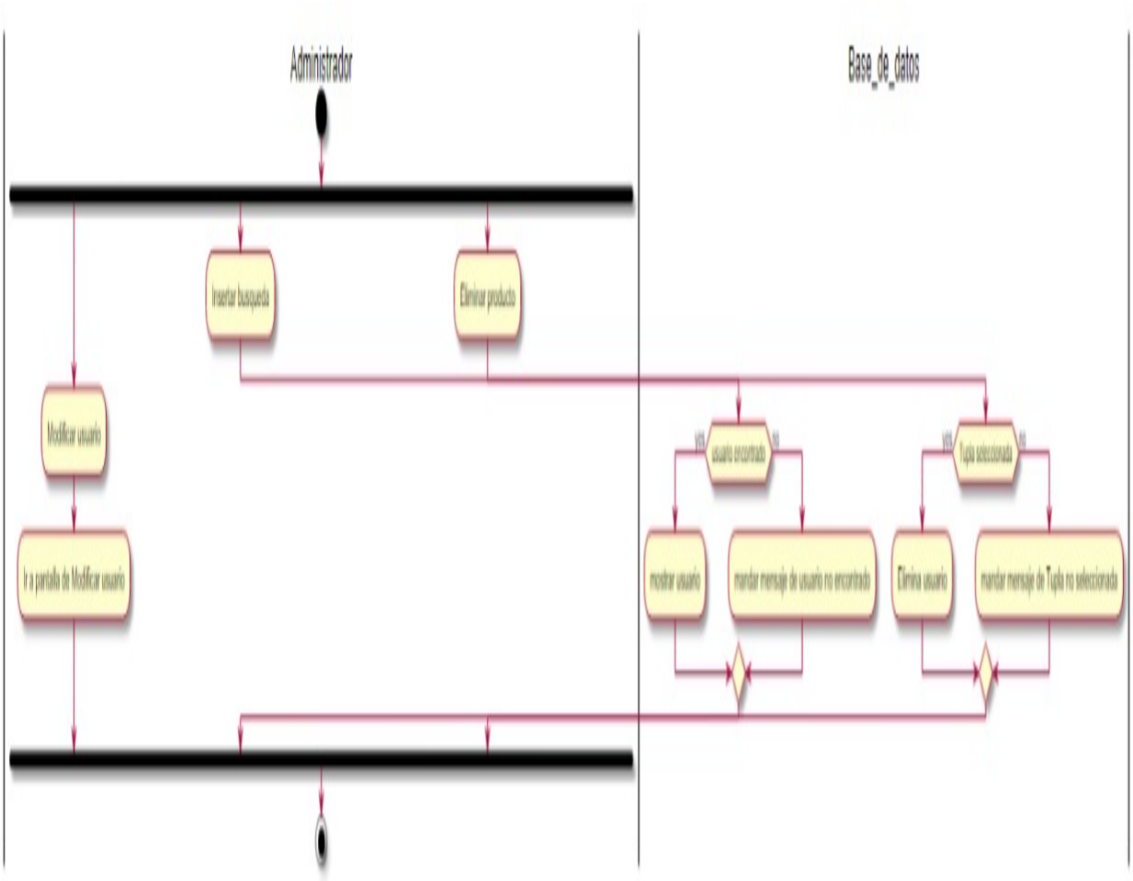


Figura 3.58: Diagrama de Actividades Administrar usuario

En este diagrama de activides se nos muestra todas la actividades que puede relizar el administrador

Codigo de Diagrama de Actividades Administrar usuario

```
    @startuml
|Administrador|
start
  fork
   :Modificar usuario;
   :Ir a pantalla de Modificar usuario;
  fork again
   :Insertar busqueda;
|Base_de_datos|
```

```
    if (usuario encontrado) then (yes)
      :mostrar usuario;
    else (no)
      :mandar mensaje de usuario no encontrado;
    endif

|Administrador|
fork again
   :Eliminar producto;
|Base_de_datos|
    if (Tupla seleccionada) then (yes)
      :Elimina usuario;
    else (no)
      :mandar mensaje de Tupla no seleccionada;
    endif
endfork
|Administrador|

stop
@enduml
```

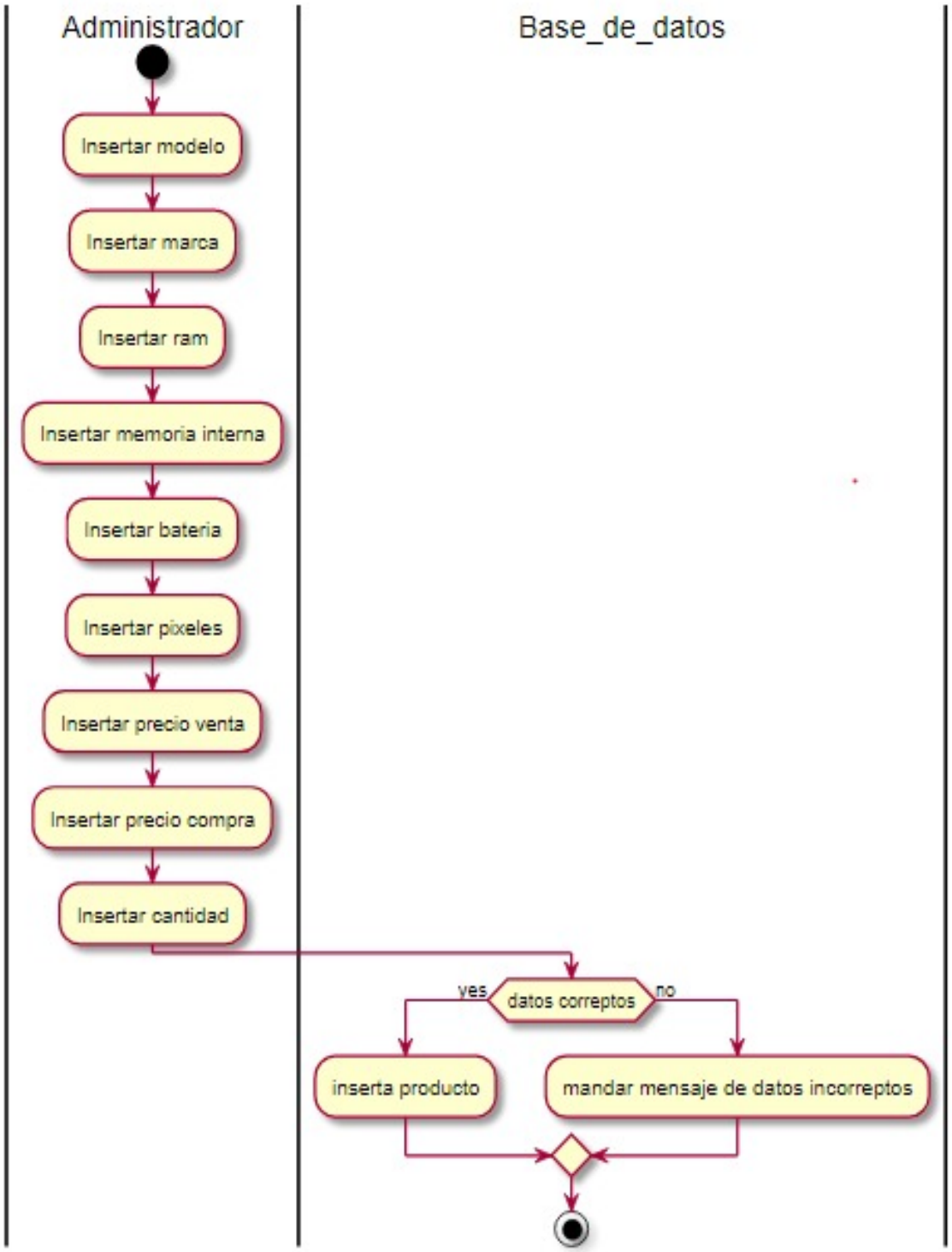


Figura 3.59: Diagrama de Actividades Agregar producto

En este diagrama se representan las actividades que se realizan cuando se agrega un producto a la base de datos, se insertan los campos de las caracteristicas de nuestro dispositivo y se guadan en la base de datos.

```
Codigo de Diagrama de Actividades Agregar producto

@startuml
|Administrador|
start

  :Insertar modelo;
  :Insertar marca;
  :Insertar ram;
  :Insertar memoria interna;
  :Insertar bateria;
  :Insertar pixeles;
  :Insertar precio venta;
  :Insertar precio compra;
  :Insertar cantidad;


|Base_de_datos|

    if (datos correptos) then (yes)
      :inserta producto;
    else (no)
      :mandar mensaje de datos incorreptos;
    endif

stop
@enduml
```

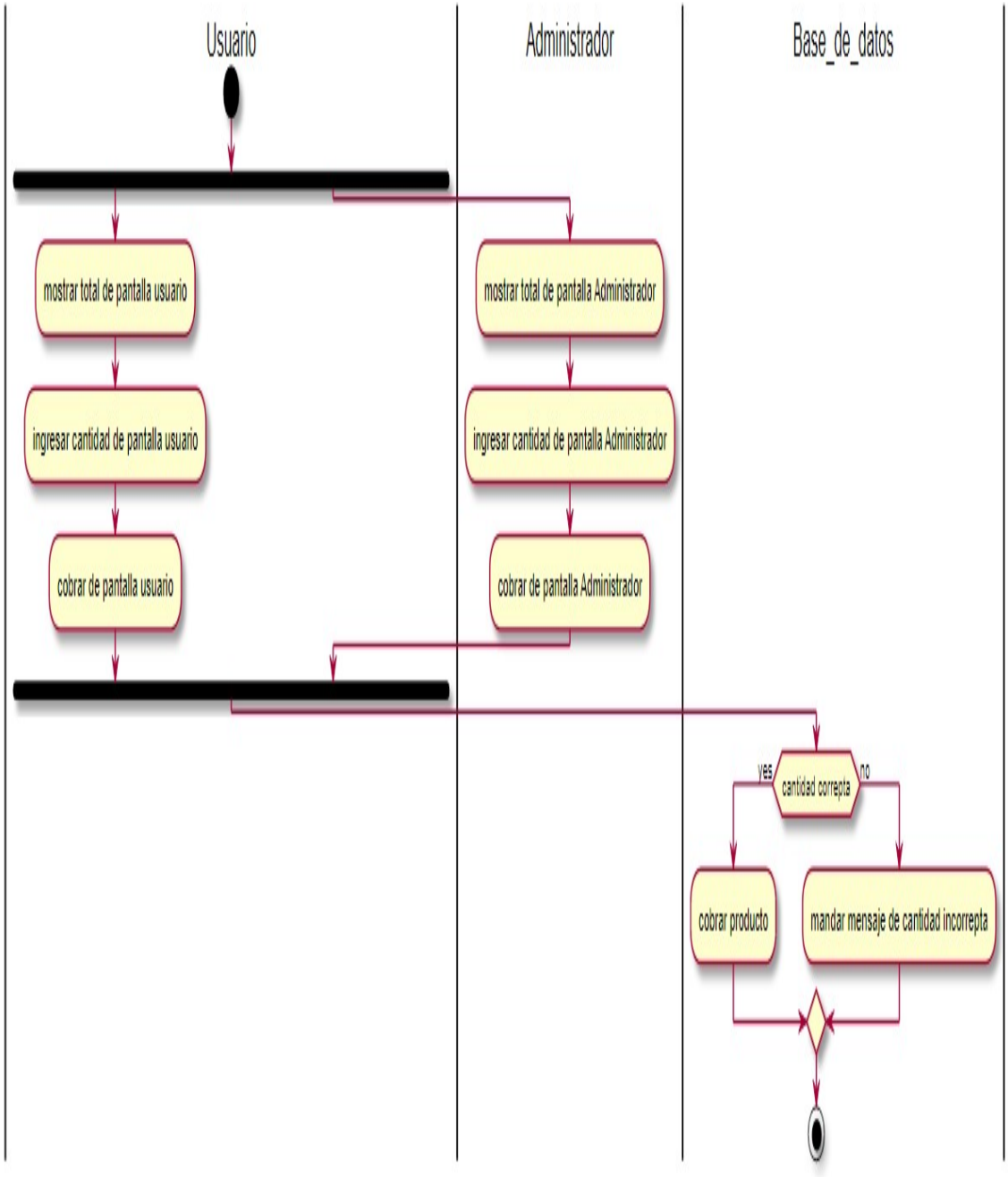


Figura 3.60: Diagrama de Actividades cobrar

```
Codigo de Diagrama de Actividades cobrar

@startuml
|Usuario|
```

```
start
  fork
   :mostrar total de pantalla usuario;
   :ingresar cantidad de pantalla usuario;
   :cobrar de pantalla usuario;
|Administrador|
  fork again
   :mostrar total de pantalla Administrador;
   :ingresar cantidad de pantalla Administrador;
   :cobrar de pantalla Administrador;
   endfork
|Base_de_datos|
    if (cantidad correpta) then (yes)
      :cobrar producto;
    else (no)
      :mandar mensaje de cantidad incorrepta;
    endif


stop
@enduml
```

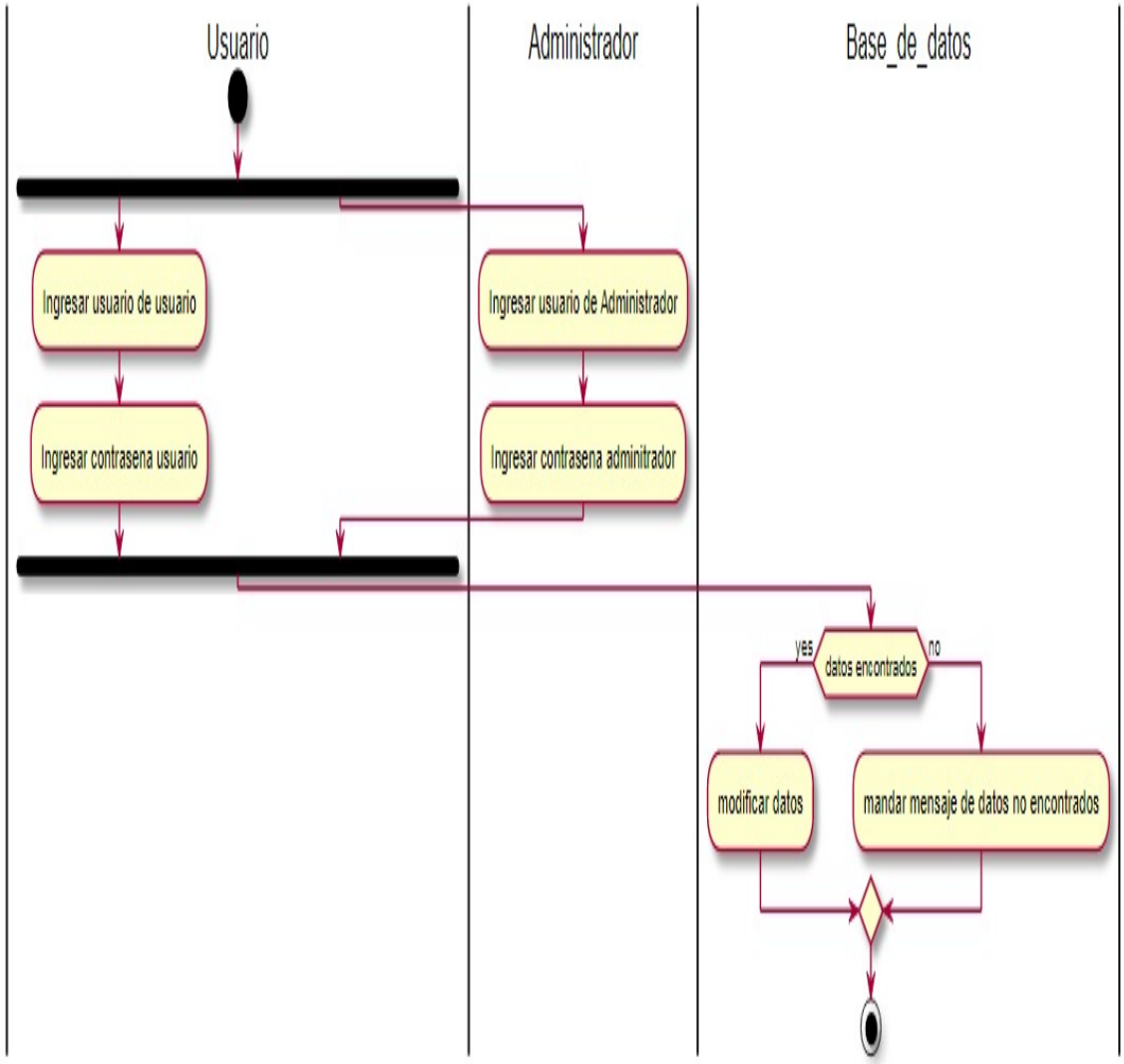


Figura 3.61: Diagrama de Actividades Inicio sesión

En este diagrama se representan las actividades que se reliza el usuario y el administrador en el logueo de nuestra aplicación, en el cual los datos insertados se validan para verificar si son correctos.

Codigo de Diagrama de Actividades Inicio sesión

```
@startuml
|Usuario|
start
  fork
   :Ingresar usuario de usuario;
   :Ingresar contrasena usuario;
|Administrador|
```

```
  fork again
   :Ingresar usuario de Administrador;
   :Ingresar contrasena adminitrador;
   endfork
|Base_de_datos|
    if (datos encontrados) then (yes)
      :modificar datos;
    else (no)
      :mandar mensaje de datos no encontrados;
    endif


stop
@enduml
```

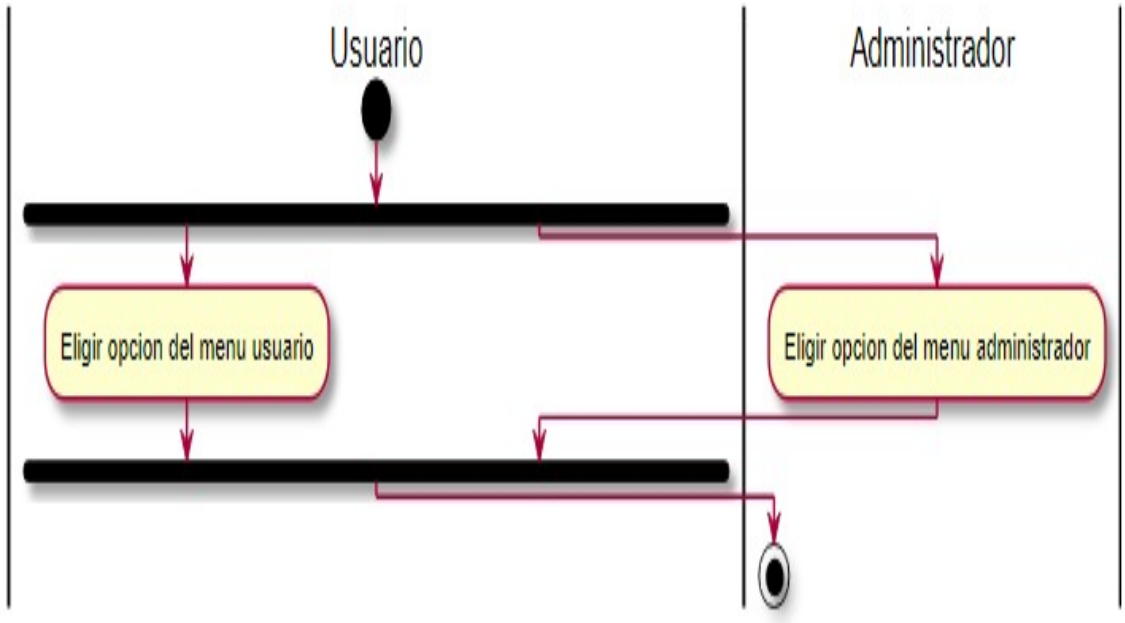


Figura 3.62: Diagrama de actividades de iniciar sesion

Codigo de Diagrama de actividades de iniciar sesion

```
@startuml
|Usuario|
start
  fork
   :Eligir opcion del menu usuario;
|Administrador|
  fork again
   :Eligir opcion del menu administrador;
```

```
   endfork

stop
@enduml
```

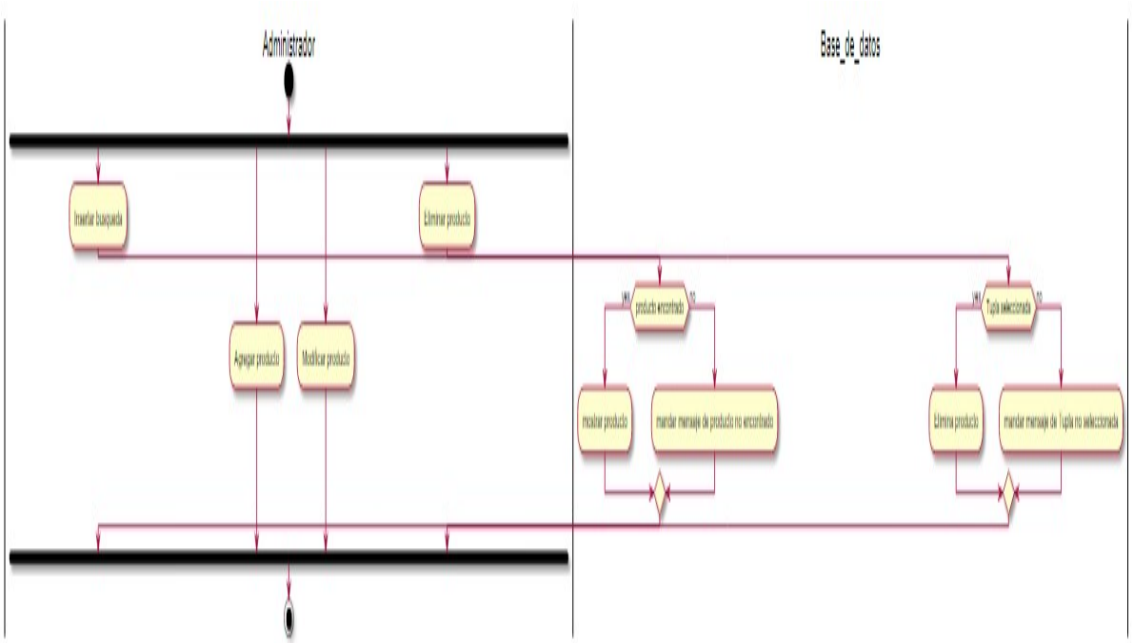


Figura 3.63: Diagrama de Actividades Inventario

Se muestran las actividades que realiza el administrador en la isersión, modificación y eliminación de productos en la base de datos.

Codigo de Diagrama de Actividades Inventario

```
@startuml
|Administrador|
start
  fork
   :Insertar busqueda;
|Base_de_datos|
    if (producto encontrado) then (yes)
      :mostrar producto;
    else (no)
      :mandar mensaje de producto no encontrado;
    endif
|Administrador|
  fork again
   :Agregar producto;
  fork again
```

```
   :Modificar producto;
  fork again
   :Eliminar producto;

|Base_de_datos|
    if (Tupla seleccionada) then (yes)
      :Elimina producto;
    else (no)
      :mandar mensaje de Tupla no seleccionada;
    endif
endfork
|Administrador|


stop
@enduml
```

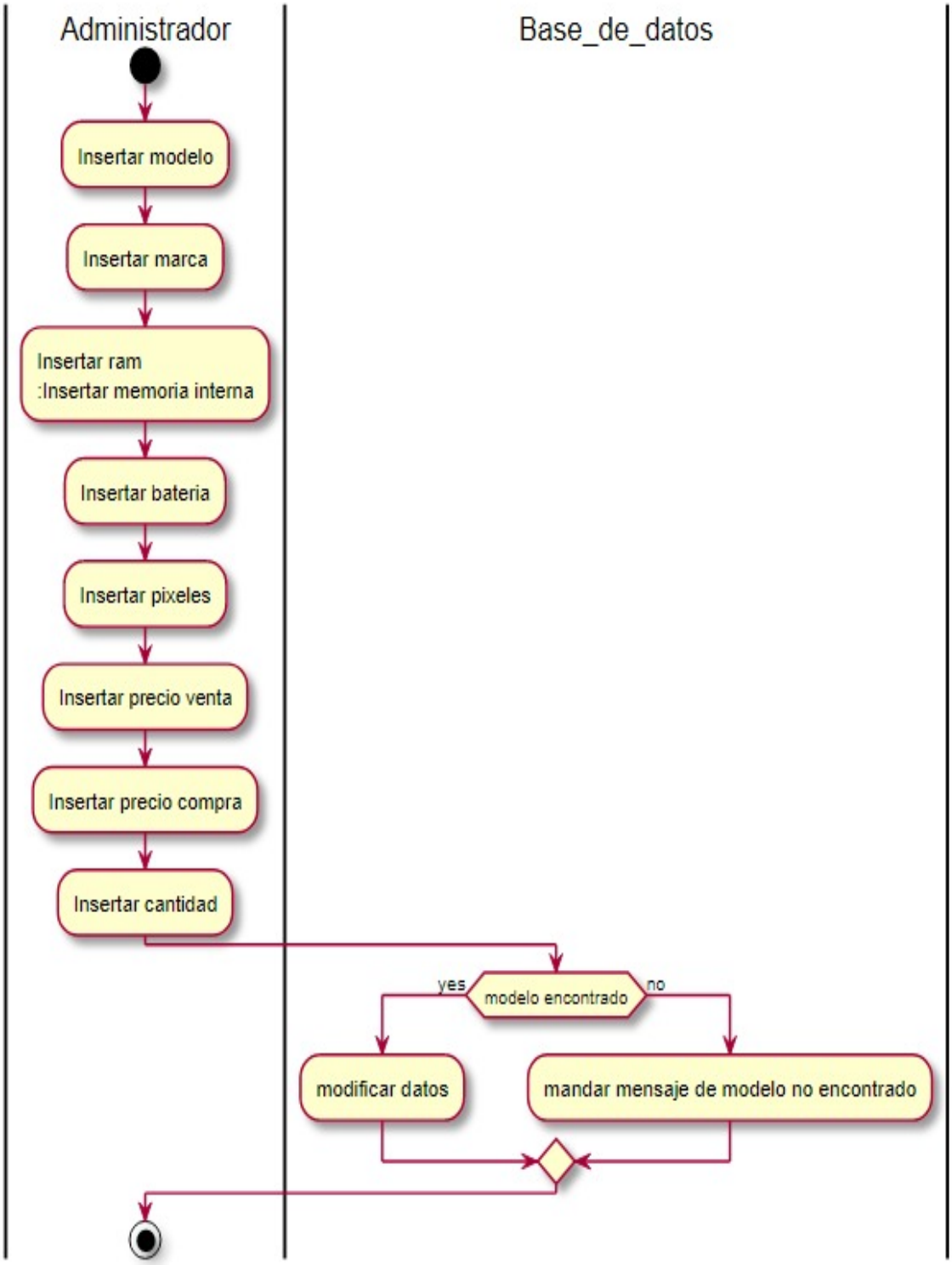


Figura 3.64: Diagrama de Actividades Modoficar producto

El diagrama de modificar producto nos muestra las actividades que realizamos cuando modificamos un producto, en el cual si el modelo que deaseamos modificar en nuestra base de datos existe, se modifica, si no es encontrado se envía un mensaje de que no existe.

```
Codigo de Diagrama de Actividades Modoficar producto

@startuml
|Administrador|
start
  :Insertar modelo;
  :Insertar marca;
  :Insertar ram
  :Insertar memoria interna;
  :Insertar bateria;
  :Insertar pixeles;
  :Insertar precio venta;
  :Insertar precio compra;
  :Insertar cantidad;

|Base_de_datos|
    if (modelo encontrado) then (yes)
      :modificar datos;
    else (no)
      :mandar mensaje de modelo no encontrado;
    endif
|Administrador|


stop
@enduml
```

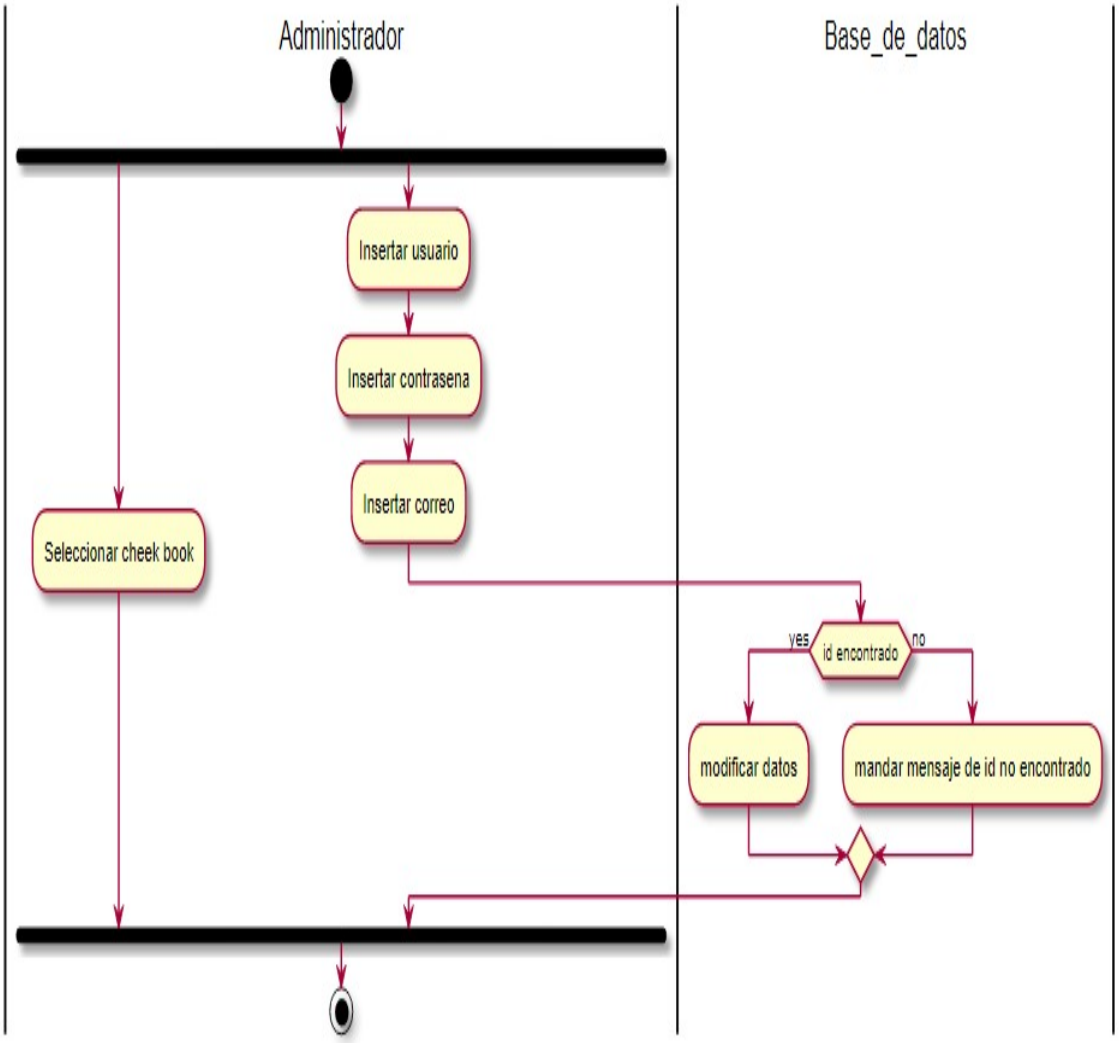


Figura 3.65: Diagrama de actividades de modificar usuario

En este diagrama se representan las actividades que se realizan en nuestra pantalla inicial, en la cual se elige la opción deseada.

```
Codigo de Diagrama de actividades de modificar usuario
    @startuml
|Administrador|
start
  fork
 :Seleccionar cheek book;
  fork again
  :Insertar usuario;
  :Insertar contrasena;
```

```
  :Insertar correo;

|Base_de_datos|
    if (id encontrado) then (yes)
      :modificar datos;
    else (no)
      :mandar mensaje de id no encontrado;
    endif
    endfork
|Administrador|


stop
@enduml
```

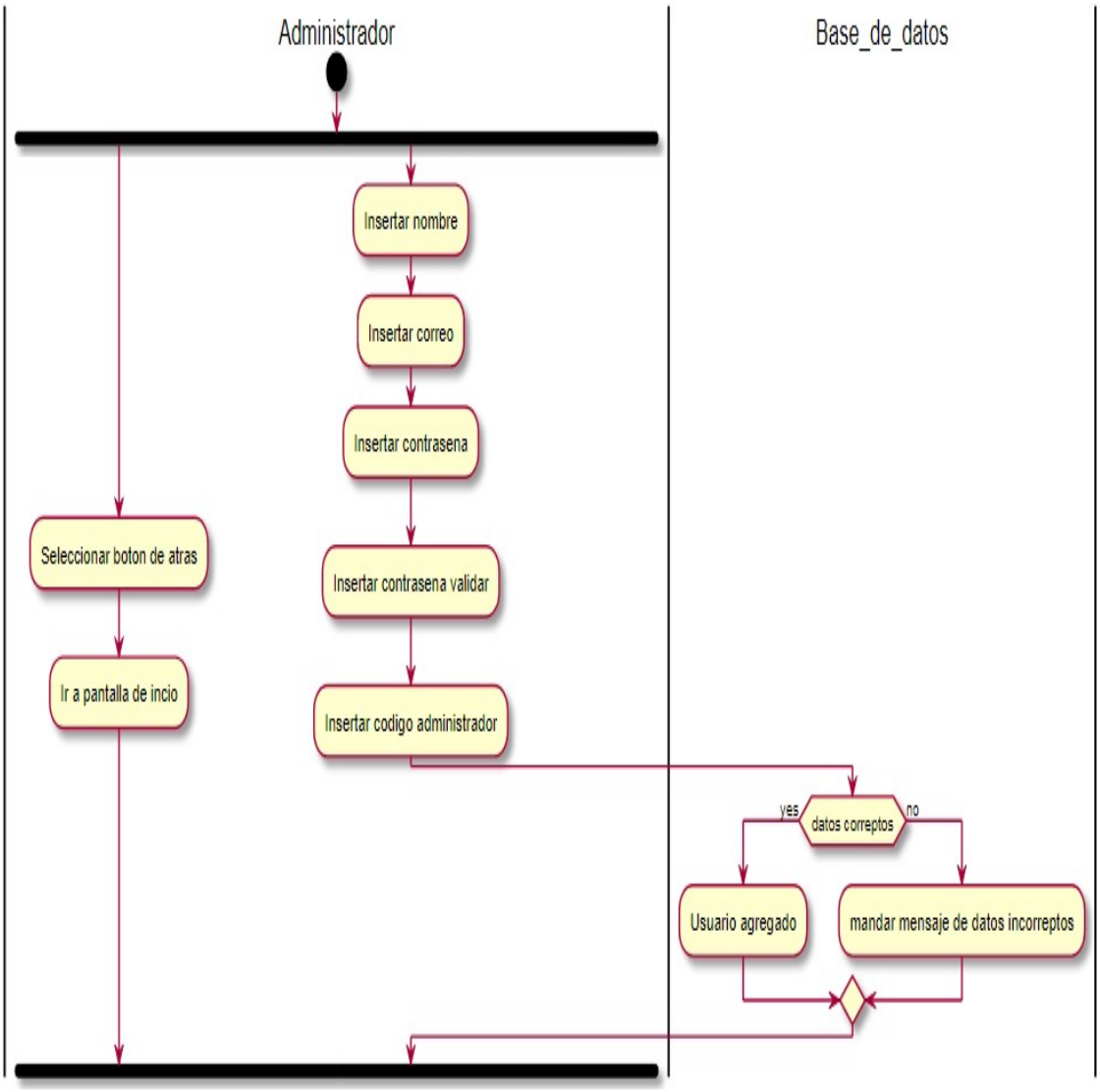


Figura 3.66: Diagrama de Actividades Registro

Este diagrama nos muestra las actividades que podemos realizar en nuestra pantalla de registro, la cual si se valida que los datos son correctos se inserta el nuevo producto.

Codigo de Diagrama de Actividades Registro

```
@startuml
|Administrador|
start
  fork
 :Seleccionar boton de atras;
 :Ir a pantalla de incio;
  fork again
```

```
  :Insertar nombre;
  :Insertar correo;
  :Insertar contrasena;
  :Insertar contrasena validar;
  :Insertar codigo administrador;

|Base_de_datos|
    if (datos correptos) then (yes)
      :Usuario agregado;
    else (no)
      :mandar mensaje de datos incorreptos;
    endif
    endfork
|Administrador|


stop
@enduml
```

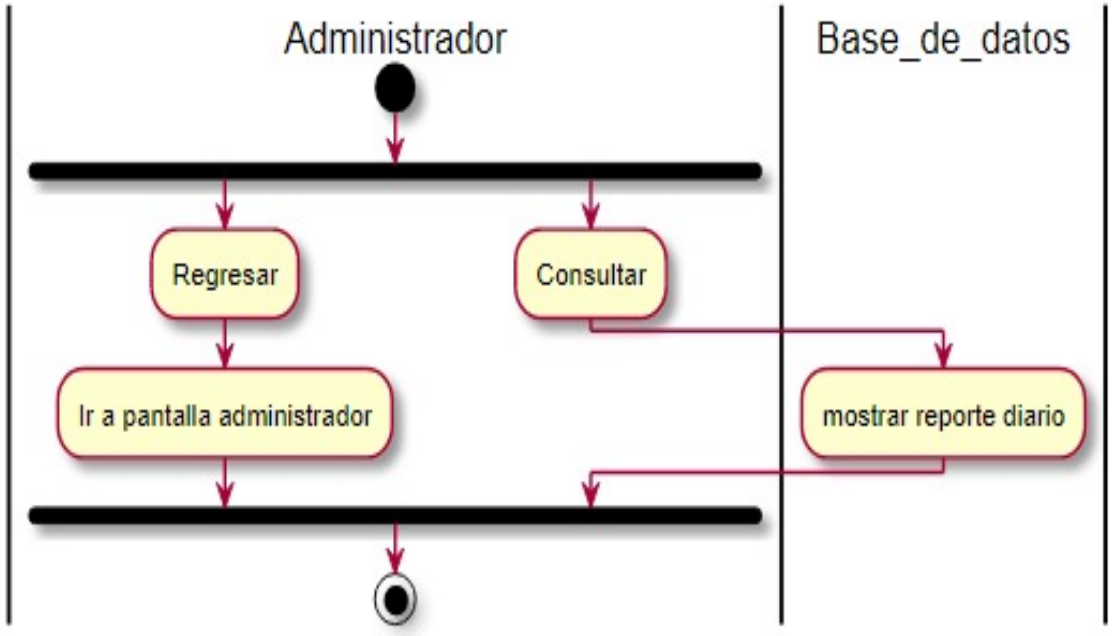


Figura 3.67: Diagrama de Actividades Reporte

Este diagrama muestra las actividades que realizamos en la pantalla de reporte como regresar y ir a pantalla administrador y generar un reporte.

```
Codigo de Diagrama de Actividades Reporte

@startuml
```

```
|Administrador|
start
  fork
 :Regresar;
 :Ir a pantalla administrador;
  fork again
 :Consultar;

|Base_de_datos|
 :mostrar reporte diario;
 endfork
|Administrador|

stop
@enduml
```

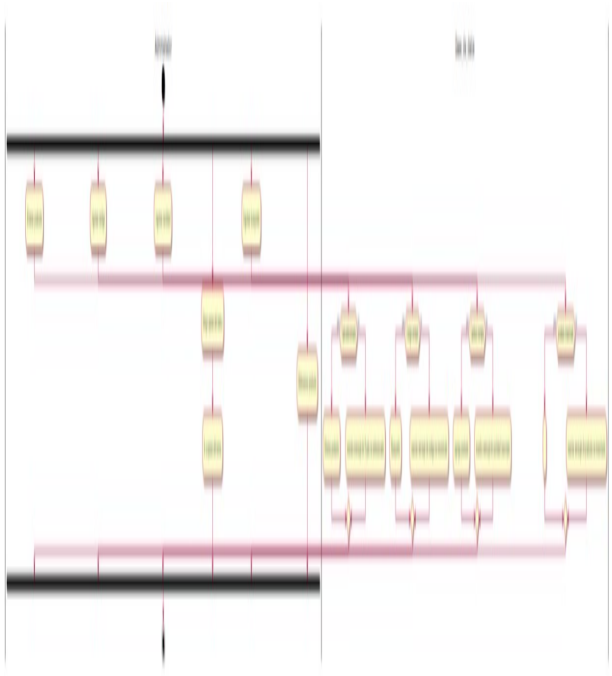


Figura 3.68: Diagrama de Actividades Tabla administrador

El diagrama de actividades de tabla administrador nos muestra las activides que se relizamos en nuestra pantalla administrador como la principal actividad que es realizar venta.

`Codigo de Diagrama de Actividades Tabla administrador`

```
@startuml
|Administrador|
start
```

```
  fork
 :Eliminar producto;
 |Base_de_datos|
    if (Tupla seleccionada) then (yes)
      :Elimina producto;
    else (no)
      :mandar mensaje de Tupla no seleccionada;
    endif
  |Administrador|
  fork again
 :ingresar codigo;
 |Base_de_datos|
    if (Codigo correpto) then (yes)
      :Busqueda;
    else (no)
      :mandar mensaje de codigo no encontrado;
    endif
  |Administrador|
  fork again
 :ingresar cantidad;
 |Base_de_datos|
    if (Cantidad correpta) then (yes)
      :agrega producto;
    else (no)
      :mandar mensaje de cantidad incorrepta;
    endif
  |Administrador|

fork again
 :Elegir opcion del menu;
 :Ir a opcion del menu;
fork again
 :Ingresar busqueda;
 |Base_de_datos|
    if (producto encontrado) then (yes)
      :;
    else (no)
      :mandar mensaje de producto no encontrado;
    endif
  |Administrador|
fork again
 :Seleccionar producto;
 endfork
|Administrador|

stop
@enduml
```

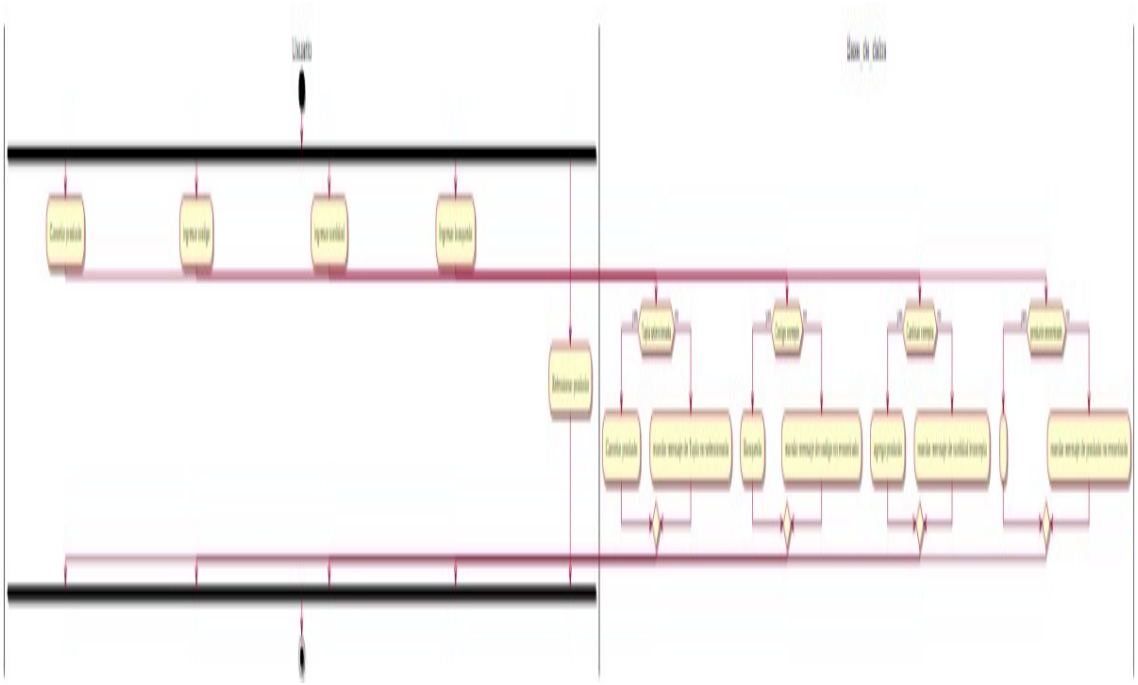


Figura 3.69: Diagrama de Actividades Tabla usuario
Este diagrama de acividades nos muestra las actividades que realiza la pantalla de usuario, asi como cobrar y eliminar producto.

```
Codigo de Diagrama de Actividades Tabla usuario

@startuml
|Usuario|
start
  fork
 :Cancelar producto;
 |Base_de_datos|
    if (Tupla seleccionada) then (yes)
      :Cancelar producto;
    else (no)
      :mandar mensaje de Tupla no seleccionada;
    endif
  |Usuario|
  fork again
 :ingresar codigo;
 |Base_de_datos|
    if (Codigo correpto) then (yes)
      :Busqueda;
    else (no)
      :mandar mensaje de codigo no encontrado;
    endif
  |Usuario|
```

```
  fork again
 :ingresar cantidad;
 |Base_de_datos|
    if (Cantidad correpta) then (yes)
      :agrega producto;
    else (no)
      :mandar mensaje de cantidad incorrepta;
    endif
  |Usuario|

fork again
 :Ingresar busqueda;
 |Base_de_datos|
    if (producto encontrado) then (yes)
      :;
    else (no)
      :mandar mensaje de producto no encontrado;
    endif
  |Usuario|
fork again
 :Seleccionar producto;
 endfork
|Usuario|

stop
@enduml
```

Capítulo 4

Diseño

4.1. Introducción

En este capitulo se abordara la descripción de nuestro sistema de ventas en donde se describirán las diversas funcionalidades que implementamos en nuestro sistema de ventas para que pudiera ser funcional y de ayuda para el usuario, así como también en esta sección se mencionara su arquitectura, de forma física y lógica de el sistema ventas, también se mostrara el estándar de codificación que utilizamos para la realización de nuestro proyecto desarrollado en java, y se describirá a detalle el diseño de datos, para que pueda haber una mayor comprensión sobre nuestro sistema, de igual manera se vera el el modelo conceptual, el modelo lógico y diccionario de datos, los cuales nos permitieron llevar un mejor desarrollo en el diseño de nuestra aplicación, ya que con todos estos puntos se puede tener un mayor conocimiento del sistema de ventas.

4.2. Descripción del sistema

4.3. Arquitectura del sistema

4.3.1. Arquitectura física

Figura 4.1: Interface de inicio

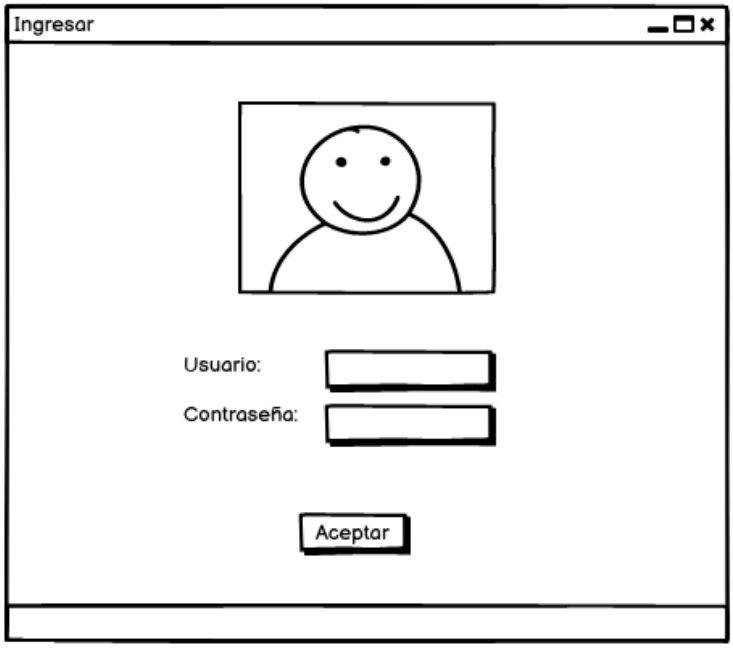


Figura 4.2: Interface de ingreso al usuario

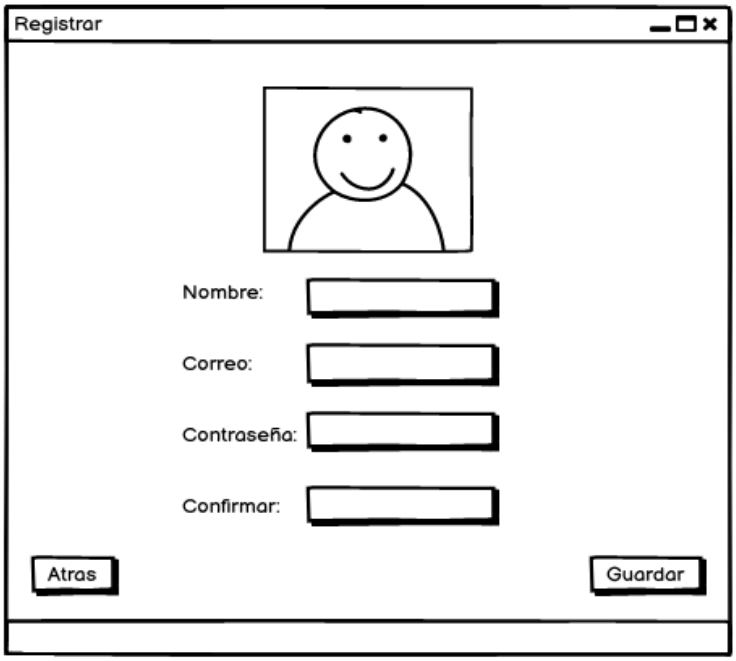


Figura 4.3: Interface de registro de ingreso del usuario

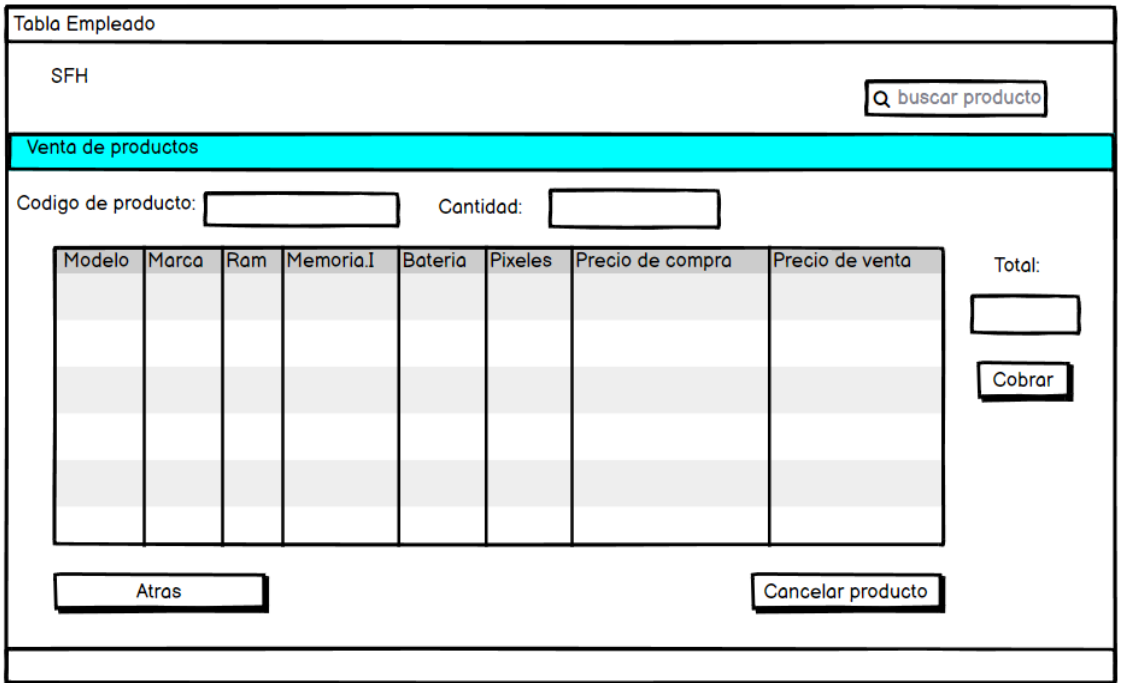


Figura 4.4: Interface de ventas del usuario

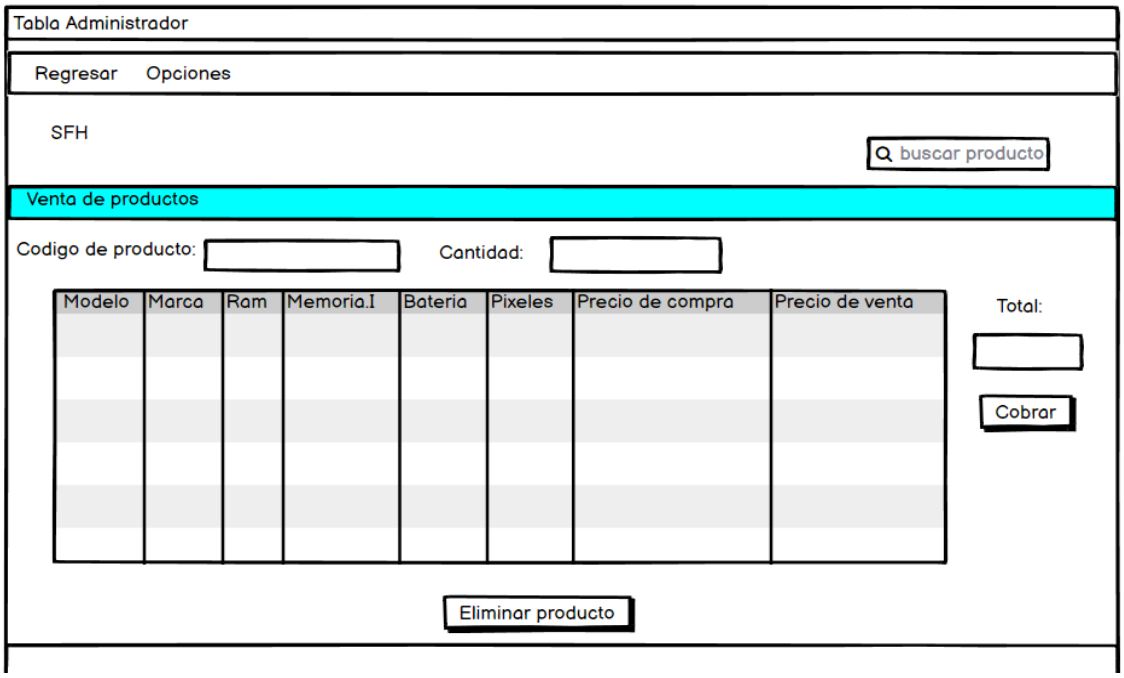


Figura 4.5: Interface de ventas del administrador

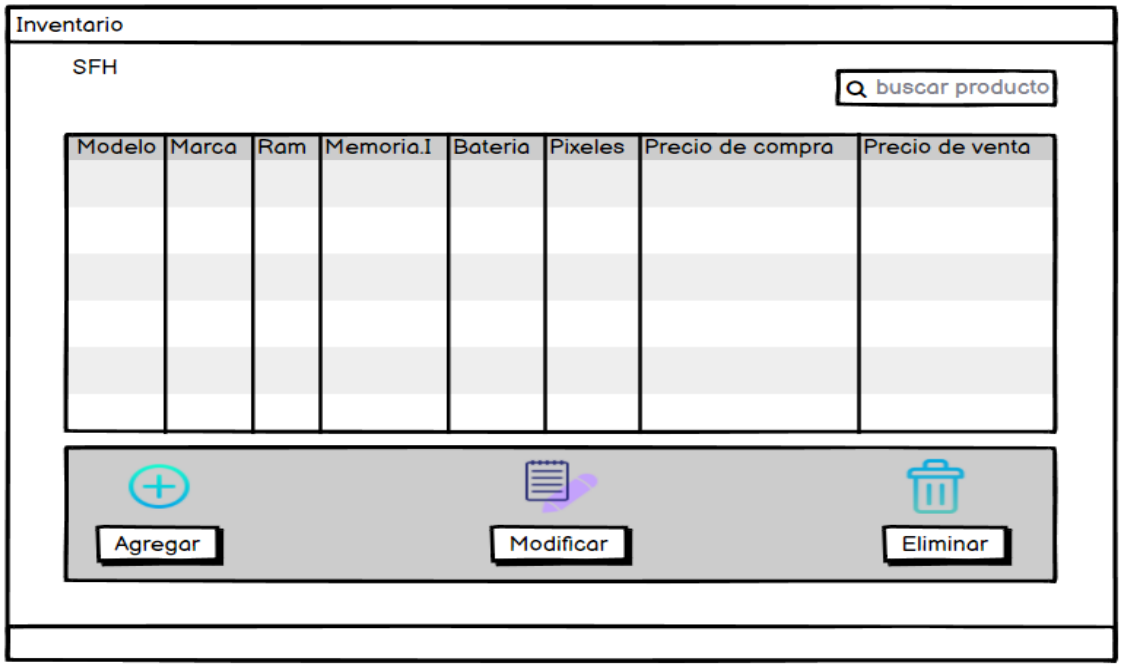


Figura 4.6: Interface de inventario del administrador

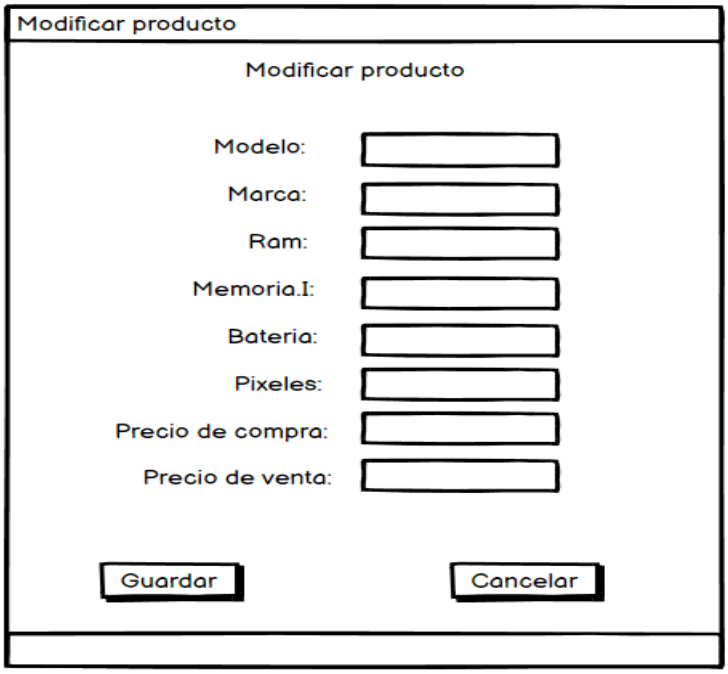


Figura 4.7: Interface de modificación de productos

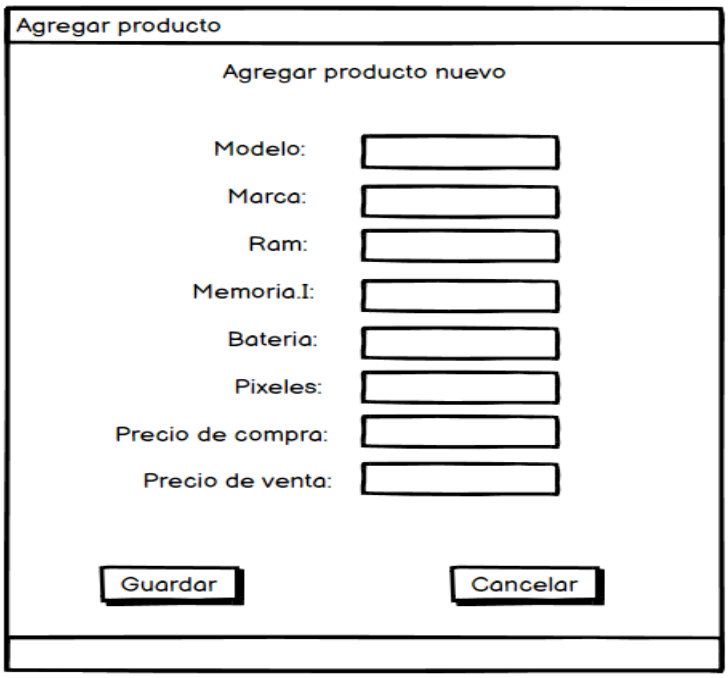


Figura 4.8: Interface de agregación de productos

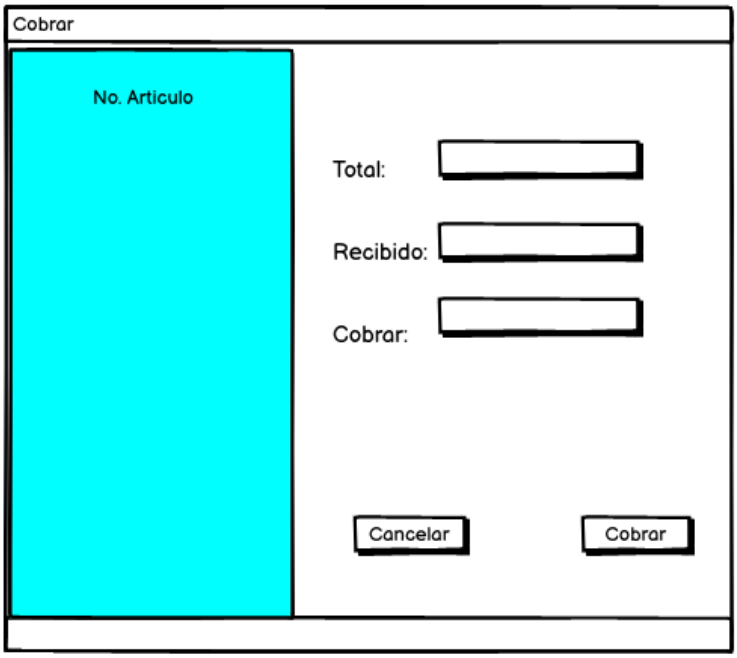


Figura 4.9: Interface de cobro

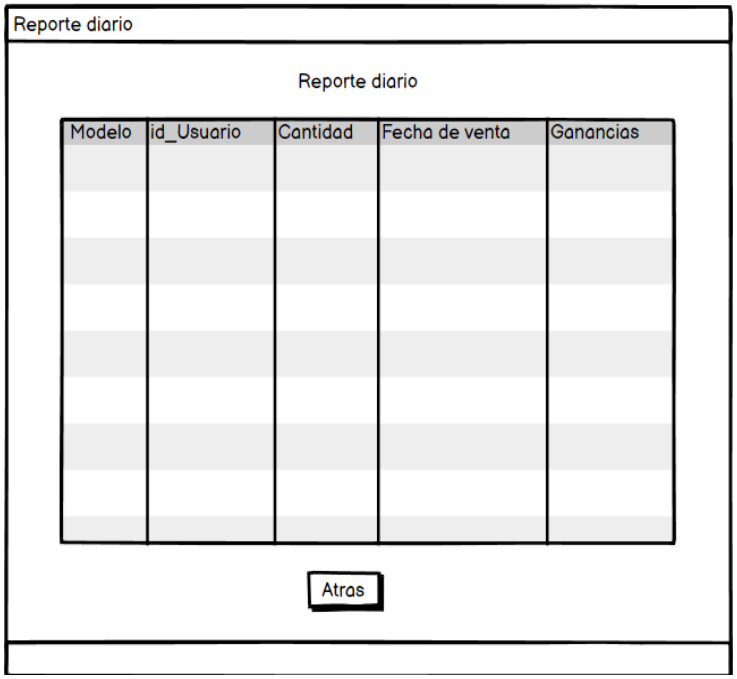


Figura 4.10: Interface de reporte

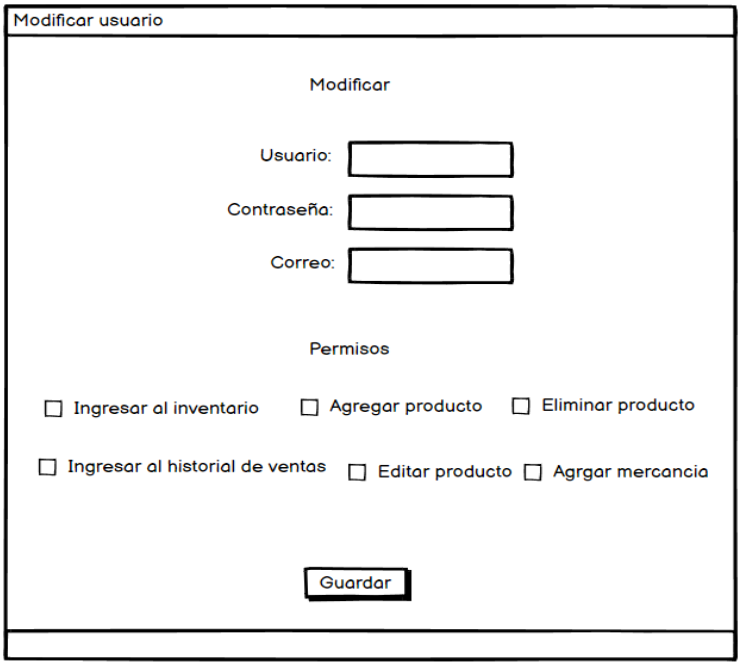


Figura 4.11: Interface de modificar usuario

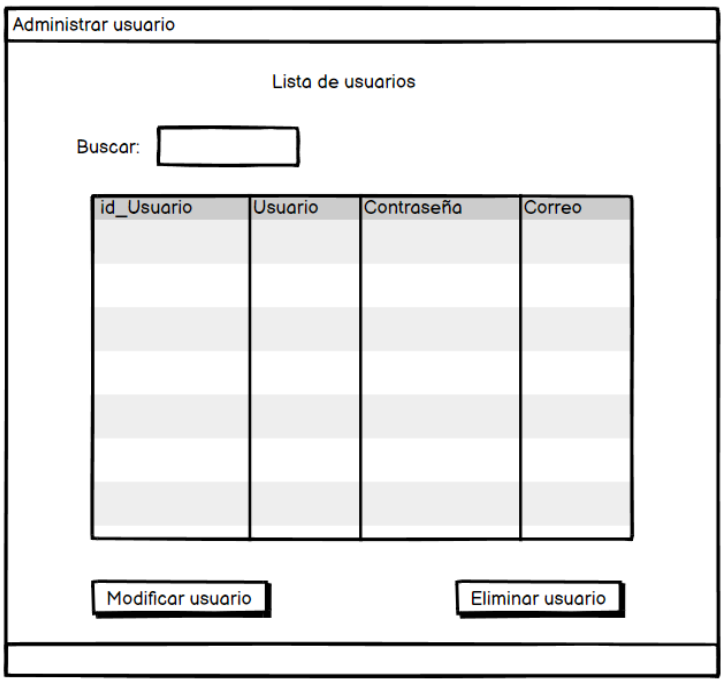


Figura 4.12: Interface de lista de usuarios

4.3.2. Arquitectura lógica

4.4. Estándar de codificación

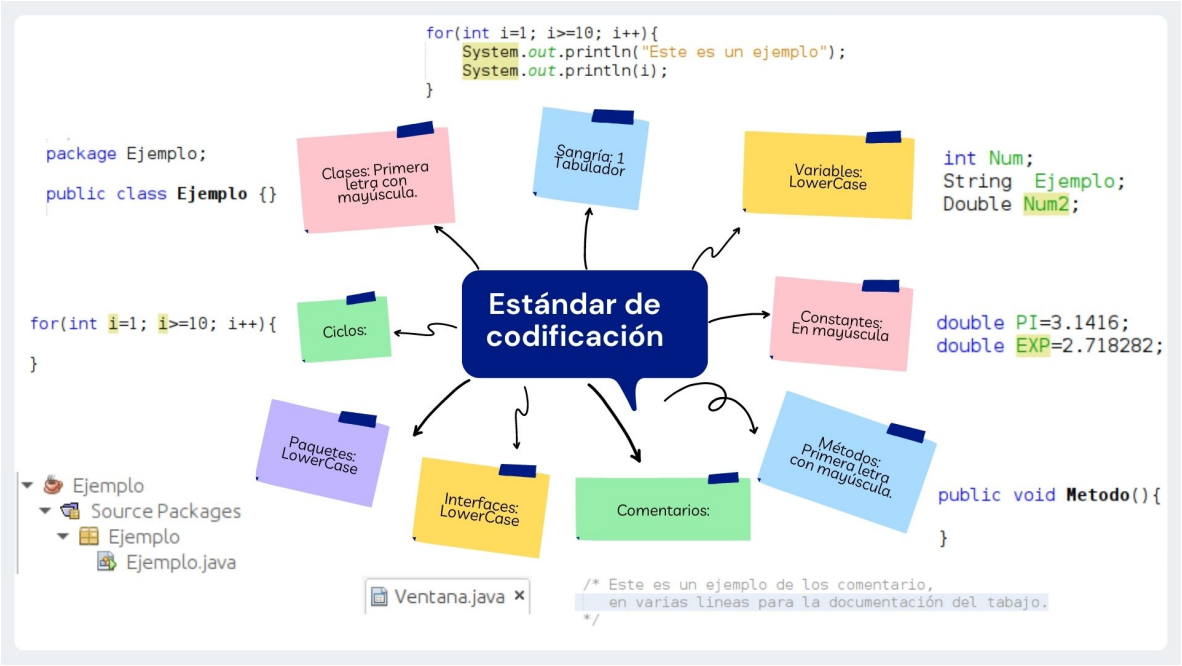


Figura 4.13: Estándar de codificación

4.5. Diseño de datos

4.5.1. Diagrama conceptual

Para poder iniciar a desarrollar el proyecto de sistema de ventas de teléfonos celulares, el cual hasta el momento sigue en proceso de desarrollo. Se requirió de una base de datos el cual lleva como nombre SFH dicha base de datos cuenta con tres tablas las cuales son: tabla de teléfonos, tabla de reporte y tabla de usuarios.

La tabla de teléfonos cuenta con los siguientes campos: campo idmodelo de tipo varchar almacenará el id del modelo este campo actuara como llave primaria, el campo marca de tipo varchar almacenara la marca del teléfono, el campo ram de tipo integer contendrá la capacidad de la memoria ram (memoria de acceso aleatorio), en el campo memoriaint de tipo integer contendrá la capacidad de memoria interna del dispositivo, en el campo batería se especifica la capacidad de la batería del dispositivo, pixeles en este campo se especifica la resolución de la cámara del teléfono este campo es de tipo integer, preven este campo contendrá el precio del dispositivo dicho campo es de tipo flotante, precomp en este campo se almacena el precio de adquisición del dispositivo este campo es de tipo flotante y por ultimo tenemos el campo cantidad el cual contendrá el la cantidad de producto.

La tabla usuarios cuenta con los siguientes campos idU, usuario, contraseña, correo. El campo idU almacenara el número de identificación del usuario, usuario almacenará el nombre del usuario, el campo contraseña la contraseña del usuario y por ultimo el campo correo que como su nombre lo dice contendrá el correo del usuario. Estos tres últimos campos son de tipo varchar mientras que el campo idU es de tipo serial además actuará como llave primaria.

La tabla reporte cuenta con los siguientes campos idmodelo de tipo varchar, idU de tipo serial, cantidad de tipo integer y fechacom de tipo date. El campó idmodelo tendrá el tipo de modelo para poder identificar el dispositivo, idU contendrá el número de identificación del usuario, cantidad tendrá la cantidad de dispositivos vendidos y por último en el campo fechacom va a almacenar la fecha de compra de dispositivo. El campo idmodelo se relaciona con la tabla teléfonos y el campo idU se relaciona con la tabla usuarios.

4.5.2. Diagrama Entidad-Relación

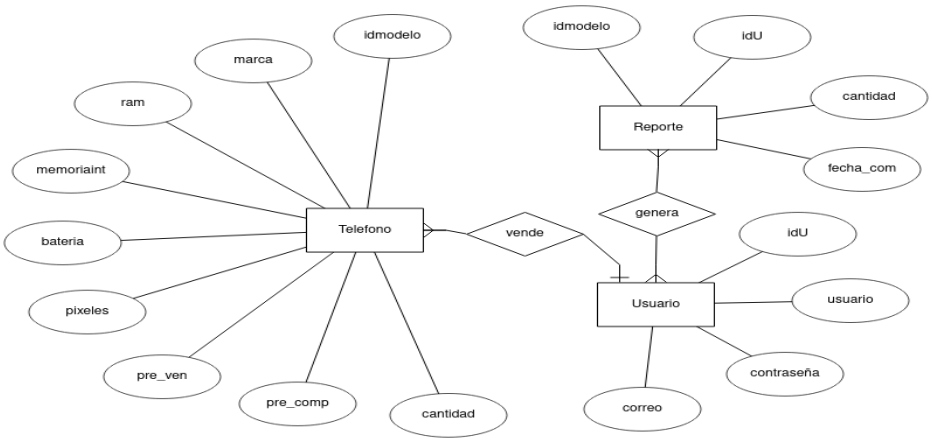


Figura 4.14: Diagrama Entidad-Relación del proyecto de telélefonos celulares.

4.5.3. Diagrama de Modelo Relacional

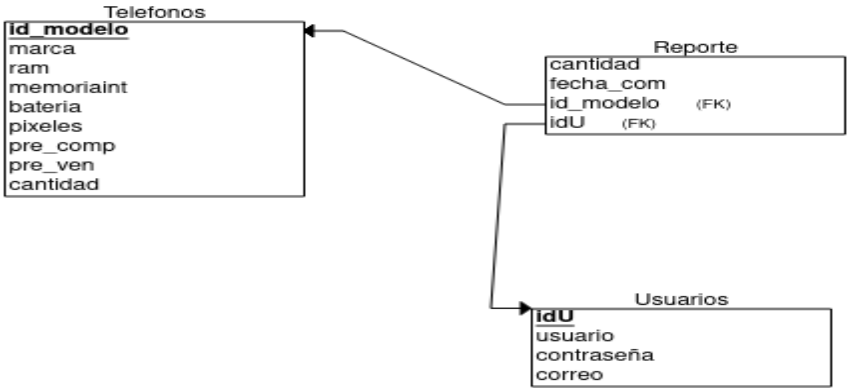


Figura 4.15: Diagrma de modelo Relacional del proyecto de telélefonos celulares

4.5.4. Diagrama físico

```
1
2  CREATE DATABASE SFH
3
4  CREATE TABLE TELEFONOS (id_modelo varchar(20) primary key,
5  marca varchar(20),
6  ram integer,
7  memoriaint integer,
8  bateria integer,
9  pixeles integer,
10 pre_ven float,
11 pre_comp float,
12 cantidad integer on UPDATE CASCADE ON DELETE CASCADE);
13
14 CREATE TABLE USUARIOS (idU serial primary key,
15 usuario varchar(20),
16 contrasena varchar(20),
17 correo varchar(20) UPDATE CASCADE ON DELETE CASCADE);
18
19 CREATE TABLE REPORTE (id_modelo varchar(20),
20 idU serial,
21 cantidad integer,
22 fecha_com varchar,
23 primary key(id_modelo,idU),
24 foreign key (id_modelo) references TELEFONOS(id_modelo),
25 foreign key (idU) references USUARIOS(idU) ON UPDATE RESTRINT ON DELETE RESTRINT);
```

Figura 4.16: Diagrama físico de la creación de base de datos SFH.

4.5.5. Diccionario de datos

Esta tabla contiene 9 columnas con diferentes tipos de datos, los cuales incluye de tipo integer, varchar y float, cuenta con solo una llave primaria. Los datos van referenciados a las caracteristicas de los teléfonos.

TABLA TELÉFONOS

	Columna	Tipo de dato	Descripción
PK	id_modelo	varchar(20)	Tipo de modelo para identificar el dispositivo
	marca	varchar(20)	Marca del dispositivo
	ram	integer	Capacidad de Ram del dispositivo
	memoriaint	integer	capacidad de memoria interna del dispositivo
	bateria	integer	Capacidad de batería del dispositivo
	pixeles	integer	Cantidad de pixeles del dispositivo
	pre_ven	float	Precio de venta del dispositivo
	pre_comp	float	Precio de compra del dispositivo
	cantidad	integer	Cantidad de dispositivos en existencia

Figura 4.17: Tabla Teléfonos

Esta tabla contiene 4 columnas con dos tipos de datos, los cuales incluye de tipo varchar y serial, el cual es un entero que se irá actualizando automáticamente, solo contiene una llave primaria. Los datos se desglosan en datos que ayudaran al usuario a ingresar.

TABLA USUARIOS

	Columna	Tipo de dato	Descripción
PK	idU	serial	Numero de identificación para el usuario
	usuario	varchar(30)	Nombre del usuario
	contrasena	varchar(20)	Contraseña del usuario
	correo	varchar(30)	Correo del usuario

Figura 4.18: Tabla Usuarios

Esta tabla contiene 4 columnas con tres tipos de datos, los cuales incluye de tipo varchar y serial,

date, también solo contiene una llave primaria. La tabla contiene datos de reportes generados en dierentes fechas.

TABLA REPORTE

	Columna	Tipo de dato	Descripción
FK	id_modelo	varchar(20)	Tipo de modelo para identificar el dispositivo
FK	idU	serial	Numero de identificación para el usuario
	cantidad	integer	Cantidad de dispositivos vendidos
	fecha_com	date	Fecha de compra del dispositivo

Figura 4.19: Tabla Reporte

Capítulo 5

Implementación

5.1. Intruducción

La implementación es la parte del proceso en que los desarrolladores programan el código para el proyecto. Las pruebas del software son parte esencial del proceso de desarrollo del software. Esta parte del proceso tiene la función de detectar los errores de software lo antes posible. Durante el desarrollo del software se realiza la documentación del diseño del código interiormente como exteriormente esto con el objetivo de facilitar su mejora y su mantenimiento futuro.

5.2. Código fuente

```
Codigo de Framer ventas
package venta;

import java.applet.AudioClip;
import java.io.FileNotFoundException;
import static java.lang.Thread.sleep;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import java.util.logging.*;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.RowFilter;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
import org.apache.poi.xssf.usermodel.*;

/**
 *
 * @author emili
```

```
 */
public class Ventas extends javax.swing.JFrame implements Runnable{

    /**
     * Creates new form Ventas
     */
     DefaultTableModel modelo1= new DefaultTableModel();
     DefaultTableModel modelo2= new DefaultTableModel();
     DefaultTableModel modelo3= new DefaultTableModel();
     DefaultTableModel modelo4= new DefaultTableModel();
     DefaultTableModel modelo5= new DefaultTableModel();
     DefaultTableModel modelo6= new DefaultTableModel();
     DefaultTableModel modelo7= new DefaultTableModel();
     DefaultTableModel modelo8= new DefaultTableModel();

    public Ventas() {
        initComponents();
        Thread hilo1 =new Thread(this);
        hilo1.start();
        this.setIconImage(m.Seticono());
        this.setLocationRelativeTo(null);
        modelo1=(DefaultTableModel)tablaInventario.getModel();
        modelo2=(DefaultTableModel)tablaInventario1.getModel();
        modelo3=(DefaultTableModel)tablaInventario2.getModel();
        modelo4=(DefaultTableModel)tablaUsuario.getModel();
        modelo5=(DefaultTableModel)tablaReporte.getModel();
        modelo6=(DefaultTableModel)tablaInventario3.getModel();
        modelo7=(DefaultTableModel)Tablapre_venta.getModel();
        modelo8=(DefaultTableModel)Tablapre_venta1.getModel();
    }
    static int cont=0; static boolean ban=true;
    static String a="VENTA DE PRODUCTOS
    static String a2="
    Metodos m=new Metodos();
    String bd="sfh";
    String url=null;
    String tableu="USUARIOS";
    String tablet="TELEFONOS";
    String tabler="REPORTE";
    String tablev="ventas";
    String id;
    String usuario;
    Double total=0.0;
    Double obtener=0.0;

    int idp,canti,confirmar;

    Double resta;
    int rou,cant1,cant2,validar;
```

```
String modelo;
int cd[];
String mode[];
int columna;
String hora, minutos,segundos;
public Clip clip;
public String rut ="/audio/";

   public void consultarTelefono(DefaultTableModel mod,JTable tab){
      // Con este método consultamos los ptoductos haciendo una
      consulta en la base de datos
      // posteriormente pintamos esos datos en la tabla
       Object[] ob = new Object[9];
    try{
      Statement s=  m.conectaBase(bd).createStatement();
       //Declaramos un objeto de tipo Resulset
       ResultSet rs= s.executeQuery("Select *from "+tablet+" ORDER
       BY id_modelo ASC;");

       while (rs.next()){// insertamos el reultado obtenido a la
       tabla
           ob[0] = rs.getObject("id_modelo");
           ob[1] = rs.getObject("marca");
           ob[2] = rs.getObject("ram");
           ob[3] = rs.getObject("memoriaint");
           ob[4] = rs.getObject("bateria");
           ob[5] = rs.getObject("pixeles");
           ob[6] = rs.getObject("pre_ven");
           ob[7] = rs.getObject("pre_comp");
           ob[8] = rs.getObject("cantidad");
           mod.addRow(ob);
           }
      m.conectaBase(bd).close();
     }catch(Exception e){
       System.out.println(e.getMessage());
   }
      tab.setModel(mod);
}



   public void consultarTelefonoventa(DefaultTableModel mod,JTable
   tab){
      //Este método se implementa para extraer de la base de datos,
      de la tabla telefonos ciertos datos que se requieren para
      // poder realizar la venta.
       Object[] ob = new Object[9];
    try{
      Statement s=  m.conectaBase(bd).createStatement();
```

```
        //Declaramos un objeto de tipo Resulset
        ResultSet rs= s.executeQuery("Select id_modelo,marca,pre_ven,cantidad from "+tablet+" wher
        cantidad>0 ORDER BY id_modelo ASC;");

        while (rs.next()){
            ob[0] = rs.getObject("id_modelo");
            ob[1] = rs.getObject("marca");
            ob[2] = rs.getObject("pre_ven");
            ob[3] = rs.getObject("cantidad");
            mod.addRow(ob);
            }
       m.conectaBase(bd).close();
      }catch(Exception e){
        System.out.println(e.getMessage());
    }
       tab.setModel(mod);
}

    public void consultarTelefonoventa2(DefaultTableModel mod,
    JTable tab, String modelo){
       // ConsultarTelefonoventa2, permite realizar una consulta
       a la base de datos para extraer los datos de
       //  id_modelo,pre_ven,cantidad de la tabla Telefonos
       y poder pintarlos en la tabla
        Object[] ob = new Object[9];
     try{
       Statement s=  m.conectaBase(bd).createStatement();
        //Declaramos un objeto de tipo Resulset

        ResultSet rs= s.executeQuery("Select id_modelo,pre_ven,
        cantidad from "+tablet+" where id_modelo='"+modelo+"'
        ORDER BY id_modelo ASC;");

        while (rs.next()){
            ob[0] = rs.getObject("id_modelo");
            ob[1] = rs.getObject("pre_ven");
            ob[2] = cant1;
            mod.addRow(ob);
            }
        cant1=0;
        cant2=0;
       m.conectaBase(bd).close();
      }catch(Exception e){
        System.out.println(e.getMessage());
    }
       tab.setModel(mod);
}
    public void consultarTelefonoventa3(DefaultTableModel
    mod,JTable tab, String modelo){
```

```
      // ConsultarTelefonoventa2, permite realizar una consulta
      a la base de datos para extraer los datos de
      //  id_modelo,pre_ven,cantidad de la tabla Telefonos y poder pintarlos en la tabla

       Object[] ob = new Object[9];
    try{
      Statement s=  m.conectaBase(bd).createStatement();
       //Declaramos un objeto de tipo Resulset

       ResultSet rs= s.executeQuery("Select id_modelo,pre_ven,
       cantidad from "+tablet+" where id_modelo='"+modelo+"'
       ORDER BY id_modelo ASC;");
      // String canti =txtCantidad.getText();
       while (rs.next()){
           ob[0] = rs.getObject("id_modelo");
           ob[1] = rs.getObject("pre_ven");
           ob[2] = rs.getObject("cantidad");
           mod.addRow(ob);
           }
      m.conectaBase(bd).close();
     }catch(Exception e){
       System.out.println(e.getMessage());
   }
      tab.setModel(mod);
}


   public void consultarUsuarios(){
      //El método consultarUsuario permite realizar una
      consulta a la base de datos para poder
      // extraer todos los usuarios para poder pintarlo en la
      tabla tablaUsuario para poder administrarlos
       Object[] ob = new Object[4];
    try{
      Statement s=  m.conectaBase(bd).createStatement();
       //Declaramos un objeto de tipo Resulset
       ResultSet rs= s.executeQuery("Select *from "+tableu+"
       ORDER BY idU ASC;");
       while (rs.next()){
           ob[0] = rs.getObject("idU");
           ob[1] = rs.getObject("usuario");
           ob[2] = rs.getObject("correo");
           modelo4.addRow(ob);
           }
      m.conectaBase(bd).close();
     }catch(Exception e){
       System.out.println(e.getMessage());
   }
      tablaUsuario.setModel(modelo4);
```

```
    }



    public void consultarReportes(){
        //Se realiza una consulta a la base de datos de la
        tabla venta para extrar toda los registros de los
        telefonos que
        //fueron vendidos en sus respectivas fechas, la
        información extraida se pintara en la tabla reporte perimitiendo
        //observar el flujo de las ventas de los productos
         Object[] ob = new Object[8];
        try{
          Statement s=  m.conectaBase(bd).createStatement();
           //Declaramos un objeto de tipo Resulset
           ResultSet rs= s.executeQuery("Select *from "+tablev+"
           ORDER BY id_venta ASC;");
           while (rs.next()){
               ob[0] = rs.getObject("modelo");
               ob[1] = rs.getObject("cantidad");
               ob[2] = rs.getObject("fecha");
//               ob[3] = rs.getObject("ganancia");
               modelo5.addRow(ob);
               }
          m.conectaBase(bd).close();
         }catch(Exception e){
           System.out.println(e.getMessage());
       }
          tablaReporte.setModel(modelo5);
    }




    private void limpiaTabla(DefaultTableModel mod){
    //   El método limpiaTabla permite limpiar sierta tabla al
    ingreasr su modelo
         int a=mod.getRowCount();
     while(a!=0){ // Ciclo para Borrar la Tabla 1
         if(a!=0)
         mod.removeRow(0);
         a=mod.getRowCount();
    }
    }


    /**
     * This method is called from within the constructor to
     initialize the form.
```

```
 * WARNING: Do NOT modify this code. The content of this
 method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    Registrar = new javax.swing.JDialog();
    panelRegistro = new javax.swing.JPanel();
    lblNombre = new javax.swing.JLabel();
    lblCorreo = new javax.swing.JLabel();
    lblContrasena = new javax.swing.JLabel();
    lblConfirmar = new javax.swing.JLabel();
    txtConfirmar = new javax.swing.JPasswordField();
    txtContrasena = new javax.swing.JPasswordField();
    txtCorreo = new javax.swing.JTextField();
    txtNombre = new javax.swing.JTextField();
    lblImagen = new javax.swing.JLabel();
    btnGuardar = new javax.swing.JButton();
    btnAtras = new javax.swing.JButton();
    c = new javax.swing.JLabel();
    jSeparator3 = new javax.swing.JSeparator();
    jSeparator4 = new javax.swing.JSeparator();
    jSeparator5 = new javax.swing.JSeparator();
    jSeparator6 = new javax.swing.JSeparator();
    jSeparator7 = new javax.swing.JSeparator();
    contraAdmin = new javax.swing.JPasswordField();
    Inventario = new javax.swing.JDialog();
    panelInventario = new javax.swing.JPanel();
    scrollpaneInventario = new javax.swing.JScrollPane();
    tablaInventario = new javax.swing.JTable();
    panelImagen = new javax.swing.JPanel();
    jLabel13 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    panelCambios = new javax.swing.JPanel();
    btnAgregar = new javax.swing.JButton();
    btnModificar = new javax.swing.JButton();
    btnEliminar = new javax.swing.JButton();
    imagenEliminar = new javax.swing.JLabel();
    imagenAgregar = new javax.swing.JLabel();
    imagenModificar = new javax.swing.JLabel();
    btnAtrasinventario = new javax.swing.JButton();
    jLabel28 = new javax.swing.JLabel();
    txtBuscar = new javax.swing.JTextField();
    lblBuscador = new javax.swing.JLabel();
    TablaEmpleado = new javax.swing.JDialog();
    panelEmpleado = new javax.swing.JPanel();
    btnCobrar = new javax.swing.JButton();
```

```
lblTotal = new javax.swing.JLabel();
txtTotal = new javax.swing.JTextField();
btnCancelarproductor = new javax.swing.JButton();
panelColor = new javax.swing.JPanel();
letrero2 = new javax.swing.JLabel();
hora2 = new javax.swing.JLabel();
logo = new javax.swing.JLabel();
lblCodigo = new javax.swing.JLabel();
txtCodigo = new javax.swing.JTextField();
lblCantidad = new javax.swing.JLabel();
atrasBtn = new javax.swing.JButton();
letrasLogo = new javax.swing.JLabel();
scrollpaneInventario2 = new javax.swing.JScrollPane();
tablaInventario2 = new javax.swing.JTable();
jToolBar1 = new javax.swing.JToolBar();
inventario1 = new javax.swing.JButton();
historialv = new javax.swing.JButton();
jScrollPane4 = new javax.swing.JScrollPane();
Tablapre_venta1 = new javax.swing.JTable();
cp1 = new javax.swing.JSpinner();
TablaAdministrador = new javax.swing.JDialog();
panelAdministrador = new javax.swing.JPanel();
colorPanel = new javax.swing.JPanel();
letrero = new javax.swing.JLabel();
relog = new javax.swing.JLabel();
imagenPanel = new javax.swing.JPanel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
panelProducto = new javax.swing.JPanel();
jLabel25 = new javax.swing.JLabel();
tfCodigo = new javax.swing.JTextField();
jLabel26 = new javax.swing.JLabel();
pc = new javax.swing.JSpinner();
botonCobrar = new javax.swing.JButton();
botonEliminarp = new javax.swing.JButton();
textCobrar = new javax.swing.JTextField();
labelTotal = new javax.swing.JLabel();
scrollpaneInventario1 = new javax.swing.JScrollPane();
tablaInventario1 = new javax.swing.JTable();
jScrollPane3 = new javax.swing.JScrollPane();
Tablapre_venta = new javax.swing.JTable();
jMenuBar1 = new javax.swing.JMenuBar();
menuAtras = new javax.swing.JMenu();
jMenuItem3 = new javax.swing.JMenuItem();
menuPermisos = new javax.swing.JMenu();
itemInventario = new javax.swing.JMenuItem();
itemUsuarios = new javax.swing.JMenuItem();
itemReporte = new javax.swing.JMenuItem();
Cobrar = new javax.swing.JDialog();
```

```
panelCobrar = new javax.swing.JPanel();
btnCancelarp = new javax.swing.JButton();
totalLbl = new javax.swing.JLabel();
txtRecibido = new javax.swing.JTextField();
lblRecibido = new javax.swing.JLabel();
btnCobrarp = new javax.swing.JButton();
txtTotalp = new javax.swing.JTextField();
panelProductos = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jLabel30 = new javax.swing.JLabel();
NoP = new javax.swing.JTextField();
fd = new javax.swing.JLabel();
jSeparator8 = new javax.swing.JSeparator();
jSeparator9 = new javax.swing.JSeparator();
jSeparator11 = new javax.swing.JSeparator();
mensajeReci = new javax.swing.JLabel();
AgregarP = new javax.swing.JDialog();
panelAgregarp = new javax.swing.JPanel();
lblCodigop = new javax.swing.JLabel();
lblNombrep = new javax.swing.JLabel();
lblPreciov = new javax.swing.JLabel();
lblPrecioc = new javax.swing.JLabel();
lblCantidadp = new javax.swing.JLabel();
lblMarcap = new javax.swing.JLabel();
txtModelo = new javax.swing.JTextField();
txtMarca = new javax.swing.JTextField();
txtRam = new javax.swing.JTextField();
txtMemoria = new javax.swing.JTextField();
txtBateria = new javax.swing.JTextField();
txtPixeles = new javax.swing.JTextField();
lblLetrero = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
txtPrecioc = new javax.swing.JTextField();
txtPreciov = new javax.swing.JTextField();
txtCantidadm = new javax.swing.JTextField();
btnGuardarp = new javax.swing.JButton();
btnCamcelarpro = new javax.swing.JButton();
jLabel31 = new javax.swing.JLabel();
jSeparator13 = new javax.swing.JSeparator();
jSeparator14 = new javax.swing.JSeparator();
jSeparator15 = new javax.swing.JSeparator();
jSeparator16 = new javax.swing.JSeparator();
jSeparator17 = new javax.swing.JSeparator();
jSeparator18 = new javax.swing.JSeparator();
jSeparator19 = new javax.swing.JSeparator();
jSeparator20 = new javax.swing.JSeparator();
jSeparator21 = new javax.swing.JSeparator();
```

```
jSeparator22 = new javax.swing.JSeparator();
ModificarP = new javax.swing.JDialog();
panelAgregarp1 = new javax.swing.JPanel();
lblNombrep1 = new javax.swing.JLabel();
lblPreciov1 = new javax.swing.JLabel();
lblPrecioc1 = new javax.swing.JLabel();
lblCantidadp1 = new javax.swing.JLabel();
lblMarcap1 = new javax.swing.JLabel();
txtMarcam = new javax.swing.JTextField();
txtRamm = new javax.swing.JTextField();
txtMemoriam = new javax.swing.JTextField();
txtBateriam = new javax.swing.JTextField();
txtPixelesm = new javax.swing.JTextField();
lblLetrero1 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
txtcPreciovm = new javax.swing.JTextField();
txtPreciocm = new javax.swing.JTextField();
txtcantm = new javax.swing.JTextField();
btnGuardarp1 = new javax.swing.JButton();
btnCamcelarpro1 = new javax.swing.JButton();
jLabel32 = new javax.swing.JLabel();
jSeparator12 = new javax.swing.JSeparator();
jSeparator23 = new javax.swing.JSeparator();
jSeparator24 = new javax.swing.JSeparator();
jSeparator25 = new javax.swing.JSeparator();
jSeparator26 = new javax.swing.JSeparator();
jSeparator27 = new javax.swing.JSeparator();
jSeparator28 = new javax.swing.JSeparator();
jSeparator29 = new javax.swing.JSeparator();
jSeparator30 = new javax.swing.JSeparator();
jSeparator31 = new javax.swing.JSeparator();
jSeparator32 = new javax.swing.JSeparator();
Usuario = new javax.swing.JDialog();
panelUsuario = new javax.swing.JPanel();
imagenUsuario = new javax.swing.JLabel();
aceptarAdministrador = new javax.swing.JButton();
txtNombreu = new javax.swing.JTextField();
imgNombre = new javax.swing.JLabel();
imgContrasena = new javax.swing.JLabel();
txtContrasenau = new javax.swing.JPasswordField();
lblNombreu = new javax.swing.JLabel();
lblContraseñau = new javax.swing.JLabel();
mensajeA = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
jSeparator2 = new javax.swing.JSeparator();
Administrarusuario = new javax.swing.JDialog();
jPanel1 = new javax.swing.JPanel();
```

```
jScrollPane1 = new javax.swing.JScrollPane();
tablaUsuario = new javax.swing.JTable();
jLabel1 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
buscarU = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel33 = new javax.swing.JLabel();
Modificarusuario = new javax.swing.JDialog();
jPanel2 = new javax.swing.JPanel();
jLabel4 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
txfusuario = new javax.swing.JTextField();
txfcorreo = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
ingresarInventario = new javax.swing.JCheckBox();
ingresarHistorial = new javax.swing.JCheckBox();
agregarProducto = new javax.swing.JCheckBox();
editarProducto = new javax.swing.JCheckBox();
eliminarProducto = new javax.swing.JCheckBox();
jLabel10 = new javax.swing.JLabel();
btnguardaru = new javax.swing.JButton();
btncancelaru1 = new javax.swing.JButton();
jLabel34 = new javax.swing.JLabel();
Reporte = new javax.swing.JDialog();
jLabel12 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
tablaReporte = new javax.swing.JTable();
botonatrasRep = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
buscarR = new javax.swing.JTextField();
jLabel35 = new javax.swing.JLabel();
btnExcel = new javax.swing.JButton();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
Inventario1 = new javax.swing.JDialog();
panelInventario1 = new javax.swing.JPanel();
scrollpaneInventario3 = new javax.swing.JScrollPane();
tablaInventario3 = new javax.swing.JTable();
panelImagen1 = new javax.swing.JPanel();
jLabel24 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
panelCambios1 = new javax.swing.JPanel();
btnAgregar1 = new javax.swing.JButton();
btnModificar1 = new javax.swing.JButton();
btnEliminar1 = new javax.swing.JButton();
imagenEliminar1 = new javax.swing.JLabel();
```

```
imagenAgregar1 = new javax.swing.JLabel();
imagenModificar1 = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
btnAtrasinventario1 = new javax.swing.JButton();
txtBuscar1 = new javax.swing.JTextField();
lblBuscador1 = new javax.swing.JLabel();
panelInicio = new javax.swing.JPanel();
imagenLogo = new javax.swing.JLabel();
Menu = new javax.swing.JMenuBar();
menuInicio = new javax.swing.JMenu();
iniciarSesion = new javax.swing.JMenuItem();
registrarse = new javax.swing.JMenuItem();

panelRegistro.setBackground(new java.awt.Color
(241, 255, 255));
panelRegistro.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

lblNombre.setFont(new java.awt.Font(
"DejaVu Serif", 3, 15)); // NOI18N
lblNombre.setText("Nombre:");
panelRegistro.add(lblNombre, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 160, -1, -

lblCorreo.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
lblCorreo.setText("Teléfono:");
panelRegistro.add(lblCorreo, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 190, -1, -

lblContrasena.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
lblContrasena.setText("Contraseña:");
panelRegistro.add(lblContrasena, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 220, -

lblConfirmar.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
lblConfirmar.setText("Confirmar C:");
panelRegistro.add(lblConfirmar, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 250, -1

txtConfirmar.setBackground(new java.awt.Color(241, 255, 255));
txtConfirmar.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtConfirmar.setBorder(null);
txtConfirmar.addActionListener(new java.awt.event.ActionListener
() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtConfirmarActionPerformed(evt);
    }
});
txtConfirmar.addKeyListener(new java.awt.event.KeyAdapter() {
```

```
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtConfirmarKeyTyped(evt);
    }
});
panelRegistro.add(txtConfirmar, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 250, 1

txtContrasena.setBackground(new java.awt.Color(241, 255, 255));
txtContrasena.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtContrasena.setBorder(null);
txtContrasena.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtContrasenaKeyTyped(evt);
    }
});
panelRegistro.add(txtContrasena, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 220,

txtCorreo.setBackground(new java.awt.Color(241, 255, 255));
txtCorreo.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtCorreo.setBorder(null);
txtCorreo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCorreoKeyTyped(evt);
    }
});
panelRegistro.add(txtCorreo, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 190, 140,

txtNombre.setBackground(new java.awt.Color(241, 255, 255));
txtNombre.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtNombre.setAutoscrolls(false);
txtNombre.setBorder(null);
txtNombre.setDisabledTextColor(new java.awt.Color(241, 255, 255));
txtNombre.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNombreKeyTyped(evt);
    }
});
panelRegistro.add(txtNombre, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 160, 140,

lblImagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/usuario (3).png")
panelRegistro.add(lblImagen, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 10, -1, 1

btnGuardar.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
btnGuardar.setText("Guardar");
btnGuardar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGuardarActionPerformed(evt);
    }
});
```

```
panelRegistro.add(btnGuardar, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 350, -1,

btnAtras.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); //
NOI18N
btnAtras.setText("Regresar");
btnAtras.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAtrasActionPerformed(evt);
    }
});
panelRegistro.add(btnAtras, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 350, 116, -

c.setFont(new java.awt.Font("DejaVu Serif", 3, 14)); // NOI18N
c.setText("Codigo Adm:");
panelRegistro.add(c, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 280, 113, -1));
panelRegistro.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 300, 14
panelRegistro.add(jSeparator4, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 180, 14

jSeparator5.setForeground(new java.awt.Color(0, 0, 0));
panelRegistro.add(jSeparator5, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 210, 14

jSeparator6.setBackground(new java.awt.Color(153, 153, 153));
panelRegistro.add(jSeparator6, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 240, 14
panelRegistro.add(jSeparator7, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 270, 14

contraAdmin.setBackground(new java.awt.Color(241, 255, 255));
contraAdmin.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
contraAdmin.setBorder(null);
panelRegistro.add(contraAdmin, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 280, 14

javax.swing.GroupLayout RegistrarLayout = new javax.swing.GroupLayout(Registrar.getContentPan
Registrar.getContentPane().setLayout(RegistrarLayout);
RegistrarLayout.setHorizontalGroup(
    RegistrarLayout.createParallelGroup(javax.swing.GroupLayout.
    Alignment.LEADING)
    .addComponent(panelRegistro, javax.swing.GroupLayout.DEFAULT_SIZE, 333, Short.MAX_VALUE)
);
RegistrarLayout.setVerticalGroup(
    RegistrarLayout.createParallelGroup(javax.swing.GroupLayout.
    Alignment.LEADING)
    .addComponent(panelRegistro, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.Grou
);

tablaInventario.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N
tablaInventario.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
```

```
    new String [] {
        "Modelo", "Marca", "Ram", "M.Interna", "Bateria", "Pixeles", "Precio de compra", "Pre
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false,
        false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex)
    {
        return canEdit [columnIndex];
    }
});
tablaInventario.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tablaInventarioMouseClicked(evt);
    }
});
scrollpaneInventario.setViewportView(tablaInventario);
if (tablaInventario.getColumnModel().getColumnCount() > 0) {
    tablaInventario.getColumnModel().getColumn(0).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(1).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(2).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(3).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(4).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(5).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(6).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(7).setResizable
    (false);
    tablaInventario.getColumnModel().getColumn(8).setResizable
    (false);
}

jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/iphone-5.
png"))); // NOI18N

jLabel15.setFont(new java.awt.Font("DejaVu Serif", 3, 18));
// NOI18N
jLabel15.setText("SFH");
```

```
javax.swing.GroupLayout panelImagenLayout = new javax.swing.GroupLayout(panelImagen);
panelImagen.setLayout(panelImagenLayout);
panelImagenLayout.setHorizontalGroup(
    panelImagenLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelImagenLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel13)
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
        .addComponent(jLabel15, javax.swing.GroupLayout.
        DEFAULT_SIZE, 54, Short.MAX_VALUE)
        .addGap(47, 47, 47))
);
panelImagenLayout.setVerticalGroup(
    panelImagenLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelImagenLayout.createSequentialGroup()
        .addGroup(panelImagenLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel13, javax.swing.GroupLayout.DEFAULT_SIZE, 44, Short.MAX_VALUE
            .addComponent(jLabel15, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLa
            Short.MAX_VALUE))
        .addContainerGap())
);

panelCambios.setBackground(new java.awt.Color(120, 130, 134));

btnAgregar.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
btnAgregar.setText("Agregar ");
btnAgregar.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnAgregarActionPerformed(evt);
    }
});

btnModificar.setFont(new java.awt.Font("DejaVu Serif", 3,
15)); // NOI18N
btnModificar.setText("Modificar ");
btnModificar.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnModificarActionPerformed(evt);
    }
});
```

```
btnEliminar.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
btnEliminar.setText("Eliminar ");
btnEliminar.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnEliminarActionPerformed(evt);
    }
});

imagenEliminar.setIcon(new javax.swing.ImageIcon(getClass().getResource
("/Imagen/contenedor-de-basura.png"))); // NOI18N

imagenAgregar.setIcon(new javax.swing.ImageIcon(getClass().getResource
("/Imagen/signo-de-mas.png"))); // NOI18N

imagenModificar.setIcon(new javax.swing.ImageIcon(getClass().getResource
("/Imagen/libretas.png"))); // NOI18N

btnAtrasinventario.setFont(new java.awt.Font
("DejaVu Serif", 3, 15)); // NOI18N
btnAtrasinventario.setText("Regresar");
btnAtrasinventario.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnAtrasinventarioActionPerformed(evt);
    }
});

jLabel28.setIcon(new javax.swing.ImageIcon(getClass().getResource
("/Imagen/regreso.png"))); // NOI18N

javax.swing.GroupLayout panelCambiosLayout = new javax.swing.GroupLayout(panelCambios);
panelCambios.setLayout(panelCambiosLayout);
panelCambiosLayout.setHorizontalGroup(
    panelCambiosLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelCambiosLayout.createSequential
        .addGap(73, 73, 73)
        .addGroup(panelCambiosLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(btnAtrasinventario, javax.swing.GroupLayout.DEFAULT_SIZE, 115, Shor
            .addComponent(jLabel28, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLa
            Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED, 181, Short.MAX_VALUE)
        .addGroup(panelCambiosLayout.createParallelGroup
```

```
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swin
            .addGroup(panelCambiosLayout.createSequentialGroup()
                .addGap(19, 19, 19)
                .addComponent(imagenAgregar)))
        .addGap(156, 156, 156)
        .addGroup(panelCambiosLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnModificar, javax.swing.GroupLayout.PREFERRED_SIZE, 126, javax.sw
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelCambiosLayout.createSe
                .addComponent(imagenModificar)
                .addGap(31, 31, 31)))
        .addGap(172, 172, 172)
        .addGroup(panelCambiosLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnEliminar, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swi
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelCambiosLayout.createSe
                .addComponent(imagenEliminar)
                .addGap(23, 23, 23)))
        .addGap(111, 111, 111))
);
panelCambiosLayout.setVerticalGroup(
    panelCambiosLayout.createParallelGroup(javax.swing
    .GroupLayout.Alignment.LEADING)
    .addGroup(panelCambiosLayout.createSequentialGroup()
        .addGroup(panelCambiosLayout.createParallelGroup(
        javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelCambiosLayout.createSequential
            Group()
                .addContainerGap(javax.swing.GroupLayout.
                DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(panelCambiosLayout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.TRAILING,
                false)
                    .addComponent(imagenEliminar, javax.swing.GroupLayout.DEFAULT_SIZE, javax
                    .addComponent(imagenModificar, javax.swing.GroupLayout.DEFAULT_SIZE, java
                    .addComponent(imagenAgregar, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
            .addGroup(panelCambiosLayout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(jLabel28, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.Gro
        .addPreferredGap(javax.swing.LayoutStyle.Component
        Placement.RELATED)
        .addGroup(panelCambiosLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnAgregar)
            .addComponent(btnModificar)
            .addComponent(btnEliminar)
            .addComponent(btnAtrasinventario))
        .addGap(21, 21, 21))
```

```
);

txtBuscar.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N
txtBuscar.setBorder(null);
txtBuscar.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtBuscarActionPerformed(evt);
    }
});
txtBuscar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBuscarKeyReleased(evt);
    }
});

lblBuscador.setIcon(new javax.swing.ImageIcon(getClass().
getResource
("/Imagen/buscador.png"))); // NOI18N

javax.swing.GroupLayout panelInventarioLayout = new javax.swing.GroupLayout(panelInventario);
panelInventario.setLayout(panelInventarioLayout);
panelInventarioLayout.setHorizontalGroup(
    panelInventarioLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelInventarioLayout.createSequentialGroup()
        .addGroup(panelInventarioLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelInventarioLayout.
            createSequentialGroup()
                .addComponent(panelImagen, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swin
                .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_V
                .addComponent(lblBuscador)
                .addPreferredGap(javax.swing.LayoutStyle
                .ComponentPlacement.RELATED)
                .addComponent(txtBuscar, javax.swing.GroupLayout.PREFERRED_SIZE, 140, javax.s
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelInventarioLayout.creat
                .addContainerGap(28, Short.MAX_VALUE)
                .addComponent(panelCambios, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swi
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelInventarioLayout.creat
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(scrollpaneInventario, javax.swing.GroupLayout.PREFERRED_SIZE, 1
        .addContainerGap())
);
panelInventarioLayout.setVerticalGroup(
    panelInventarioLayout.createParallelGroup(javax.swing.
```

```
    GroupLayout.Alignment.LEADING)
    .addGroup(panelInventarioLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(panelInventarioLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(panelImagen, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.Gr
            .addComponent(txtBuscar, javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swing.
            .addComponent(lblBuscador, javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swin
        .addPreferredGap(javax.swing.LayoutStyle
        .ComponentPlacement.UNRELATED)
        .addComponent(scrollpaneInventario, javax.swing.GroupLayout.DEFAULT_SIZE, 253,
        Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
        .addComponent(panelCambios, javax.swing.GroupLayout.PREFERRED_SIZE, 134, javax.swing.
        .addGap(65, 65, 65))
);

javax.swing.GroupLayout InventarioLayout = new javax.swing.GroupLayout(Inventario.getContentP
Inventario.getContentPane().setLayout(InventarioLayout);
InventarioLayout.setHorizontalGroup(
    InventarioLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(InventarioLayout.createSequentialGroup()
        .addComponent(panelInventario, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.Gr
        .addGap(0, 37, Short.MAX_VALUE))
);
InventarioLayout.setVerticalGroup(
    InventarioLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(InventarioLayout.createSequentialGroup()
        .addComponent(panelInventario, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.Grou
        .addContainerGap())
);

btnCobrar.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
btnCobrar.setText("Cobrar");
btnCobrar.addActionListener(new java.awt.event.ActionListener
() {
    public void actionPerformed(java.awt.event.ActionEvent ev
    t) {
        btnCobrarActionPerformed(evt);
    }
});

lblTotal.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
lblTotal.setText("Total:");
```

```
txtTotal.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N
txtTotal.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtTotalKeyTyped(evt);
    }
});

btnCancelarproductor.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
btnCancelarproductor.setText("Cancelar producto");
btnCancelarproductor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btnCancelarproductorActionPerformed(evt);
    }
});

panelColor.setBackground(new java.awt.Color(0, 204, 255));

letrero2.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N

hora2.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N

javax.swing.GroupLayout panelColorLayout = new javax.swing.GroupLayout(panelColor);
panelColor.setLayout(panelColorLayout);
panelColorLayout.setHorizontalGroup(
    panelColorLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelColorLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(letrero2, javax.swing.GroupLayout.PREFERRED_SIZE, 1159, javax.swing.Gro
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(hora2, javax.swing.GroupLayout.
        PREFERRED_SIZE, 92, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(81, 81, 81))
);
panelColorLayout.setVerticalGroup(
    panelColorLayout.createParallelGroup(javax.swing
    .GroupLayout.Alignment.TRAILING)
    .addGroup(panelColorLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(panelColorLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(hora2, javax.swing.GroupLayout.DEFAULT_SIZE, 19, Short.MAX_VALUE)
            .addComponent(letrero2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLa
```

```
            Short.MAX_VALUE)))
);

logo.setIcon(new javax.swing.ImageIcon(getClass().getResource
("/Imagen/iphone-5.png"))); // NOI18N

lblCodigo.setFont(new java.awt.Font
("DejaVu Serif", 3, 15)); // NOI18N
lblCodigo.setText("Código del producto:");

txtCodigo.setFont(new java.awt.Font
("DejaVu Serif", 3, 14)); // NOI18N
txtCodigo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtCodigoKeyReleased(evt);
    }
});

lblCantidad.setFont(new java.awt.Font
("DejaVu Serif", 3, 15)); // NOI18N
lblCantidad.setText("Cantidad:");

atrasBtn.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
atrasBtn.setText("Regresar");
atrasBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        atrasBtnActionPerformed(evt);
    }
});

letrasLogo.setFont(new java.awt.Font("DejaVu Serif", 3, 18));
// NOI18N
letrasLogo.setText("SFH");

tablaInventario2.setFont(new java.awt.Font
("DejaVu Serif", 3, 14)); // NOI18N
tablaInventario2.setModel(new javax.swing.table.
DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Modelo", "Marca", "Precio de venta", "Cantidad"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };
```

```
    public boolean isCellEditable(int rowIndex, int
    columnIndex) {
        return canEdit [columnIndex];
    }
});
tablaInventario2.addMouseListener(new java.awt.event.
MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tablaInventario2MouseClicked(evt);
    }
});
scrollpaneInventario2.setViewportView(tablaInventario2);
if (tablaInventario2.getColumnModel().getColumnCount() > 0) {
    tablaInventario2.getColumnModel().getColumn(3).setResizable
    (false);
}

jToolBar1.setRollover(true);

inventario1.setFont(new java.awt.Font("DejaVu Serif", 3, 12));
// NOI18N
inventario1.setForeground(new java.awt.Color(0, 153, 204));
inventario1.setText("Inventario");
inventario1.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        inventario1ActionPerformed(evt);
    }
});
jToolBar1.add(inventario1);

historialv.setFont(new java.awt.Font("DejaVu Serif", 3, 12));
// NOI18N
historialv.setForeground(new java.awt.Color(0, 102, 153));
historialv.setText("Historial de venta");
historialv.setFocusable(false);
historialv.setHorizontalTextPosition(javax.swing.
SwingConstants.CENTER);
historialv.setVerticalTextPosition(javax.swing.
SwingConstants.BOTTOM);
historialv.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        historialvActionPerformed(evt);
    }
});
```

```
jToolBar1.add(historialv);

Tablapre_venta1.setModel(new javax.swing.table.
DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Producto", "Precio  de venta", "Cantidad"
    }
));
jScrollPane4.setViewportView(Tablapre_venta1);

cp1.setModel(new javax.swing.SpinnerNumberModel(0, 0, 150, 1));

javax.swing.GroupLayout panelEmpleadoLayout = new javax.swing.GroupLayout(panelEmpleado);
panelEmpleado.setLayout(panelEmpleadoLayout);
panelEmpleadoLayout.setHorizontalGroup(
    panelEmpleadoLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelEmpleadoLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(panelEmpleadoLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelEmpleadoLayout.
            createSequentialGroup()
                .addComponent(logo, javax.swing.GroupLayout.PREFERRED_SIZE, 41, javax.swing.G
                .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.UNRELATED)
                .addComponent(letrasLogo, javax.swing.GroupLayout.PREFERRED_SIZE, 45, javax.s
                .addGap(96, 96, 96)
                .addComponent(jToolBar1, javax.swing.GroupLayout.PREFERRED_SIZE, 268, javax.s
                .addContainerGap(javax.swing.GroupLayout.D
                EFAULT_SIZE, Short.MAX_VALUE))
            .addComponent(panelColor, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.Group
            .addGroup(panelEmpleadoLayout.createSequentialGroup()
                .addGroup(panelEmpleadoLayout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(panelEmpleadoLayout.
                    createSequentialGroup()
                        .addGap(22, 22, 22)
                        .addComponent(lblCodigo)
                        .addGap(18, 18, 18)
                        .addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
                        .addPreferredGap(javax.swing.
                        LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(lblCantidad)
                        .addPreferredGap(javax.swing.
                        LayoutStyle.ComponentPlacement.RELATED)
```

```
                        .addComponent(cp1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swin
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(javax.swing.GroupLayout.
                    Alignment.LEADING, panelEmpleadoLayout.createSequentialGroup()
                        .addGap(10, 10, 10)
                        .addGroup(panelEmpleadoLayout
                        .createParallelGroup(javax.swing.
                        GroupLayout.Alignment.LEADING)
                            .addGroup(panelEmpleadoLayout
                            .createSequentialGroup()
                                .addComponent(atrasBtn)
                                .addGap(0, 0, Short.MAX_VALUE))
                            .addComponent(scrollpaneInventario2))))
                .addGap(47, 47, 47)
                .addGroup(panelEmpleadoLayout.createParallel
                Group(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(panelEmpleadoLayout.
                    createSequentialGroup()
                        .addComponent(jScrollPane4, javax.swing.GroupLayout.
                        PREFERRED_SIZE, 493, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(45, 45, 45)
                        .addGroup(panelEmpleadoLayout.
                        createParallelGroup(javax.swing.
                        GroupLayout.Alignment.LEADING)
                            .addComponent(lblTotal, javax.swing.GroupLayout.
                            PREFERRED_SIZE, 76, javax.swing.GroupLayout.
                            PREFERRED_SIZE)
                            .addComponent(txtTotal, javax.swing.GroupLayout.
                            PREFERRED_SIZE, 90, javax.swing.GroupLayout.
                            PREFERRED_SIZE)
                            .addComponent(btnCobrar))
                        .addGap(38, 38, 38))
                    .addGroup(javax.swing.GroupLayout
                    .Alignment.TRAILING, panelEmpleadoLayout.createSequentialGroup()
                        .addComponent(btnCancelarproductor)
                        .addGap(347, 347, 347))))))
);
panelEmpleadoLayout.setVerticalGroup(
    panelEmpleadoLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelEmpleadoLayout.createSequentialGroup()
        .addGroup(panelEmpleadoLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelEmpleadoLayout.
            createSequentialGroup()
                .addGroup(panelEmpleadoLayout.
                createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)
                    .addGroup(panelEmpleadoLayout.
```

```
            createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(logo, javax.swing.GroupLayout.PREFERRED_SIZE, 31, javax
            .addGroup(panelEmpleadoLayout.
            createSequentialGroup()
                .addContainerGap()
                .addComponent(letrasLogo, javax.swing.GroupLayout.PREFERRE
                D_SIZE, 48, javax.swing.GroupLayout.PREFERR
                ED_SIZE)))
        .addGap(0, 27, Short.MAX_VALUE))
    .addGroup(panelEmpleadoLayout.createSequential
    Group()
        .addGap(20, 20, 20)
        .addComponent(jToolBar1, javax.swing.GroupLayout.PREFERRED_SIZE, 25, javax.sw
        .addPreferredGap(javax.swing.LayoutStyle.Compone

        ntPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
.addComponent(panelColor, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLa
.addGap(30, 30, 30)
.addGroup(panelEmpleadoLayout.createParallelGroup(javax.
swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblCodigo)
    .addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.Grou
    .addComponent(lblCantidad)
    .addComponent(cp1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayou
.addGroup(panelEmpleadoLayout.createParallelGroup(javax
.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelEmpleadoLayout.createSequentialGrou
    p()
        .addGap(23, 23, 23)
        .addComponent(lblTotal)
        .addPreferredGap(javax.swing.LayoutStyle.Compon
        entPlacement.RELATED)
        .addComponent(txtTotal, javax.swing.GroupLayout.PREFERRED_SIZE, 25, javax.swi
        .addPreferredGap(javax.swing.LayoutStyle.Compon
        entPlacement.UNRELATED)
        .addComponent(btnCobrar))
    .addGroup(panelEmpleadoLayout.createSequentialGroup()
        .addGap(15, 15, 15)
        .addGroup(panelEmpleadoLayout.createParallelGr
        oup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 233,
            .addGroup(panelEmpleadoLayout.createSeque
            ntialGroup()
                .addGap(3, 3, 3)
                .addComponent(scrollpaneInventario2, javax.swing.GroupLayout.PREFERRE
        .addPreferredGap(javax.swing.LayoutStyle.Comp
        onentPlacement.UNRELATED)
        .addGroup(panelEmpleadoLayout.createParallelGro
```

```
                up(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnCancelarproductor)
                    .addComponent(atrasBtn))))
        .addGap(5, 5, 5))
);

javax.swing.GroupLayout TablaEmpleadoLayout = new javax.swing.GroupLayout(TablaEmpleado.getCo
TablaEmpleado.getContentPane().setLayout(TablaEmpleadoLayout);
TablaEmpleadoLayout.setHorizontalGroup(
    TablaEmpleadoLayout.createParallelGroup(javax.swing.Group
    Layout.Alignment.LEADING)
    .addComponent(panelEmpleado, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayou
);
TablaEmpleadoLayout.setVerticalGroup(
    TablaEmpleadoLayout.createParallelGroup(javax.swing.Group
    Layout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, TablaEmpleadoLayout.createSequentia
        .addContainerGap()
        .addComponent(panelEmpleado, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupL
);

colorPanel.setBackground(new java.awt.Color(102, 204, 255));

letrero.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N

relog.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N

javax.swing.GroupLayout colorPanelLayout = new javax.swing.GroupLayout(colorPanel);
colorPanel.setLayout(colorPanelLayout);
colorPanelLayout.setHorizontalGroup(
    colorPanelLayout.createParallelGroup(javax.swing.GroupLa
    yout.Alignment.LEADING)
    .addGroup(colorPanelLayout.createSequentialGroup()
        .addComponent(letrero, javax.swing.GroupLayout.PREFERRED_SIZE, 1139, javax.swing.Grou
        .addPreferredGap(javax.swing.LayoutStyle.Component
        Placement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(relog, javax.swing.GroupLayout.
        PREFERRED_SIZE, 94, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(42, 42, 42))
);
colorPanelLayout.setVerticalGroup(
    colorPanelLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(colorPanelLayout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(letrero, javax.swing.GroupLayout
        .DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
        .addComponent(relog, javax.swing.GroupLayout.P
        REFERRED_SIZE, 19, javax.swing.GroupLayout.PREFERRED_SIZE))
);

jLabel21.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/
iphone-5.png"))); // NOI18N


jLabel22.setFont(new java.awt.Font("DejaVu Serif", 3, 18));
// NOI18N
jLabel22.setText("SFH");

javax.swing.GroupLayout imagenPanelLayout = new javax.swing.GroupLayout(imagenPanel);
imagenPanel.setLayout(imagenPanelLayout);
imagenPanelLayout.setHorizontalGroup(
    imagenPanelLayout.createParallelGroup(javax.swing.G
    roupLayout.Alignment.LEADING)
    .addGroup(imagenPanelLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel21, javax.swing.GroupLayout.PREFERRED_SIZE, 42, javax.swing.Group
        .addPreferredGap(javax.swing.LayoutStyle.Compo
        nentPlacement.RELATED)
        .addComponent(jLabel22, javax.swing.GroupLayout.PREFERRED_SIZE, 49, javax.swing.Group
        .addContainerGap(javax.swing.GroupLayout.DEFAUL
        T_SIZE, Short.MAX_VALUE))
);
imagenPanelLayout.setVerticalGroup(
    imagenPanelLayout.createParallelGroup(javax.swing.G
    roupLayout.Alignment.LEADING)
    .addComponent(jLabel21, javax.swing.GroupLayout.DEF
    AULT_SIZE, 42, Short.MAX_VALUE)
    .addComponent(jLabel22, javax.swing.GroupLayout.DEF
    AULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short
    .MAX_VALUE)
);

panelProducto.setBackground(new java.awt.Color(255, 255, 255));

jLabel25.setFont(new java.awt.Font("DejaVu Serif", 3, 15
)); // NOI18N
jLabel25.setText("Codigo de producto:");

tfCodigo.setFont(new java.awt.Font("DejaVu Serif", 3, 14
)); // NOI18N
tfCodigo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tfCodigoActionPerformed(evt);
    }
});
```

```
tfCodigo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        tfCodigoKeyReleased(evt);
    }
});

jLabel26.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel26.setText("Cantidad:");

pc.setModel(new javax.swing.SpinnerNumberModel(0, 0, 50, 1));

javax.swing.GroupLayout panelProductoLayout = new javax.swing.GroupLayout(panelProducto);
panelProducto.setLayout(panelProductoLayout);
panelProductoLayout.setHorizontalGroup(
    panelProductoLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelProductoLayout.createSequentialGroup()
        .addGap(26, 26, 26)
        .addComponent(jLabel25)
        .addGap(2, 2, 2)
        .addComponent(tfCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 130, javax.swing.Grou
        .addPreferredGap(javax.swing.LayoutStyle.C
        omponentPlacement.RELATED)
        .addComponent(jLabel26)
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
        .addComponent(pc, javax.swing.GroupLayout.
        PREFERRED_SIZE, 51, javax.swing.GroupLayout.P
        REFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
);
panelProductoLayout.setVerticalGroup(
    panelProductoLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelProductoLayout.createSequentia
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(panelProductoLayout.createParallelGrou
        (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel25)
            .addComponent(tfCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.Group
            .addComponent(jLabel26)
            .addComponent(pc, javax.swing.GroupLayout.
            PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PRE
        .addContainerGap())
);

botonCobrar.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
botonCobrar.setText("Cobrar");
```

```
botonCobrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botonCobrarActionPerformed(evt);
    }
});

botonEliminarp.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
botonEliminarp.setText("Eliminar producto");
botonEliminarp.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botonEliminarpActionPerformed(evt);
    }
});

textCobrar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        textCobrarKeyTyped(evt);
    }
});

labelTotal.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
labelTotal.setText("Total:");

tablaInventario1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
tablaInventario1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Modelo", "Marca", "Precio de venta", "Cantidad"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex)
    {
        return canEdit [columnIndex];
    }
});
tablaInventario1.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tablaInventario1MouseClicked(evt);
    }
});
scrollpaneInventario1.setViewportView(tablaInventario1);
if (tablaInventario1.getColumnModel().getColumnCount() > 0) {
```

```
    tablaInventario1.getColumnModel().getColumn(0).setResizable
    (false);
    tablaInventario1.getColumnModel().getColumn(1).setResizable
    false);
    tablaInventario1.getColumnModel().getColumn(2).setResizable
    (false);
    tablaInventario1.getColumnModel().getColumn(3).setResizable
    (false);
}

Tablapre_venta.setModel(new javax.swing.table.DefaultTableModel
(
    new Object [][] {

    },
    new String [] {
        "Producto", "Precio  de venta", "Cantidad"
    }
));
jScrollPane3.setViewportView(Tablapre_venta);

javax.swing.GroupLayout panelAdministradorLayout = new javax.swing.GroupLayout(panelAdministr
panelAdministrador.setLayout(panelAdministradorLayout);
panelAdministradorLayout.setHorizontalGroup(
    panelAdministradorLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelAdministradorLayout.createSequentialGroup()
        .addGroup(panelAdministradorLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelAdministradorLayout.createSequenti
            alGroup()
                .addContainerGap()
                .addGroup(panelAdministradorLayout.createPar
                allelGroup(javax.swing.GroupLayout.Alignment
                .LEADING)
                    .addComponent(colorPanel, javax.swing.GroupLayout.Alignment.TRAILING, jav
                    .addComponent(panelProducto, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
                    .addGroup(panelAdministradorLayout.creat
                    eSequentialGroup()
                        .addGap(12, 12, 12)
                        .addComponent(scrollpaneInventario1, javax.swing.GroupLayout.PREFERRE
                        E, 534, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutSty
                        le.ComponentPlacement.RELATED, 103, Short.MAX_VALUE)
                        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZ
                        E, 499, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGroup(panelAdministradorLayout.
                        createParallelGroup(javax.swing.
                        GroupLayout.Alignment.LEADING)
```

```
                                .addGroup(panelAdministradorLayout.
                                createSequentialGroup()
                                    .addPreferredGap(javax.swing.
                                    LayoutStyle.ComponentPlacement.
                                    RELATED, javax.swing.GroupLayout.DEFAULT
                                    _SIZE, Short.MAX_VALUE)
                                    .addGroup(panelAdministrador
                                    Layout.
                                    createParallelGroup(javax.swing
                                    .GroupLayout.Alignment.LEADING)
                                        .addComponent(botonCobrar, javax.swing.GroupLayout.DEFAUL
                                        .addComponent(textCobrar)))
                                .addGroup(javax.swing.GroupLayout.
                                Alignment.TRAILING, panelAdministradorLayout.create
                                SequentialGroup()
                                    .addGap(86, 86, 86)
                                    .addComponent(labelTotal, javax.swing.GroupLayout.PREFERRED
                                    _SIZE, 60, javax.swing.GroupLayout.PREFERRED_
                                    SIZE)
                                    .addGap(25, 25, 25))))))
            .addGroup(panelAdministradorLayout.createSequential
            Group()
                .addGap(24, 24, 24)
                .addComponent(imagenPanel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swin
                .addPreferredGap(javax.swing.LayoutStyle
                ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_V
        .addContainerGap())
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelAdministradorLayout.createSequ
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(botonEliminarp)
        .addGap(372, 372, 372))
);
panelAdministradorLayout.setVerticalGroup(
    panelAdministradorLayout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelAdministradorLayout.createSequential
    Group()
        .addGap(10, 10, 10)
        .addComponent(imagenPanel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupL
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
        .addComponent(colorPanel, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLa
        .addPreferredGap(javax.swing.LayoutStyle.C
        omponentPlacement.RELATED)
        .addComponent(panelProducto, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.Grou
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
        .addGroup(panelAdministradorLayout.
        createParallelGroup(javax.swing.GroupLayout.
```

```
        Alignment.LEADING)
            .addGroup(panelAdministradorLayout.
            createSequentialGroup()
                .addGroup(panelAdministradorLayout.
                createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)
                    .addGroup(panelAdministradorLayout.
                    createSequentialGroup()
                        .addComponent(labelTotal, javax.swing.GroupLayout.
                        PREFERRED_SIZE, 17, javax.swing.
                        GroupLayout.
                        PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.
                        LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(textCobrar, javax.swing.GroupLayout.PREFERRED_SIZE, jav
                        .addGap(24, 24, 24)
                        .addComponent(botonCobrar, javax.swing.GroupLayout.PREFERRED_SIZE,
                        37, javax.swing.GroupLayout.
                        PREFERRED_SIZE))
                    .addComponent(scrollpaneInventario1, javax.swing.GroupLayout.PREFERRED_SI
                    227, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(0, 47, Short.MAX_VALUE))
            .addGroup(panelAdministradorLayout.
            createSequentialGroup()
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE,
                218, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.
                LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
                .addComponent(botonEliminarp)))
        .addContainerGap())
);

menuAtras.setText("Regresar");
menuAtras.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.A
    ctionEvent evt) {
        menuAtrasActionPerformed(evt);
    }
});

jMenuItem3.setFont(new java.awt.Font("DejaVu Serif",
3, 12)); // NOI18N
jMenuItem3.setText("Aceptar");
jMenuItem3.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
```

```
    }
});
menuAtras.add(jMenuItem3);

jMenuBar1.add(menuAtras);

menuPermisos.setText("Opciones");
menuPermisos.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.A
    ctionEvent evt) {
        menuPermisosActionPerformed(evt);
    }
});

itemInventario.setFont(new java.awt.Font("DejaVu Serif",
3, 12)); // NOI18N
itemInventario.setText("Inventario");
itemInventario.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        itemInventarioActionPerformed(evt);
    }
});
menuPermisos.add(itemInventario);

itemUsuarios.setFont(new java.awt.Font("DejaVu Ser
if", 3, 12)); // NOI18N
itemUsuarios.setText("Usuarios");
itemUsuarios.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.Ac
    tionEvent evt) {
        itemUsuariosActionPerformed(evt);
    }
});
menuPermisos.add(itemUsuarios);

itemReporte.setFont(new java.awt.Font("DejaVu S
erif", 3, 12)); // NOI18N
itemReporte.setText("Reporte");
itemReporte.addActionListener(new java.awt.event
.ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        itemReporteActionPerformed(evt);
    }
});
```

```
menuPermisos.add(itemReporte);

jMenuBar1.add(menuPermisos);

TablaAdministrador.setJMenuBar(jMenuBar1);

javax.swing.GroupLayout TablaAdministradorLayout
= new javax.swing.GroupLayout(TablaAdministrador.
getContentPane());
TablaAdministrador.getContentPane().setLayout(
TablaAdministradorLayout);
TablaAdministradorLayout.setHorizontalGroup(
    TablaAdministradorLayout.createParallelGroup
    javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(TablaAdministradorLayout.createSe
    quentialGroup()
        .addContainerGap()
        .addComponent(panelAdministrador, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.G
        Short.MAX_VALUE)
        .addContainerGap())
);
TablaAdministradorLayout.setVerticalGroup(
    TablaAdministradorLayout.createParallelGroup
    (javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(TablaAdministradorLayout.
    createSequentialGroup()
        .addContainerGap()
        .addComponent(panelAdministrador, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.G
        Short.MAX_VALUE)
        .addContainerGap())
);

panelCobrar.setBackground(new java.awt.Color
(255, 255, 255));
panelCobrar.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

btnCancelarp.setFont(new java.awt.Font
("DejaVu Serif", 3, 15)); // NOI18N
btnCancelarp.setText("Cancelar");
btnCancelarp.addActionListener(new java.awt.event
.ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnCancelarpActionPerformed(evt);
    }
});
panelCobrar.add(btnCancelarp, new org.netbeans.lib.awtextra.AbsoluteConstraints
(307, 264, -1, -1));
```

```
totalLbl.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
totalLbl.setText("Total:");
panelCobrar.add(totalLbl, new org.netbeans.lib.awtextra.AbsoluteConstraints(350,
40, 57, -1));

txtRecibido.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
txtRecibido.setBorder(null);
txtRecibido.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        txtRecibidoActionPerformed(evt);
    }
});
txtRecibido.addKeyListener(new java.awt
.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtRecibidoKeyTyped(evt);
    }
});
panelCobrar.add(txtRecibido, new org.netbeans.lib.awtextra.AbsoluteConstraints
(410, 80, 143, -1));

lblRecibido.setFont(new java.awt.Font("DejaVu
Serif", 3, 15)); // NOI18N
lblRecibido.setText("Recibido:");
panelCobrar.add(lblRecibido, new org.netbeans.lib.awtextra.AbsoluteConstraints
(320, 80, -1, -1));

btnCobrarp.setFont(new java.awt.Font("DejaVu
Serif", 3, 15)); // NOI18N
btnCobrarp.setText("Cobrar");
btnCobrarp.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnCobrarpActionPerformed(evt);
    }
});
panelCobrar.add(btnCobrarp, new org.netbeans.lib.awtextra.AbsoluteConstraints
(468, 264, 97, -1));

txtTotalp.setFont(new java.awt.Font
("DejaVu Serif", 3, 15)); // NOI18N
txtTotalp.setBorder(null);
txtTotalp.addKeyListener(new java.awt.event.
KeyAdapter() {
```

```
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtTotalpKeyTyped(evt);
    }
});
panelCobrar.add(txtTotalp, new org.netbeans.lib.awtextra.AbsoluteConstraints
(405, 38, 143, -1));

panelProductos.setBackground(new java.awt.Color
(89, 182, 183));
panelProductos.setForeground(new java.awt.Color
(1, 1, 1));

jLabel2.setFont(new java.awt.Font("DejaVu Serif"
, 3, 15)); // NOI18N
jLabel2.setText("No. Productos:");

jLabel30.setIcon(new javax.swing.ImageIcon(getClass()
.getResource
("/Imagen/transferencia-de-dinero.png"))); // NOI18N

NoP.setBackground(new java.awt.Color(89, 182, 183));
NoP.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N
NoP.setBorder(null);

fd.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N

javax.swing.GroupLayout panelProductosLayout = new javax.swing.GroupLayout(panelProductos);
panelProductos.setLayout(panelProductosLayout);
panelProductosLayout.setHorizontalGroup(
    panelProductosLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(panelProductosLayout.createSequentialGroup()
        .addGroup(panelProductosLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelProductosLayout.createSequential
            Group()
                .addGap(66, 66, 66)
                .addComponent(jLabel2))
            .addGroup(panelProductosLayout.createSequential
            Group()
                .addGap(94, 94, 94)
                .addGroup(panelProductosLayout.createParall
                elGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel30, javax.swing.GroupLayout.PREFERRED_SIZE,
                    89, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(panelProductosLayout.createSeq
                    uentialGroup()
```

```
                        .addGap(21, 21, 21)
                        .addComponent(NoP, javax.swing.GroupLayout.PREFERRED_SIZE, 30, javax.
            .addGroup(panelProductosLayout.createSequentialGroup()
                .addGap(77, 77, 77)
                .addComponent(fd, javax.swing.GroupLayout.PREFERRED_SIZE, 97, javax.swing.Gro
        .addContainerGap(83, Short.MAX_VALUE))
);
panelProductosLayout.setVerticalGroup(
    panelProductosLayout.createParallelGroup(javax.swing.G
    roupLayout.Alignment.LEADING)
    .addGroup(panelProductosLayout.createSequentialGroup()
        .addGap(34, 34, 34)
        .addComponent(jLabel2)
        .addGap(13, 13, 13)
        .addComponent(NoP, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
        .addGap(18, 18, 18)
        .addComponent(jLabel30, javax.swing.GroupLayout.PREFERRED_SIZE, 84, javax.swing.Group
        .addGap(18, 18, 18)
        .addComponent(fd, javax.swing.GroupLayout.PREFERRED_S
        IZE, 32, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

panelCobrar.add(panelProductos, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 0, -1,
panelCobrar.add(jSeparator8, new org.netbeans.lib.awtextra.AbsoluteConstraints(405, 60, 143,
panelCobrar.add(jSeparator9, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 110, -1,
panelCobrar.add(jSeparator11, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 100, 140

mensajeReci.setFont(new java.awt.Font("DejaVu Serif", 3, 12)); // NOI18N
mensajeReci.setForeground(new java.awt.Color(0, 102, 204));
panelCobrar.add(mensajeReci, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 150, 140,

javax.swing.GroupLayout CobrarLayout = new javax.swing.GroupLayout(Cobrar.getContentPane());
Cobrar.getContentPane().setLayout(CobrarLayout);
CobrarLayout.setHorizontalGroup(
    CobrarLayout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
    .addComponent(panelCobrar, javax.swing.GroupLayout.DEFA
    ULT_SIZE, 586, Short.MAX_VALUE)
);
CobrarLayout.setVerticalGroup(
    CobrarLayout.createParallelGroup(javax.swing.GroupLayout.
    Alignment.LEADING)
    .addComponent(panelCobrar, javax.swing.GroupLayout.DEFAU
    LT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.M
    AX_VALUE)
);

panelAgregarp.setBackground(new java.awt.Color(241, 255, 255));
```

```
panelAgregarp.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

lblCodigop.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblCodigop.setText("Modelo:");
panelAgregarp.add(lblCodigop, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 100,
-1, -1));

lblNombrep.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N

lblNombrep.setText("RAM:");
panelAgregarp.add(lblNombrep, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 160, 4
9, 25));

lblPreciov.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
lblPreciov.setText("Bateria:");
panelAgregarp.add(lblPreciov, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 220,
-1, -1));

lblPrecioc.setFont(new java.awt.Font("DejaVu Serif", 3, 15
)); // NOI18N
lblPrecioc.setText("Precio de venta:");
panelAgregarp.add(lblPrecioc, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 310,
-1, -1));

lblCantidadp.setFont(new java.awt.Font("DejaVu Serif", 3
, 15)); // NOI18N
lblCantidadp.setText("Precio de compra:");
panelAgregarp.add(lblCantidadp, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 280, -1, -1));

lblMarcap.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblMarcap.setText("Memoria interna:");
panelAgregarp.add(lblMarcap, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 190, -1, -

txtModelo.setBackground(new java.awt.Color(241, 255, 255));
txtModelo.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtModelo.setBorder(null);
txtModelo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtModeloKeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtModeloKeyTyped(evt);
    }
});
panelAgregarp.add(txtModelo, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 100,
154, -1));
```

```
txtMarca.setBackground(new java.awt.Color(241, 255, 255));
txtMarca.setFont(new java.awt.Font("DejaVu Serif", 3, 15
)); // NOI18N
txtMarca.setBorder(null);
txtMarca.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMarcaKeyTyped(evt);
    }
});
panelAgregarp.add(txtMarca, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 130, 1
54, -1));

txtRam.setBackground(new java.awt.Color(241, 255, 255));
txtRam.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtRam.setBorder(null);
txtRam.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtRamKeyTyped(evt);
    }
});
panelAgregarp.add(txtRam, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 160, 15
4, -1));

txtMemoria.setBackground(new java.awt.Color(241, 255, 255));
txtMemoria.setFont(new java.awt.Font("DejaVu Serif", 3, 1
5)); // NOI18N
txtMemoria.setBorder(null);
txtMemoria.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMemoriaKeyTyped(evt);
    }
});
panelAgregarp.add(txtMemoria, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 190,
154, -1));

txtBateria.setBackground(new java.awt.Color(241, 255, 255));
txtBateria.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtBateria.setBorder(null);
txtBateria.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBateriaKeyTyped(evt);
    }
});
panelAgregarp.add(txtBateria, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 220,
154, -1));

txtPixeles.setBackground(new java.awt.Color(241, 255, 255));
```

```
txtPixeles.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtPixeles.setBorder(null);
txtPixeles.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPixelesKeyTyped(evt);
    }
});
panelAgregarp.add(txtPixeles, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 250, 154
-1));

lblLetrero.setFont(new java.awt.Font("DejaVu Serif", 3, 24));
// NOI18N
lblLetrero.setText("Agregar producto");
panelAgregarp.add(lblLetrero, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 20, -1,

jLabel14.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); //
NOI18N
jLabel14.setText("Pixeles:");
panelAgregarp.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 250, -1, -

jLabel16.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); //
NOI18N
jLabel16.setText("Marca:");
panelAgregarp.add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 130, -1, -

jLabel17.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); //
NOI18N
jLabel17.setText("Cantidad:");
panelAgregarp.add(jLabel17, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 340, -1, -

txtPrecioc.setBackground(new java.awt.Color(241, 255, 255));
txtPrecioc.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtPrecioc.setBorder(null);
txtPrecioc.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtPreciocActionPerformed(evt);
    }
});
txtPrecioc.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPreciocKeyTyped(evt);
    }
});
panelAgregarp.add(txtPrecioc, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 310, 154
1));
```

```
txtPreciov.setBackground(new java.awt.Color(241, 255, 255));
txtPreciov.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtPreciov.setBorder(null);
txtPreciov.addActionListener(new java.awt.event.ActionListener(
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtPreciovActionPerformed(evt);
    }
});
txtPreciov.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPreciovKeyTyped(evt);
    }
});
panelAgregarp.add(txtPreciov, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 280, 154

txtCantidadm.setBackground(new java.awt.Color(241, 255, 255));
txtCantidadm.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtCantidadm.setBorder(null);
txtCantidadm.addActionListener(new java.awt.event.ActionListener
() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtCantidadmActionPerformed(evt);
    }
});
txtCantidadm.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCantidadmKeyTyped(evt);
    }
});
panelAgregarp.add(txtCantidadm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 340, 1
));

btnGuardarp.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
btnGuardarp.setText("Guardar");
btnGuardarp.addActionListener(new java.awt.event.ActionListen
er() {
    public void actionPerformed(java.awt.event.ActionEvent e
    vt) {
        btnGuardarpActionPerformed(evt);
    }
});
panelAgregarp.add(btnGuardarp, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 400, -1
1));
```

```
btnCamcelarpro.setFont(new java.awt.Font("DejaVu Serif", 3, 1
5)); // NOI18N
btnCamcelarpro.setText("Cancelar");
btnCamcelarpro.addActionListener(new java.awt.event.ActionLis
tener() {
    public void actionPerformed(java.awt.event.ActionEvent ev
    t) {
        btnCamcelarproActionPerformed(evt);
    }
});
panelAgregarp.add(btnCamcelarpro, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 400,

jLabel31.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/lapiz (4).png")));
panelAgregarp.add(jLabel31, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 60, 72, 125
panelAgregarp.add(jSeparator13, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 120, 1
panelAgregarp.add(jSeparator14, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 150, 1
panelAgregarp.add(jSeparator15, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 180, 1
panelAgregarp.add(jSeparator16, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 210, 1
panelAgregarp.add(jSeparator17, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 282, 1
panelAgregarp.add(jSeparator18, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 240, 1
panelAgregarp.add(jSeparator19, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 270, 1
panelAgregarp.add(jSeparator20, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 300, 1
panelAgregarp.add(jSeparator21, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 330, 1
panelAgregarp.add(jSeparator22, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 360, 1

javax.swing.GroupLayout AgregarPLayout = new javax.swing.GroupLayout(AgregarP.getContentPane(
AgregarP.getContentPane().setLayout(AgregarPLayout);
AgregarPLayout.setHorizontalGroup(
    AgregarPLayout.createParallelGroup(javax.swing.GroupLay
    out.Alignment.LEADING)
    .addComponent(panelAgregarp, javax.swing.GroupLayout.PREFERRED_SIZE, 401, javax.swing.Gro
);
AgregarPLayout.setVerticalGroup(
    AgregarPLayout.createParallelGroup(javax.swing.GroupLayo
    ut.Alignment.LEADING)
    .addComponent(panelAgregarp, javax.swing.GroupLayout.DEFAULT_SIZE, 443, Short.MAX_VALUE)
);

panelAgregarp1.setBackground(new java.awt.Color(241, 255
, 255));
panelAgregarp1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

lblNombrep1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblNombrep1.setText("RAM:");
panelAgregarp1.add(lblNombrep1, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 120, -

lblPreciov1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblPreciov1.setText("Bateria:");
panelAgregarp1.add(lblPreciov1, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 180, -
```

```
lblPrecioc1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblPrecioc1.setText("Precio de compra:");
panelAgregarp1.add(lblPrecioc1, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 270, -1

lblCantidadp1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblCantidadp1.setText("Precio de venta:");
panelAgregarp1.add(lblCantidadp1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 240,

lblMarcap1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
lblMarcap1.setText("Memoria interna:");
panelAgregarp1.add(lblMarcap1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 150, -1,

txtMarcam.setBackground(new java.awt.Color(241, 255, 255));
txtMarcam.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
txtMarcam.setBorder(null);
txtMarcam.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMarcamKeyTyped(evt);
    }
});
panelAgregarp1.add(txtMarcam, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 90, 154
, -1));

txtRamm.setBackground(new java.awt.Color(241, 255, 255));
txtRamm.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); /
/ NOI18N
txtRamm.setBorder(null);
txtRamm.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtRammKeyTyped(evt);
    }
});
panelAgregarp1.add(txtRamm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 120, 154,
-1));

txtMemoriam.setBackground(new java.awt.Color(241, 255, 255));
txtMemoriam.setFont(new java.awt.Font("DejaVu Serif", 3, 15
)); // NOI18N
txtMemoriam.setBorder(null);
txtMemoriam.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMemoriamKeyTyped(evt);
    }
});
panelAgregarp1.add(txtMemoriam, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 150, 1
, -1));

txtBateriam.setBackground(new java.awt.Color(241, 255, 255));
```

```
txtBateriam.setFont(new java.awt.Font("DejaVu Serif", 3, 1
5)); // NOI18N
txtBateriam.setBorder(null);
txtBateriam.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtBateriamKeyTyped(evt);
    }
});
panelAgregarp1.add(txtBateriam, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 180, 1
, 20));

txtPixelesm.setBackground(new java.awt.Color(241, 255, 255));
txtPixelesm.setFont(new java.awt.Font("DejaVu Serif", 3, 15
)); // NOI18N
txtPixelesm.setBorder(null);
txtPixelesm.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPixelesmKeyTyped(evt);
    }
});
panelAgregarp1.add(txtPixelesm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 210, 1
-1));

lblLetrero1.setFont(new java.awt.Font("DejaVu Serif", 3, 24)
); // NOI18N
lblLetrero1.setText("Modificar producto");
panelAgregarp1.add(lblLetrero1, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 20, -1
1));

jLabel18.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel18.setText("Pixeles:");
panelAgregarp1.add(jLabel18, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 210, -1,
-1));

jLabel19.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel19.setText("Marca:");
panelAgregarp1.add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 90, -1, -

jLabel20.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel20.setText("Cantidad:");
panelAgregarp1.add(jLabel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 300, -1,
-1));

txtcPreciovm.setEditable(false);
txtcPreciovm.setBackground(new java.awt.Color(241, 255, 255));
txtcPreciovm.setFont(new java.awt.Font("DejaVu Serif", 3, 15
```

```
)); // NOI18N
txtcPreciovm.setBorder(null);
txtcPreciovm.addActionListener(new java.awt.event.ActionListe
ner() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        txtcPreciovmActionPerformed(evt);
    }
});
txtcPreciovm.addKeyListener(new java.awt.event.KeyAdapter()
{
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtcPreciovmKeyTyped(evt);
    }
});
panelAgregarp1.add(txtcPreciovm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 240,
-1));

txtPreciocm.setBackground(new java.awt.Color(241, 255, 255));
txtPreciocm.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtPreciocm.setBorder(null);
txtPreciocm.addActionListener(new java.awt.event.ActionListene
r() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtPreciocmActionPerformed(evt);
    }
});
txtPreciocm.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtPreciocmKeyTyped(evt);
    }
});
panelAgregarp1.add(txtPreciocm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 270, 1
-1));

txtcantm.setBackground(new java.awt.Color(241, 255, 255));
txtcantm.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txtcantm.setBorder(null);
txtcantm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        txtcantmActionPerformed(evt);
    }
});
txtcantm.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
```

```
        txtcantmKeyTyped(evt);
    }
});
panelAgregarp1.add(txtcantm, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 300, 154,
-1));

btnGuardarp1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)
); // NOI18N
btnGuardarp1.setText("Guardar");
btnGuardarp1.addActionListener(new java.awt.event.ActionLis
tener
() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt)
    {
        btnGuardarp1ActionPerformed(evt);
    }
});
panelAgregarp1.add(btnGuardarp1, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 380,
1, -1));

btnCamcelarpro1.setFont(new java.awt.Font("DejaVu Serif",
3, 15));
// NOI18N
btnCamcelarpro1.setText("Cancelar");
btnCamcelarpro1.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnCamcelarpro1ActionPerformed(evt);
    }
});
panelAgregarp1.add(btnCamcelarpro1, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 38
-1, -1));

jLabel32.setIcon(new javax.swing.ImageIcon(getClass().
getResource
("/Imagen/escribiendo.png"))); // NOI18N
panelAgregarp1.add(jLabel32, new org.netbeans.lib.awtextra.AbsoluteConstraints
(40, 60, 88, 90));
panelAgregarp1.add(jSeparator12, new org.netbeans.lib.awtextra.AbsoluteConstraints
(210, 110, 154, 10));
panelAgregarp1.add(jSeparator23, new org.netbeans.lib.awtextra.AbsoluteConstraints
(210, 140, 154, 10));
panelAgregarp1.add(jSeparator24, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 170,
4, 10));
panelAgregarp1.add(jSeparator25, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 200,
60, 10));
panelAgregarp1.add(jSeparator26, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 248,
```

```
60, 0));
panelAgregarp1.add(jSeparator27, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 252,
40, 0));
panelAgregarp1.add(jSeparator28, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 230,
0, 10));
panelAgregarp1.add(jSeparator29, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 260, 1
60, 10));
panelAgregarp1.add(jSeparator30, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 300,
-1, -1));
panelAgregarp1.add(jSeparator31, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 290,
, 10));
panelAgregarp1.add(jSeparator32, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 322,
0, 30));

javax.swing.GroupLayout ModificarPLayout = new javax.swing.GroupLayout(ModificarP.getContentP
ModificarP.getContentPane().setLayout(ModificarPLayout);
ModificarPLayout.setHorizontalGroup(
    ModificarPLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addComponent(panelAgregarp1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayo
);
ModificarPLayout.setVerticalGroup(
    ModificarPLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addComponent(panelAgregarp1, javax.swing.GroupLayout.DEFAULT_SIZE, 421, Short.MAX_VALUE)
);

Usuario.setUndecorated(true);

panelUsuario.setBackground(new java.awt.Color(241, 255, 255));
panelUsuario.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

imagenUsuario.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/usuario (3).p
panelUsuario.add(imagenUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints
(90, 10, 147, -1));

aceptarAdministrador.setBackground(new java.awt.Color(153, 255, 153));
aceptarAdministrador.setFont(new java.awt.Font("DejaVu Serif
", 3, 15)); // NOI18N
aceptarAdministrador.setText("Aceptar");
aceptarAdministrador.setBorder(javax.swing.Borde
rFactory.createBevelBorder(javax.swing.border.BevelBorder.
RAISED));
aceptarAdministrador.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        aceptarAdministradorMouseClicked(evt);
    }
});
```

```
aceptarAdministrador.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.Actio
    nEvent evt) {
        aceptarAdministradorActionPerformed(evt);
    }
});
panelUsuario.add(aceptarAdministrador, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
1, -1));

txtNombreu.setBackground(new java.awt.Color(241, 255, 255
));
txtNombreu.setFont(new java.awt.Font("DejaVu Serif", 3, 1
2)); // NOI18N
txtNombreu.setText("Ingresar usuario.....");
txtNombreu.setBorder(null);
txtNombreu.addMouseListener(new java.awt.event.MouseAda
pter() {
    public void mouseClicked(java.awt.event.MouseEven
    t evt) {
        txtNombreuMouseClicked(evt);
    }
});
txtNombreu.addActionListener(new java.awt.event.ActionLis
tener() {
    public void actionPerformed(java.awt.event.ActionEven
    t evt) {
        txtNombreuActionPerformed(evt);
    }
});
txtNombreu.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNombreuKeyTyped(evt);
    }
});
panelUsuario.add(txtNombreu, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 170, 14
0, 30));

imgNombre.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/
usuario16px.png"))); // NOI18N
panelUsuario.add(imgNombre, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, -1
, 29));

imgContrasena.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/contr
asena16px.png"))); // NOI18N
panelUsuario.add(imgContrasena, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 250, -1, -1));

txtContrasenau.setBackground(new java.awt.Color(241, 255
, 255));
```

```
txtContrasenau.setFont(new java.awt.Font("DejaVu Serif
", 3, 8)); // NOI18N
txtContrasenau.setText("**********");
txtContrasenau.setBorder(null);
txtContrasenau.addMouseListener(new java.awt.event.Mou
seAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent e
    vt) {
        txtContrasenauMouseClicked(evt);
    }
});
txtContrasenau.addActionListener(new java.awt.event.Act
ionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        txtContrasenauActionPerformed(evt);
    }
});
txtContrasenau.addKeyListener(new java.awt.event.KeyAdapt
er() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtContrasenauKeyTyped(evt);
    }
});
panelUsuario.add(txtContrasenau, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 250,
130, 27));

lblNombreu.setFont(new java.awt.Font("DejaVu Serif", 3,
15)); // NOI18N
lblNombreu.setText("Nombre:");
panelUsuario.add(lblNombreu, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 150,
78, 29));

lblContraseñau.setFont(new java.awt.Font("DejaVu Serif
", 3, 15)); // NOI18N
lblContraseñau.setText("Contraseña:");
panelUsuario.add(lblContraseñau, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 220, -
1, 27));

mensajeA.setFont(new java.awt.Font("DejaVu Serif", 3, 1
4)); // NOI18N
mensajeA.setForeground(new java.awt.Color(255, 0, 51));
panelUsuario.add(mensajeA, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 280, 17
1, 30));

jSeparator1.setBackground(new java.awt.Color(0, 0, 0));
jSeparator1.setForeground(new java.awt.Color(0, 0, 0));
panelUsuario.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 200, 18
0, -1));
```

```
jSeparator2.setBackground(new java.awt.Color(0, 0, 0));
jSeparator2.setForeground(new java.awt.Color(0, 0, 0));
panelUsuario.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 280, 18
0, -1));

javax.swing.GroupLayout UsuarioLayout = new javax.swing.GroupLayout(Usuario.getContentPane())
Usuario.getContentPane().setLayout(UsuarioLayout);
UsuarioLayout.setHorizontalGroup(
    UsuarioLayout.createParallelGroup(javax.swing.Group
    Layout.Alignment.LEADING)
    .addComponent(panelUsuario, javax.swing.GroupLayout
    .DEFAULT_SIZE, 318, Short.MAX_VALUE)
);
UsuarioLayout.setVerticalGroup(
    UsuarioLayout.createParallelGroup(javax.swing.GroupLa
    yout.Alignment.LEADING)
    .addComponent(panelUsuario, javax.swing.GroupLayout.DE
    FAULT_SIZE, 391, Short.MAX_VALUE)
);

tablaUsuario.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
tablaUsuario.setModel(new javax.swing.table.DefaultTabl
eModel(
    new Object [][] {

    },
    new String [] {
        "idVendedor", "Vendedor", "Número"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false
    };

    public boolean isCellEditable(int rowIndex, int c
    olumnIndex) {
        return canEdit [columnIndex];
    }
});
tablaUsuario.addMouseListener(new java.awt.event.MouseA
dapter() {
    public void mouseClicked(java.awt.event.MouseEvent
    evt) {
        tablaUsuarioMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tablaUsuario);
```

```
jLabel1.setFont(new java.awt.Font("DejaVu Serif", 3, 1
8)); // NOI18N
jLabel1.setText("Lista de usuarios");

jLabel3.setFont(new java.awt.Font("DejaVu Serif", 3, 1
5)); // NOI18N
jLabel3.setText("Buscar:");

buscarU.setFont(new java.awt.Font("DejaVu Serif", 3, 14
)); // NOI18N
buscarU.addActionListener(new java.awt.event.ActionLi
stener() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        buscarUActionPerformed(evt);

    }
});
buscarU.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        buscarUKeyReleased(evt);
    }
});

jButton1.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jButton1.setText("Modificar usuario");
jButton1.addActionListener(new java.awt.event.ActionLis
tener() {
    public void actionPerformed(java.awt.event.ActionE
    vent
    evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton3.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jButton3.setText("Eliminar usario");
jButton3.addActionListener(new java.awt.event.ActionListen
er() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        jButton3ActionPerformed(evt);
    }
});

jButton2.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
```

```
// NOI18N
jButton2.setText("Regresar");
jButton2.addActionListener(new java.awt.event.ActionList
ener() {
    public void actionPerformed(java.awt.event.ActionEve
    nt evt) {
        jButton2ActionPerformed(evt);
    }
});

jLabel33.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/poema.
png"))); // NOI18N

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayo
    ut.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGroup(jPanel1Layout.createParallelGroup(javax.sw
        ing.GroupLayout.Alignment.TRAILING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(
                javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentia
                    lGroup()
                        .addGap(39, 39, 39)
                        .addComponent(jLabel3)
                        .addPreferredGap(javax.swing.Layout
                        Style.ComponentPlacement.UNRELATED)
                        .addComponent(buscarU, javax.swing.GroupLayout.PREFERRED
                        SIZE, 173, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.Lay
                        outStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_
                    .addGroup(javax.swing.GroupLayout.Al
                    gnment.TRAILING, jPanel1Layout.createSequentialGroup()
                        .addContainerGap(javax.swing.GroupL
                        ayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED
                        _SIZE, 193, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(134, 134, 134)))
                .addComponent(jLabel33, javax.swing.GroupLayout.PREFERRED_SIZE, 74, javax.swi
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequenti
                .addGap(44, 44, 44)
                .addGroup(jPanel1Layout.createParallelGroup(
                javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addGroup(jPanel1Layout.createSequentialGr
                    oup()
                        .addComponent(jButton2)
```

```
                        .addPreferredGap(javax.swing.LayoutStyl
                        e.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, S
                        .addComponent(jButton1)
                        .addGap(62, 62, 62)
                        .addComponent(jButton3))
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 607,
        .addContainerGap(42, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLa
    yout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGroup(jPanel1Layout.createParallelGroup(java
        x.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(jLabel1)
                .addGap(35, 35, 35)
                .addGroup(jPanel1Layout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(buscarU, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swin
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(39, 39, 39)
                .addComponent(jLabel33, javax.swing.GroupLayout.PREFERRED_SIZE, 74, javax.swi
        .addPreferredGap(javax.swing.LayoutStyle.ComponentP
        lacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 274, Shor
        t.MAX_VALUE)
        .addGap(15, 15, 15)
        .addGroup(jPanel1Layout.createParallelGroup(javax
        .swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton3)
            .addComponent(jButton1)
            .addComponent(jButton2))
        .addGap(49, 49, 49))
);

javax.swing.GroupLayout AdministrarusuarioLayout = new javax.swing.GroupLayout(Administrarusu
Administrarusuario.getContentPane().setLayout(Administrarus
uarioLayout);
AdministrarusuarioLayout.setHorizontalGroup(
    AdministrarusuarioLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFA
);
AdministrarusuarioLayout.setVerticalGroup(
    AdministrarusuarioLayout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
```

```
    .addGroup(AdministrarusuarioLayout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayou
        .addGap(0, 1, Short.MAX_VALUE))
);

jPanel2.setBackground(new java.awt.Color(241, 255, 255));

jLabel4.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel4.setText("Usuario:");

jLabel6.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); //
NOI18N
jLabel6.setText("Número:");

txfusuario.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txfusuario.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txfusuarioKeyTyped(evt);
    }
});

txfcorreo.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
txfcorreo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txfcorreoKeyTyped(evt);
    }
});

jLabel9.setFont(new java.awt.Font("DejaVu Serif", 3, 18));
// NOI18N
jLabel9.setText("Permisos");

ingresarInventario.setFont(new java.awt.Font("DejaVu Serif",
3, 12)); // NOI18N
ingresarInventario.setText("Ingresar al inventario");
ingresarInventario.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        ingresarInventarioActionPerformed(evt);
    }
});

ingresarHistorial.setFont(new java.awt.Font("DejaVu Serif", 3,
12)); // NOI18N
ingresarHistorial.setText("Ingresar al historial de ventas");
ingresarHistorial.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        ingresarHistorialActionPerformed(evt);
    }
});

agregarProducto.setFont(new java.awt.Font("DejaVu Serif", 3, 12));
// NOI18N
agregarProducto.setText("Agregar producto");
agregarProducto.addActionListener(new java.awt.event.ActionL
istener
() {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        agregarProductoActionPerformed(evt);
    }
});

editarProducto.setFont(new java.awt.Font("DejaVu Serif", 3, 12));
// NOI18N
editarProducto.setText("Editar producto");

eliminarProducto.setFont(new java.awt.Font("DejaVu Serif", 3,
12)); // NOI18N
eliminarProducto.setText("Eliminar producto");

jLabel10.setFont(new java.awt.Font("DejaVu Serif", 3, 18)); //
NOI18N
jLabel10.setText("Modificar");

btnguardaru.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
btnguardaru.setText("Guardar");
btnguardaru.addActionListener(new java.awt.event.ActionLis
tener(
) {
    public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        btnguardaruActionPerformed(evt);
    }
});

btncancelaru1.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
btncancelaru1.setText("Cancelar");
btncancelaru1.addActionListener(new java.awt.event.ActionListene
r() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
```

```
        btncancelaru1ActionPerformed(evt);
    }
});

jLabel34.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/contrato
.png"))); // NOI18N

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayo
    ut.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(227, 227, 227)
        .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 143, javax.swing.Grou
        .addGap(0, 0, Short.MAX_VALUE))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.s
        wing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(169, 169, 169)
                .addGroup(jPanel2Layout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel6)
                    .addComponent(jLabel4)))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(53, 53, 53)
                .addComponent(jLabel7)))
        .addGap(18, 18, 18)
        .addGroup(jPanel2Layout.createParallelGroup(javax.sw
        ing.GroupLayout.Alignment.LEADING)
            .addComponent(txfusuario, javax.swing.GroupLayout.PREFERRED_SIZE, 109, javax.swin
            .addComponent(txfcorreo, javax.swing.GroupLayout.PREFERRED_SIZE, 109, javax.swing
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPl
        acement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel34, javax.swing.GroupLayout.PREFERRED_SIZE, 71, javax.swing.Group
        .addGap(125, 125, 125))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.sw
        ing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(54, 54, 54)
                .addComponent(ingresarHistorial)
                .addGap(46, 46, 46)
                .addComponent(editarProducto))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(235, 235, 235)
                .addComponent(jLabel9))
            .addGroup(jPanel2Layout.createSequentialGroup()
```

```
                .addGap(18, 18, 18)
                .addComponent(ingresarInventario)
                .addGap(33, 33, 33)
                .addComponent(agregarProducto, javax.swing.GroupLayout.PREFERRED_SIZE, 193, j
                .addPreferredGap(javax.swing.LayoutStyle.Comp
                onentPlacement.RELATED)
                .addComponent(eliminarProducto)))
        .addContainerGap(26, Short.MAX_VALUE))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(65, 65, 65)
        .addComponent(btncancelaru1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentP
        lacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnguardaru)
        .addGap(93, 93, 93))
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swi
        ng.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(26, 26, 26)
                .addComponent(jLabel10)
                .addGroup(jPanel2Layout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialG
                    roup()
                        .addGap(33, 33, 33)
                        .addGroup(jPanel2Layout.createParal
                        lelGroup(javax.swing.GroupLayout.Ali
                        gnment.BASELINE)
                            .addComponent(jLabel4)
                            .addComponent(txfusuario, javax.swing.GroupLayout.PREFE
                            RRED_SIZE, javax.swing.GroupLayout.DEFAU
                            LT_SIZE, javax.swing.GroupLayo
                            ut.PREFERR
                            ED_SIZE)))
                    .addGroup(jPanel2Layout.createSequentia
                    lGroup()
                        .addGap(57, 57, 57)
                        .addComponent(jLabel34, javax.swing.GroupLayout.PREFERRED_SI
                        ZE, 88, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addPreferredGap(javax.swing.LayoutStyle.Comp
                onentPlacement.RELATED, 69, Short.MAX_VALUE)
                .addComponent(jLabel9)
                .addGap(35, 35, 35))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequent
```

```
                .addContainerGap(javax.swing.GroupLayout.DEFA
                ULT_SIZE, Short.MAX_VALUE)
                .addGroup(jPanel2Layout.createParallelGroup(ja
                vax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(jPanel2Layout.createParallelGroup
                    (javax.swing.GroupLayout.Alignment.B
                    ASELINE)
                        .addComponent(jLabel6)
                        .addComponent(txfcorreo, javax.swing.GroupLayout.PREFERRED_SIZE, java
                        ULT_SIZE, javax.swing.GroupLayout.PREFE
                        RRED_SIZE))
                    .addComponent(jLabel7))
                .addGap(140, 140, 140)))
        .addGroup(jPanel2Layout.createParallelGroup(java
        x.swing
        .GroupLayout.Alignment.BASELINE)
            .addComponent(ingresarInventario)
            .addComponent(agregarProducto)
            .addComponent(eliminarProducto))
        .addGap(31, 31, 31)
        .addGroup(jPanel2Layout.createParallelGroup(java
        x.swing
        .GroupLayout.Alignment.BASELINE)
            .addComponent(ingresarHistorial)
            .addComponent(editarProducto))
        .addGap(90, 90, 90)
        .addGroup(jPanel2Layout.createParallelGroup(j
        avax.swing.
        GroupLayout.Alignment.BASELINE)
            .addComponent(btnguardaru)
            .addComponent(btncancelaru1))
        .addGap(47, 47, 47))
);

javax.swing.GroupLayout ModificarusuarioLayout = new javax.swing.GroupLayout(Modificarusuario
Modificarusuario.getContentPane().setLayout(ModificarusuarioL
ayout);
ModificarusuarioLayout.setHorizontalGroup(
    ModificarusuarioLayout.createParallelGroup(javax.swing.Gr
    oupLayout.Alignment.LEADING)
    .addGroup(ModificarusuarioLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAUL
        T_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
ModificarusuarioLayout.setVerticalGroup(
    ModificarusuarioLayout.createParallelGroup(javax.swing.Gro
    upLayout.Alignment.LEADING)
    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFA
```

```
);

jLabel12.setFont(new java.awt.Font("DejaVu Serif", 3, 18));
// NOI18N
jLabel12.setText("Reporte diario");

tablaReporte.setFont(new java.awt.Font("DejaVu Serif", 3, 14
)); // NOI18N
tablaReporte.setModel(new javax.swing.table.Default
TableModel(
    new Object [][] {

    },
    new String [] {
        "Modelo", "Cantidad", "Fecha de venta"
    }
));
jScrollPane2.setViewportView(tablaReporte);

botonatrasRep.setFont(new java.awt.Font("DejaVu Serif",
3, 15)); // NOI18N
botonatrasRep.setText("Regresar");
botonatrasRep.addActionListener(new java.awt.event.Acti
onListener() {
    public void actionPerformed(java.awt.event.ActionEven
    t evt) {
        botonatrasRepActionPerformed(evt);
    }
});

jLabel8.setFont(new java.awt.Font("DejaVu Serif", 3, 15));
// NOI18N
jLabel8.setText("Buscar:");

buscarR.setFont(new java.awt.Font("DejaVu Serif", 3, 14));
// NOI18N
buscarR.addActionListener(new java.awt.event.ActionListen
er() {
    public void actionPerformed(java.awt.event.ActionEven
    t evt) {
        buscarRActionPerformed(evt);
    }
});
buscarR.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        buscarRKeyReleased(evt);
    }
});
```

```
jLabel35.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/
memorandum.png"))); // NOI18N

btnExcel.setFont(new java.awt.Font("DejaVu Serif", 3, 1
5)); // NOI18N
btnExcel.setText("Generar reporte");
btnExcel.addActionListener(new java.awt.event.Actio
nListener() {
    public void actionPerformed(java.awt.event.Ac
    tionEvent evt) {
        btnExcelActionPerformed(evt);
    }
});

javax.swing.GroupLayout ReporteLayout = new javax.swing.GroupLayout(Reporte.getContentPane())
Reporte.getContentPane().setLayout(ReporteLayout);
ReporteLayout.setHorizontalGroup(
    ReporteLayout.createParallelGroup(javax.swing.Grou
    pLayout.Alignment.LEADING)
    .addGroup(ReporteLayout.createSequentialGroup()
        .addContainerGap(77, Short.MAX_VALUE)
        .addGroup(ReporteLayout.createParallelGroup(java
        x.swing.GroupLayout.Alignment.LEADING)
            .addGroup(ReporteLayout.createSequentialGro
            up()
                .addComponent(jLabel8)
                .addGap(4, 4, 4)
                .addComponent(buscarR, javax.swing.GroupLayout.PREFERRED_SIZE, 173, javax.swi
                .addGap(216, 216, 216))

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, ReporteLayout.createSequent
                .addComponent(jLabel12)
                .addGap(126, 126, 126)))
        .addComponent(jLabel35, javax.swing.GroupLayout.PREFERRED_SIZE, 95, javax.swing.Group
        .addGap(43, 43, 43))
    .addGroup(ReporteLayout.createSequentialGroup()
        .addGroup(ReporteLayout.createParallelGroup(ja
        vax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(ReporteLayout.createSequentialGroup()
                .addGap(149, 149, 149)
                .addComponent(botonatrasRep, javax.swing.GroupLayout.PREFERRED_SIZE, 127, jav
                .addGap(40, 40, 40)
                .addComponent(btnExcel, javax.swing.GroupLayout.PREFERRED_SIZE, 193, javax.sw
            .addGroup(ReporteLayout.createSequentialGroup()
                .addGap(45, 45, 45)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 593, java
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
ReporteLayout.setVerticalGroup(
```

```
    ReporteLayout.createParallelGroup(javax.swing.Gro
    upLayout.Alignment.LEADING)
    .addGroup(ReporteLayout.createSequentialGroup()
        .addGroup(ReporteLayout.createParallelGroup(ja
        vax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(ReporteLayout.createSequentialGr
            oup()
                .addContainerGap()
                .addComponent(jLabel12)
                .addGap(29, 29, 29)
                .addGroup(ReporteLayout.createParallelG
                roup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel8)
                    .addComponent(buscarR, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swin
            .addGroup(ReporteLayout.createSequentialGroup()
                .addGap(24, 24, 24)
                .addComponent(jLabel35)))
        .addPreferredGap(javax.swing.LayoutStyle.Compon
        entPlacement.UNRELATED)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 334, javax.swing.
        .addGap(12, 12, 12)
        .addGroup(ReporteLayout.createParallelGroup(jav
        ax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnExcel)
            .addComponent(botonatrasRep))
        .addContainerGap(30, Short.MAX_VALUE))
);

jMenuItem1.setText("jMenuItem1");

jMenuItem2.setText("jMenuItem2");

tablaInventario3.setFont(new java.awt.Font("DejaVu Serif
", 3, 14)); // NOI18N
tablaInventario3.setModel(new javax.swing.table.DefaultT
ableModel(
    new Object [][] {

    },
    new String [] {
        "Modelo", "Marca", "Ram", "M.Interna", "Bateria
        ", "Pixeles", "Precio de compra", "Precio de v
        enta", "Cantid
        ad"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, fals
        e, false, false
```

```
    };

    public boolean isCellEditable(int rowIndex, int co
    lumnIndex) {
        return canEdit [columnIndex];
    }
});
tablaInventario3.addMouseListener(new java.awt.event.M
ouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent e
    vt) {
        tablaInventario3MouseClicked(evt);
    }
});
scrollpaneInventario3.setViewportView(tablaInventario3);
if (tablaInventario3.getColumnModel().getColum
nCount() > 0) {
    tablaInventario3.getColumnModel().getColumn
    (0).setResizable(false);
    tablaInventario3.getColumnModel().getColum
    n(1).setResizable(false);
    tablaInventario3.getColumnModel().getColu
    mn(2).setResizable(false);
    tablaInventario3.getColumnModel().getColu
    mn(3).setResizable(false);
    tablaInventario3.getColumnModel().getCol
    umn(4).setResizable(false);
    tablaInventario3.getColumnModel().getCol
    umn(5).setResizable(false);
    tablaInventario3.getColumnModel().getCo
    lumn(6).setResizable(false);
    tablaInventario3.getColumnModel().getCol
    umn(7).setResizable(false);
    tablaInventario3.getColumnModel().get
    Column(8).setResizable(false);
}

jLabel24.setIcon(new javax.swing.ImageIcon
(getClass().getResour
e("/Imagen/iphone-5.png"))); // NOI18N

jLabel27.setFont(new java.awt.Font("DejaVu
Serif", 3, 18)); // NOI18N
jLabel27.setText("SFH");

javax.swing.GroupLayout panelImagen1Layout
= new javax.swing.GroupLayout(panelImagen1);
panelImagen1.setLayout(panelImagen1Layout);
panelImagen1Layout.setHorizontalGroup(
```

```
    panelImagen1Layout.createParallelGroup(javax.s
    wing.GroupLayout.Alignment.LEADING)
    .addGroup(panelImagen1Layout.createSequentia
    lGroup()
        .addContainerGap()
        .addComponent(jLabel24)
        .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
        .addComponent(jLabel27, javax.swing.GroupLayo
        ut.DEFAULT_SIZE, 54, Short.MAX_VALUE)
        .addGap(47, 47, 47))
);
panelImagen1Layout.setVerticalGroup(
    panelImagen1Layout.createParallelGroup(javax.swin
    g.GroupLayout.Alignment.LEADING)
    .addGroup(panelImagen1Layout.createSequentialGroup()
        .addGroup(panelImagen1Layout.createParallelGr
        oup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel24, javax.swing.GroupLayout.DEFAULT_SIZE, 44, Short.MAX_VALUE
            .addComponent(jLabel27, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLa
            hort.MAX_VALUE))
        .addContainerGap())
);

panelCambios1.setBackground(new java.awt.Color(120, 130, 134));

btnAgregar1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
btnAgregar1.setText("Agregar ");
btnAgregar1.addActionListener(new java.awt.event
.ActionListener() {

    public void actionPerformed(java.awt.event.
    ActionEvent evt) {
        btnAgregar1ActionPerformed(evt);
    }
});

btnModificar1.setFont(new java.awt.Font(
"DejaVu Serif", 3, 15)); // NOI18N
btnModificar1.setText("Modificar ");
btnModificar1.addActionListener(new java.awt.even
t.ActionListener() {
    public void actionPerformed(java.awt.event.A
    ctionEvent evt) {
        btnModificar1ActionPerformed(evt);
    }
});

btnEliminar1.setFont(new java.awt.Font("DejaVu Serif", 3, 15)); // NOI18N
```

```
btnEliminar1.setText("Eliminar ");
btnEliminar1.addActionListener(new java.awt.event.
ActionListener() {
    public void actionPerformed(java.awt.event.A
    ctionEvent evt) {
        btnEliminar1ActionPerformed(evt);
    }
});

imagenEliminar1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/
contenedor-de-basura.png"))); // NOI18N

imagenAgregar1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/s
igno-de-mas.png"))); // NOI18N

imagenModificar1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/l
ibretas.png"))); // NOI18N

jLabel29.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/
regreso.png"))); // NOI18N

btnAtrasinventario1.setFont(new java.awt.Font("DejaVu
Serif", 3, 15)); // NOI18N
btnAtrasinventario1.setText("Regresar");
btnAtrasinventario1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.Action
    Event evt) {
        btnAtrasinventario1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout panelCambios1Layout = new javax.swing.GroupLayout(panelCambios1);
panelCambios1.setLayout(panelCambios1Layout);
panelCambios1Layout.setHorizontalGroup(
    panelCambios1Layout.createParallelGroup(javax.swing
    .GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelCambios1Layout.createSequentia
        .addGroup(panelCambios1Layout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelCambios1Layout.createSequential
            Group()
                .addGap(104, 104, 104)
                .addComponent(jLabel29, javax.swing.GroupLayout.DEFAULT_SIZE, 314, Short.MAX_
                .addPreferredGap(javax.swing.LayoutStyle.C
                omponentPlacement.RELATED))
            .addGroup(panelCambios1Layout.createSequent
            ialGroup()
                .addGap(96, 96, 96)
                .addComponent(btnAtrasinventario1, javax.swing.GroupLayout.PREFERRED_SIZE, 11
```

```
                ZE)
                .addPreferredGap(javax.swing.LayoutSt
                yle.ComponentPlacement.RELATED, javax.sw
                ing.GroupLay
                out.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addGroup(panelCambios1Layout.createParalle
        lGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnAgregar1, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swi
            .addGroup(panelCambios1Layout.createSequ
            entialGroup()
                .addGap(19, 19, 19)
                .addComponent(imagenAgregar1)))
        .addGap(174, 174, 174)
        .addGroup(panelCambios1Layout.createParallelGro
        up(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnModificar1, javax.swing.GroupLayout.PREFERRED_SIZE, 126, javax.s
            .addGroup(javax.swing.GroupLayout.Alignme
            nt.TRAILING, panelCambios1Layout.createSequen
            tialGroup
            ()
                .addComponent(imagenModificar1)
                .addGap(31, 31, 31)))
        .addGap(148, 148, 148)
        .addGroup(panelCambios1Layout.createParallelGro
        up(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnEliminar1, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.sw
            .addGroup(javax.swing.GroupLayout.Align
            ment.TRAILING, panelCambios1Layout.createSequentialGroup()
                .addComponent(imagenEliminar1)
                .addGap(23, 23, 23)))
        .addGap(95, 95, 95))
);
panelCambios1Layout.setVerticalGroup(
    panelCambios1Layout.createParallelGroup(javax.swing.G
    roupLayout.Alignment.LEADING)
    .addGroup(panelCambios1Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SI
        ZE, Short.MAX_VALUE)
        .addGroup(panelCambios1Layout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.TRAILING, false)
            .addComponent(imagenEliminar1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.
            MAX_VALUE)
            .addComponent(imagenModificar1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing
            MAX_VALUE)
            .addComponent(imagenAgregar1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.G
            MAX_VALUE)
            .addComponent(jLabel29, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.Gr
            MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.Component
```

```
        Placement.RELATED)
        .addGroup(panelCambios1Layout.createParallelGroup(j
        avax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnAgregar1)
            .addComponent(btnModificar1)
            .addComponent(btnEliminar1)
            .addComponent(btnAtrasinventario1))
        .addGap(21, 21, 21))
);

txtBuscar1.setFont(new java.awt.Font("DejaVu Serif", 3, 14)); // NOI18N
txtBuscar1.setBorder(null);
txtBuscar1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtBuscar1ActionPerformed(evt);
    }
});
txtBuscar1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtBuscar1KeyReleased(evt);
    }
});

lblBuscador1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagen/busca
dor.png"))); // NOI18N

javax.swing.GroupLayout panelInventario1Layout = new javax.swing.GroupLayout(panelInventario1
panelInventario1.setLayout(panelInventario1Layout);
panelInventario1Layout.setHorizontalGroup(
    panelInventario1Layout.createParallelGroup(javax.swi
    ng.GroupLayout.Alignment.LEADING)
    .addGroup(panelInventario1Layout.createSequentialGroup()
        .addGroup(panelInventario1Layout.createParallelG
        roup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelInventario1Layout.createSeque
            ntialGroup()
                .addComponent(panelImagen1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swi
                .addPreferredGap(javax.swing.LayoutStyle
                .ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_
                .addComponent(lblBuscador1)
                .addGap(18, 18, 18)
                .addComponent(txtBuscar1, javax.swing.GroupLayout.PREFERRED_SIZE, 140, javax.
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelInventario1Layout.crea
                .addContainerGap(51, Short.MAX_VALUE)
                .addComponent(panelCambios1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.sw
            .addGroup(panelInventario1Layout.creat
            eSequentialGroup()
                .addGap(48, 48, 48)
                .addComponent(scrollpaneInventario3)))
```

```
        .addContainerGap())
);
panelInventario1Layout.setVerticalGroup(
    panelInventario1Layout.createParallelGroup(javax.swing.G
    rou
    pLayout.Alignment.LEADING)
    .addGroup(panelInventario1Layout.createSequentialGro
    up()
        .addGroup(panelInventario1Layout.createParallelGr
        oup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelInventario1Layout.createSeque
            ntialGroup()
                .addContainerGap()
                .addGroup(panelInventario1Layout.createP
                arallelGroup(javax.swing.GroupLayout.Ali
                gnment.TRAILIN
                G)
                    .addComponent(panelImagen1, javax.swing.GroupLayout.PREFERRED_SIZE, javax
                    .addComponent(txtBuscar1, javax.swing.GroupLayout.PREFERRED_S
                    IZE, 34, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 13, Shor

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, panelInventario1Layout.crea
                .addContainerGap(javax.swing.GroupLayout.
                DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblBuscador1, javax.swing.GroupLayout.PREFERRED_SIZ
                E, 34, javax.swing.GroupLayout.PREFER
                RED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
        .addComponent(scrollpaneInventario3, javax.swing.GroupLayout.PREFERRED_SIZE, 227, jav
        .addPreferredGap(javax.swing.LayoutStyle.Compon
        entPlacement.UNRELATED)
        .addComponent(panelCambios1, javax.swing.GroupLayout.PREFERRED_SIZE, 121, javax.swing
        .addGap(78, 78, 78))
);

javax.swing.GroupLayout Inventario1Layout = new javax.swing.GroupLayout(Inventario1.getConten
Inventario1.getContentPane().setLayout(Inventario1Layout);
Inventario1Layout.setHorizontalGroup(
    Inventario1Layout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
    .addGroup(Inventario1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(panelInventario1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.G
        .addContainerGap(javax.swing.GroupLayout.DEF
        AULT_SIZE, Short.MAX_VALUE))
);
Inventario1Layout.setVerticalGroup(
    Inventario1Layout.createParallelGroup(javax.swin
```

```
    g.GroupLayout.Alignment.LEADING)
    .addGroup(Inventario1Layout.createSequentialGroup()
        .addComponent(panelInventario1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.G
        .addGap(0, 0, Short.MAX_VALUE))
);

setDefaultCloseOperation(javax.swing.WindowConstants.
EXIT_ON_CLOSE);
setBackground(new java.awt.Color(0, 255, 204));
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.Wind
    owEvent evt) {
        formWindowClosing(evt);
    }
});

panelInicio.setBackground(new java.awt.Color(255, 255, 255));

imagenLogo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imag
en/LOGO.png"))); // NOI18N

javax.swing.GroupLayout panelInicioLayout = new javax.swing.GroupLayout(panelInicio);
panelInicio.setLayout(panelInicioLayout);
panelInicioLayout.setHorizontalGroup(
    panelInicioLayout.createParallelGroup(javax.swing.Gr
    oupLayout.Alignment.LEADING)
    .addGroup(panelInicioLayout.createSequentialGr
    oup()
        .addGap(472, 472, 472)
        .addComponent(imagenLogo, javax.swing.GroupLayout.PREFERRED_SIZE, 413, javax.swing.Gr
        .addContainerGap(554, Short.MAX_VALUE))
);
panelInicioLayout.setVerticalGroup(
    panelInicioLayout.createParallelGroup(javax.swing.Gr
    oupLayout.Alignment.LEADING)
    .addComponent(imagenLogo, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLa
    AX_VALUE)
);

menuInicio.setText("Inicio");

iniciarSesion.setFont(new java.awt.Font("DejaVu Serif",
3, 14)); // NOI18N
iniciarSesion.setText("Iniciar sesión");
iniciarSesion.addActionListener(new java.awt.event.Actio
nListener() {
    public void actionPerformed(java.awt.event.ActionE
    vent evt) {
        iniciarSesionActionPerformed(evt);
```

```
        }
    });
    menuInicio.add(iniciarSesion);


    registrarse.setFont(new java.awt.Font("DejaVu Serif", 3, 14)); // NOI18N
    registrarse.setText("Registrarse");
    registrarse.addActionListener(new java.awt.event.Ac
    tionListener() {
        public void actionPerformed(java.awt.event.ActionEven
        t evt) {
            registrarseActionPerformed(evt);
        }
    });
    menuInicio.add(registrarse);

    Menu.add(menuInicio);

    setJMenuBar(Menu);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Ali
        gnment.LEADING)
        .addComponent(panelInicio, javax.swing.GroupLayout.DEF
        AULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Al
        ignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(panelInicio, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLay
            LUE)
            .addContainerGap())
    );

    pack();
}// </editor-fold>

private void botonCobrarActionPerformed(java.awt.event.Action
Event evt) {
    Sonido();
    //Verificamos que el administrador aya seleccionado un
    producto a vender
    if(Tablapre_venta.getRowCount()!=0){
    TablaAdministrador.dispose();
    fd.setText(m.productoFecha());
    rou = Tablapre_venta.getRowCount();// FIla
```

```
        columna=Tablapre_venta.getColumnCount();// columna
        int r=rou;
         cd=new int[r];
         mode=new String[r];
        for(int i=0; i<r;i++){
         mode[i]= String.valueOf(Tablapre_venta.getValueAt(i,0)
         );// almacemos el modelo
         cd[i]= Integer.parseInt(String.valueOf(Tablapre_vent
         a.getValueAt(i,columna-1)));  //almacenamos la cantidad
         de producto
         }
        txtTotalp.setText(textCobrar.getText());
        NoP.setText(""+rou);
        Cobrar.setBounds(590, 322, 590, 322);
        Cobrar.setLocationRelativeTo(null);
        Cobrar.setModal(true);
        Cobrar.setVisible(true);
         }else{// de lo contrario le notificamos lo siguiente
          JOptionPane.showMessageDialog(null,"No ha selecciona
          do el producto a vender.");
       }
    }

    private void btnAgregarActionPerformed(java.awt.event.Act
    ionEvent evt) {
        // Boton de agregar producto despliega la ventana de
        agregar prducto
        Sonido();
        Inventario.dispose();
        AgregarP.setBounds(552, 529, 414, 470);
        AgregarP.setLocationRelativeTo(null);
        AgregarP.setModal(true);
        AgregarP.setVisible(true);
    }

    private void btnModificarActionPerformed(java.awt.event.A
    ctionEvent evt) {
//Boton de moficcar producto despliega la ventana de modif
icar producto
//Validando que se alla sellecionado una fila o registro
        Sonido();
            if (tablaInventario.getSelectedRow()!=-1) {
            Inventario.dispose();
            ModificarP.setBounds(500, 518, 406, 450);
            ModificarP.setLocationRelativeTo(null);
            ModificarP.setModal(true);
            ModificarP.setVisible(true);

        } else {
```

```
        JOptionPane.showMessageDialog(null,"Error no h
        ay fila seleccionada.");
    }


}

private void btnCobrarActionPerformed(java.awt.event.Ac
tionEvent evt) {
    Sonido();
    // Cobrar del empleado


    //validamos que el vendedor aya selelcionado un
    producto a vender
    if(Tablapre_venta1.getRowCount()!=0){


    TablaEmpleado.dispose();
    rou = Tablapre_venta1.getRowCount();// FIla
    columna=Tablapre_venta1.getColumnCount();// col
    umna

    int r=rou;
     cd=new int[r];
     mode=new String[r];

    for(int i=0; i<r;i++){
        mode[i]= String.valueOf(Tablapre_venta1.getV
        alueAt(i,0));
        cd[i]= Integer.parseInt(String.valueOf(Tabla
        pre_venta1.getValueAt(i,columna-1)));
     }

   txtTotalp.setText(txtTotal.getText());
    NoP.setText(""+Tablapre_venta1.getRowCount());
    TablaEmpleado.dispose();
    Cobrar.setBounds(590, 322, 590, 322);
    Cobrar.setLocationRelativeTo(null);
    Cobrar.setModal(true);
    Cobrar.setVisible(true);
     }else{// de lo contrario le reliazamos la siguiente
     notificación
        JOptionPane.showMessageDialog(null,"Usted aún no h
        a seleccionado el producto a vender.");
    }
}
```

```
    private void btnCancelarpActionPerformed(java.awt.event
    .ActionEvent evt) {
       Sonido();
// cancelamos la compra del producto en Jdialog cobrar
// Se valida quien es el vendedor, de esa forma se regr
esa a su respectiva tabla de ventas
       Cobrar.setVisible(false);

       if(usuario=="usuario"){
       limpiaTabla(modelo3);
       limpiaTabla(modelo8);
       textCobrar.setText("");
       txtRecibido.setText("");
       txtTotal.setText("");
       cp1.setValue(0);
       TablaEmpleado.setBounds(850, 535, 1400, 481);
       TablaEmpleado.setLocationRelativeTo(null);
       TablaEmpleado.setModal(true);
       TablaEmpleado.setVisible(true);

       }else{
       limpiaTabla(modelo2);
       limpiaTabla(modelo7);
       textCobrar.setText("");
       txtRecibido.setText("");
       pc.setValue(0);
       TablaAdministrador.setBounds(850, 535, 1400, 481);
       TablaAdministrador.setLocationRelativeTo(null);
       TablaAdministrador.setModal(true);
       TablaAdministrador.setVisible(true);
       }
    }

    private void btnGuardarpActionPerformed(java.awt.event.Acti
    onEvent evt) {
        Sonido();
        // Guardar producto

        if(txtModelo.getText().isEmpty()||txtMarca.getText()
        .isEmpty()||txtRam.getText().isEmpty()||txtMemoria.getT
        ext().isEmpty() || txtBateria.getText().isEmpty() || txtPixeles.getText
        ().isEmpty() || txtPreciov.getText().isEmpty() || txtPrecioc.getText().isEmpty() || txtCantid
        t().isEmpty()){
        JOptionPane.showMessageDialog(null,"Verifique que los
        campos no estén vacíos");


        }else{
              m.insertarModelo(bd, tablet,txtModelo.getText(),txtMarca.getText(),t
```

```
          xtRam.getText(),txtMemoria.getText(),txtBateria.getText(), txtPixeles.getText(),txtPrec
          cioc.getText(), txtCantidadm.getText());
    limpiarisertar();
    AgregarP.dispose();
    if("usuario".equals(usuario)){
    Inventario1.setBounds(1237, 560, 1237, 560);
    Inventario1.setLocationRelativeTo(null);
    Inventario1.setModal(true);
    Inventario1.setVisible(true);
    }else{limpiaTabla(modelo1);
    consultarTelefono(modelo1,tablaInventario);
    Inventario.setBounds(1237, 560, 1237, 560);
    Inventario.setLocationRelativeTo(null);
    Inventario.setModal(true);
    Inventario.setVisible(true);
    limpiaTabla(modelo1);
    consultarTelefono(modelo1,tablaInventario);
    }


    }





}

private void btnCamcelarproActionPerformed(java.awt.event
.ActionEvent evt) {
   Sonido();
    // AgregarP boton de cancelar
    AgregarP.dispose();
    if("usuario".equals(usuario)){
    Inventario1.setBounds(1237, 560, 1237, 560);
    Inventario1.setLocationRelativeTo(null);
    Inventario1.setModal(true);
    Inventario1.setVisible(true);
    }else{
    Inventario.setBounds(1237, 560, 1237, 560);
    Inventario.setLocationRelativeTo(null);
    Inventario.setModal(true);
    Inventario.setVisible(true);
    }
   JOptionPane.showMessageDialog(null, usuario);
}

private void btnAtrasinventarioActionPerformed(java.awt.ev
ent.ActionEvent evt) {
```

```
//Método que permite regresar a la tabla inventario
        Sonido();
        Inventario.dispose();
        TablaAdministrador.setBounds(850, 535, 1400, 481);
        TablaAdministrador.setLocationRelativeTo(null);
        TablaAdministrador.setModal(true);
        TablaAdministrador.setVisible(true);

    }

    private void btnGuardarActionPerformed(java.awt.event.ActionE
    vent evt) {
        Sonido();
        //Este método nos permite realizar unn registro de un usua
        rio nuevo, primero validando al gunos campo
        // como que la cantidad de digitos del campo telefono no
        sea mayo que cero, así como que el código del administrador
        // sea correcto, finalmete se validad que no se queden camp
        os vacíos.
        int  a= txtCorreo.getText().length();
        String conAdmin=String.valueOf(contraAdmin.getPassword());

       if(txtNombre.getText().isEmpty()||txtCorreo.getText().isE
       mpty()||txtContrasena.getText().isEmpty()||txtConfirmar.getTex
       t().i
       sEmpty()||contraAdmin.getText().isEmpty()||a>10||a<10){
           if(a>10){
             JOptionPane.showMessageDialog(null,"En el campo t
             elefono solo se admiten 10 digitos.");

           }else if(a<10){

           JOptionPane.showMessageDialog(null,"En el campo tel
           efono solo se admiten 10 digitos.");
           }else{
             JOptionPane.showMessageDialog(null,"No se admit
             en campos en blanco.");
           }
       }else{

        String adm="1234";
        if(adm.equals(contraAdmin.getText())){
        String con1=String.valueOf(txtContrasena.getPassword());
        String con2=String.valueOf(txtConfirmar.getPassword());
        m.insertarUsuario(bd, tableu, txtNombre.getText(),txtCorreo.getText(),con1);
        Registrar.dispose();
        txtNombre.setText("");
        txtCorreo.setText("");
        txtContrasena.setText("");
```

```
    txtConfirmar.setText("");
    contraAdmin.setText("");
    Usuario.setBounds(384, 388,320, 390);
    Usuario.setLocationRelativeTo(null);
    Usuario.setModal(true);
    Usuario.setVisible(true);
    }else {
    JOptionPane.showMessageDialog(null,"El código del
    administrador es incorrecto.");
    }

    }
   if(!ingresarInventario.isSelected()){
        inventario1.setEnabled(false);
    }else

    if(!agregarProducto.isSelected()){
        btnAgregar1.setEnabled(false);
    }
    if(!eliminarProducto.isSelected()){
        btnEliminar1.setEnabled(false);
    }

    if(!editarProducto.isSelected()){
        btnModificar1.setEnabled(false);
    }
    if(!ingresarHistorial.isSelected()){
        historialv.setEnabled(false);
    }


}

private void btnAtrasActionPerformed(java.awt.event.ActionEven
t evt) {
    // Permite regresar a la pantalla principal, limpiando los
    campo del Jdialog registrarse
    Sonido();
    txtNombre.setText("");
    txtCorreo.setText("");
    txtContrasena.setText("");
    txtConfirmar.setText("");
    contraAdmin.setText("");
    Registrar.dispose();
}

private void txtRecibidoActionPerformed(java.awt.event.ActionEv
ent evt) {
```

```
}

private void aceptarAdministradorActionPerformed(java.awt.event.ActionEvent evt) {
    Sonido();
    // Este método nos permite validad el tipo de usuario que inger
    esa al sitema. Si es usuario o administrador
    // realizando una camparación una vez identificado al tipo de
    usuario se despliegan sus respectivos Jdialog
    // en caso de que algun dato sea erroneo se le notifica al us
    uario.
    String con1=String.valueOf(txtContrasenau.getPassword());
     try{
        if(m.nombre(txtNombreu.getText(),bd)==1 && m.contraseña(con1,bd)==1){
            //Nombre del administrador
            if("Hermilo".equals(m.usuario(txtNombreu.getText(),b
            d))){
             // m.mensajeActualizado(mensajeA,"Datos validos");
              JOptionPane.showMessageDialog(null,"Acceso exitos
              o al Administrador");
              Usuario.dispose();
              usuario=null;
              TablaAdministrador.setBounds(850, 535, 1400, 481);
              TablaAdministrador.setLocationRelativeTo(null);
              TablaAdministrador.setModal(true);
              TablaAdministrador.setVisible(true);
              txtNombreu.setText("");
              txtContrasenau.setText("");
            }else {
              //m.mensajeActualizado(mensajeA,"Datos validos");
              JOptionPane.showMessageDialog(null,"Acceso exitoso al usuario");
              Usuario.dispose();
              usuario="usuario";
              TablaEmpleado.setBounds(850, 535, 1400, 481);
              TablaEmpleado.setLocationRelativeTo(null);
              TablaEmpleado.setModal(true);
              TablaEmpleado.setVisible(true);
              txtNombreu.setText("");
              txtContrasenau.setText("");


            }
        }else if(m.nombre(txtNombreu.getText(),bd)==0 && m.contraseña(con1,bd)==0) {
            m.mensajeActualizado(mensajeA,"Datos Incorrectos");
            //JOptionPane.showMessageDialog(null,"Usuario y
            contraseña incorrectos :-(");
        }else if (m.contraseña(con1,bd)==0){
            m.mensajeActualizado(mensajeA,"Datos Incorrectos");
           // JOptionPane.showMessageDialog(null, "Contraseña
           incorrecta :-(");
```

```
        }else if(m.nombre(txtNombreu.getText(),bd)==0){
            m.mensajeActualizado(mensajeA,"Datos Incorrectos");
            //JOptionPane.showMessageDialog(null,"Usuario inc
            orrecto");
        }

    }catch(Exception e){
        JOptionPane.showMessageDialog(null,"Error: "+e);
    }

}

private void txtNombreuActionPerformed(java.awt.event.Action
Event evt) {

}

private void txtContrasenauActionPerformed(java.awt.event.A
ctionEvent evt) {

}

private void iniciarSesionActionPerformed(java.awt.event.A
ctionEvent evt) {
    // El método (Boton de sesión) despliega la pantalla usu
    ario(ventana de Logeo)
    Usuario.setBounds(384, 388, 320, 390);
    Usuario.setLocationRelativeTo(null);
    Usuario.setModal(true);
    Usuario.setVisible(true);
}

private void registrarseActionPerformed(java.awt.event.Acti
onEvent evt) {
    // Método (Boton de registerarse) despliega la venta de
    registrar usuario
    Registrar.setBounds(400, 500, 337,435);
    Registrar.setLocationRelativeTo(null);
    Registrar.setModal(true);
    Registrar.setVisible(true);
}

private void atrasBtnActionPerformed(java.awt.event.ActionEv
ent evt) {
    // cierra la ventana TablaEmpleado
    Sonido();
    TablaEmpleado.dispose();
}
```

```
private void txtBuscarActionPerformed(java.awt.event.ActionEv
ent evt) {

}

private void ingresarInventarioActionPerformed(java.awt.event
.ActionEvent evt) {

}

private void ingresarHistorialActionPerformed(java.awt.event.
ActionEvent evt) {

}

private void agregarProductoActionPerformed(java.awt.event.A
ctionEvent evt) {

}

private void jButton1ActionPerformed(java.awt.event.ActionEv
ent evt) {
    //Este método nos despliega la ventana ModificarUsuario,
    selecionando primeramente al usuario
    // de lo contrario se le notifica que no ha seleccionado
    al usuario
    if (tablaUsuario.getSelectedRow()!=-1) {
        Administrarusuario.dispose();
        Modificarusuario.setBounds(600, 600, 650, 600);
        Modificarusuario.setLocationRelativeTo(null);
        Modificarusuario.setModal(true);
        Modificarusuario.setVisible(true);
    } else {
       JOptionPane.showMessageDialog(null,"Error no hay fila seleccionada.");
    }
}

private void txtPreciocActionPerformed(java.awt.event.
ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtCantidadmActionPerformed(java.awt.event
.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPreciovActionPerformed(java.awt.event
.ActionEvent evt) {
```

```
    // TODO add your handling code here:
}

private void txtcPreciovmActionPerformed(java.awt.event.
ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtPreciocmActionPerformed(java.awt.event.
ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtcantmActionPerformed(java.awt.event.
ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnGuardarp1ActionPerformed(java.awt.event
.ActionEvent evt) {
    // En esta sección se modifican los productos ya
    existentes, pero amtes de ello se valida que no
    existan campos
    //vacíos

    if(txtMarcam.getText().isEmpty()|| txtRamm.getTe
    xt().isEmpty()|| txtMemoriam.getText().isEmpty()||
    txtBateriam.ge
    tText().isEmpty()|| txtPixelesm.getText().isEmpty(
    )|| txtcPreciovm.ge
    tText().isEmpty()|| txtPreciocm.getText().isEmpty()||txtcantm.getText(
    ).isEmpty()){
        JOptionPane.showMessageDialog(null, "No se ad
        mite campos vacios");
    }else{
    m.modificarModelo(bd, tablet, txtMarcam.getText(), txtRa
    mm.getText(), txtMemoriam.getText(), txtBateriam.g
    etText(), txtP
    ixelesm.getText(), txtcPreciovm.getText(), txtPreciocm.getText(),txtcantm.getText(), tablaInv
    entario);
    limpiar();
    ModificarP.dispose();//
    limpiaTabla(modelo1);
    consultarTelefono(modelo1,tablaInventario);
    Inventario.setBounds(1237, 560, 1237, 560);
    Inventario.setLocationRelativeTo(null);
    Inventario.setModal(true);
    Inventario.setVisible(true);
    }
```

```
}

private void btnCamcelarpro1ActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // accedemos al boton de cancelar para cerrar la ve
    ntana de modificar producto sin modificaciones
    ModificarP.dispose();
    if("usuario".equals(usuario)){
    Inventario1.setBounds(850, 535, 1200, 600);
    Inventario1.setLocationRelativeTo(null);
    Inventario1.setModal(true);
    Inventario1.setVisible(true);
   }else{
    Inventario.setBounds(850, 535, 1200, 600);
    Inventario.setLocationRelativeTo(null);
    Inventario.setModal(true);
    Inventario.setVisible(true);
    }
}

private void menuAtrasActionPerformed(java.awt.event.Actio
nEvent evt) {
    //En este metodo se abre la ventana usuario
    TablaEmpleado.dispose();
    Usuario.setBounds(384, 388, 320, 390);
    Usuario.setModal(true);
    Usuario.setVisible(true);
}

private void menuPermisosActionPerformed(java.awt.event.A
ctionEvent evt) {
    // TODO add your handling code here:
}



public void usuarioEliminar(String bd, String ntabla ){
    // En este método se eliminan los usuarios desde la base d
    e datos así como en la tabla tablaUsuario
     try {
         Statement s= m.conectaBase(bd).createStatement();
         String eliminar="DELETE FROM "+ntabla+ " WHERE codigob="+tablaUsuario.getValueAt(tablaUs
         ctedRow(), 0)+";";
         System.out.print(eliminar);
         s.executeUpdate(eliminar);
         JOptionPane.showMessageDialog(null,"Fila eliminada correctamente");
         m.conectaBase(bd).close();
         JOptionPane.showMessageDialog(null, "No se eliminó l
         a fila");
```

```
        } catch (SQLException ex) {
            Logger.getLogger(Ventas.class.getName()).log(Level.
            SEVERE, null, ex);
        }
}

    public void productoEliminar(String bd, String ntabla ){
 // Con este metodo se accede a la base de datos para poder elim
 inar un producto no deseado
        try {
            Statement s= m.conectaBase(bd).createStatement();
            String eliminar="DELETE FROM "+ntabla+ " WHERE codigob="+tablaInventario1.getValueAt(tab
            1.getSelectedRow(), 0)+";";
            System.out.print(eliminar);
            s.executeUpdate(eliminar);
            JOptionPane.showMessageDialog(null,"Fila eliminada correctamente.");
            m.conectaBase(bd).close();
            JOptionPane.showMessageDialog(null, "No se eliminó la
            fila");
        } catch (SQLException ex) {

            Logger.getLogger(Ventas.class.getName()).log(Level.SE
            VERE, null, ex);
        }
}



    private void botonEliminarpActionPerformed(java.awt.event.Act
    ionEvent evt) {
        Sonido();
        // Eliminamos un produto de la base de datos
        int selecion = Tablapre_venta.getSelectedRow();
         int mul;
         int cantidad;
        //Seleccionamos la fila con getSelectedRow() para despues
        eliminarla.
         if(Tablapre_venta.getSelectedRow()!=-1){
            mul= Integer.parseInt(((String.valueOf(Tablapre_venta
            .getValueAt(selecion,2)))));
            if(mul!=1){
               cantidad= Integer.parseInt(JOptionPane.showInputDialog("Cuántos producto desea elimina
               if(cantidad >=mul){
                  if(cantidad>mul){
                    confirmar=JOptionPane.showConfirmDialog(null
                    , "La cantidad a eliminar es mayor al existente, de
                    sea eliminar todos los productos ",  "TELEFONOS",
                    JOptionPane.OK_CANCEL_OPTION,JOptionPane.INFO
                    RMATION_MESSAGE);
```

```
          }
          if(confirmar==0){
            obtener = Double.parseDouble((String.valueOf(Tablapre_
            venta.getValueAt(selecion,1))))*mul;
            total= total-obtener;
            textCobrar.setText(""+total);
            modelo7.removeRow(selecion);
            }else{
             cantidad= Integer.parseInt(JOptionPane.showInputDia
             log("Cuántos productos desea eliminar."));
             if(cantidad!=mul){
             int a=mul-cantidad;
             obtener = Double.parseDouble((String.valueOf
             (Tablapre
             _venta.getValueAt(selecion,1))))*cantidad;
             total= total-obtener;
             textCobrar.setText(""+total);
             Tablapre_venta.setValueAt(""+a, selecion, 2);
                 }else{
             obtener = Double.parseDouble((String.va
             lueOf(Tablapre_v
             enta.getValueAt(selecion,1))))*mul;

             total= total-obtener;
             textCobrar.setText(""+total);
             modelo7.removeRow(selecion);
             }
          }

    }else{
        int a=mul-cantidad;
        obtener = Double.parseDouble((String.valueOf(Tablapr
        e_venta.getValueAt(selecion,1))))*cantidad;
        total= total-obtener;
        textCobrar.setText(""+total);
        Tablapre_venta.setValueAt(""+a, selecion, 2);
       }

    }else{
        obtener = Double.parseDouble((String.valueOf(Tablapre_ve
        nta.getValueAt(selecion,1))))*mul;
        System.out.println(obtener);
        total= total-obtener;
        textCobrar.setText(""+total);
        modelo7.removeRow(selecion);
    }
}else{
// Si la tabla no contiene datos le indicamos al usua
rio que la tabla está vacía
```

```
            JOptionPane.showMessageDialog(null,"Tabla vacía");
        }
    }

    private void btnEliminarActionPerformed(java.awt.event.Acti
    onEvent evt) {
//Método que permite eliminar un producto

        Sonido();
        if (tablaInventario.getSelectedRow()!=-1) {
             String b= (String) tablaInventario.getValueAt(tablaInventario.getSel
             ectedRow(),0);
             m.eliminarModelo(bd, tablet, b);
            limpiaTabla(modelo1);
            consultarTelefono(modelo1,tablaInventario);
        } else {



           JOptionPane.showMessageDialog(null,"Error no hay fila seleccionada");
        }

    }

    private void jMenuItem3ActionPerformed(java.awt.event.Ac
    tionEvent evt) {
         // TablaAdministrador.dispose();
          TablaAdministrador.dispose();

//        Usuario.setBounds(384, 388, 384, 388);
//        Usuario.setModal(true);
//        Usuario.setVisible(true);
    }

    private void itemInventarioActionPerformed(java.awt.eve
    nt.ActionEvent evt) {
        TablaAdministrador.dispose();
        limpiaTabla(modelo1);
        consultarTelefono(modelo1,tablaInventario);
        Inventario.setBounds(1237, 560, 1237, 560);
        Inventario.setLocationRelativeTo(null);
        Inventario.setModal(true);
        Inventario.setVisible(true);

    }


    private void itemUsuariosActionPerformed(java.awt.event.Ac
    tionEvent evt) {
```

```
        TablaAdministrador.dispose();

        limpiaTabla(modelo4);
        consultarUsuarios();
        Administrarusuario.setBounds(700, 550, 700, 550);
        Administrarusuario.setLocationRelativeTo(null);
        Administrarusuario.setModal(true);
        Administrarusuario.setVisible(true);


    }

    private void itemReporteActionPerformed(java.awt.event.Action
    Event evt) {
        TablaAdministrador.dispose();
        Reporte.setBounds(384, 388, 675, 517);
        Reporte.setLocationRelativeTo(null);
        Reporte.setModal(true);
        Reporte.setVisible(true);


    }

    private void txtNombreKeyTyped(java.awt.event.KeyEvent evt) {
    // metodo que valida que solo se incerten letras
        m.MetodoValidarletras(evt);
    }
public void modificarUsuarios(String usuario, String correo){
    // realizamos  una modificacion a la tabla usuario en la base
    de datos
    try {
             Statement s=m.conectaBase(bd).createStatement();
             String uno="update "+tableu+" set usuario='"+usuario+"',correo='"+correo+"' where idU="+
             tedRow(), 0)+";";
             s.executeUpdate(uno);
             JOptionPane.showMessageDialog(null,"Datos modifica
             dos correctamente.");
             m.conectaBase(bd).close();
         } catch (SQLException ex) {
             System.out.println(ex);
         }
    }

    private void btnguardaruActionPerformed(java.awt.event.Actio
    nEvent evt) {

    //Validamos que se ayan sellecionados los checks box de la ventana modificar usuario
        if(!ingresarInventario.isSelected()){
            inventario1.setEnabled(false);
```

```
    }else{
        inventario1.setEnabled(true);
    }

    if(!agregarProducto.isSelected()){
        btnAgregar1.setEnabled(false);
    }
    else{
    btnAgregar1.setEnabled(true);

    }
    if(!eliminarProducto.isSelected()){
        btnEliminar1.setEnabled(false);
    }else{
        btnEliminar1.setEnabled(true);
    }

    if(!editarProducto.isSelected()){
        btnModificar1.setEnabled(false);
    }else{
       btnModificar1.setEnabled(true);
    }
    if(!ingresarHistorial.isSelected()){
        historialv.setEnabled(false);
    }else{
        historialv.setEnabled(true);
    }
 modificarUsuarios(txfusuario.getText(),txfcorreo.getText());
 Modificarusuario.dispose();
  limpiaTabla(modelo4);
  consultarUsuarios();
  Administrarusuario.setBounds(700, 550, 700, 550);
  Administrarusuario.setLocationRelativeTo(null);
  Administrarusuario.setModal(true);
  Administrarusuario.setVisible(true);

 }

 private void txtNombreuKeyTyped(java.awt.event.KeyEvent evt) {
   //Validamos que solo se puedan insertar número y letras
     m.MetodoValidarletras(evt);
 }

 private void txtContrasenauKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
     m.MetodoValidarletrasNumeros(evt);        // TODO add your han
     dling code here:
 }
```

```
    private void txtCorreoKeyTyped(java.awt.event.KeyEvent evt) {
 //Validamos que solo se puedan insertar número
 m.MetodoValidarNumeros(evt);      // TODO add your handling code here:
    }

    private void txtContrasenaKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
m.MetodoValidarletrasNumeros(evt);
    }

    private void txtTotalpKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }

    private void txtRecibidoKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }

    private void txtModeloKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtModeloKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
m.MetodoValidarletrasNumeros(evt);        // TODO add your handling code here:
    }

    private void txtMarcaKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número y letras
m.MetodoValidarletrasNumeros(evt);        // TODO add your handling code here:
    }

    private void txtRamKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }

    private void txtMemoriaKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }

    private void txtBateriaKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }
```

```
    private void txtPixelesKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling code here:
    }

    private void txtPreciovKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumerosfloat(evt);       // TODO add your handling
code here:
    }

    private void txtPreciocKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumerosfloat(evt);    // TODO add your handling
code here:
    }


    private void txtCantidadmKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling code
here:
    }

    private void txtMarcamKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarletrasNumeros(evt);        // TODO add your handl
ing code here:
    }

    private void txtRammKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling
code here:
    }

    private void txtMemoriamKeyTyped(java.awt.event.KeyEvent
    evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling
code here:
    }

    private void txtBateriamKeyTyped(java.awt.event.KeyEven
    t evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling
code here:
    }
```

```
    private void txtPixelesmKeyTyped(java.awt.event.KeyEvent
    evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling c
ode here:
    }

    private void txtcantmKeyTyped(java.awt.event.KeyEvent evt
    ) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumeros(evt);        // TODO add your handling co
de here:
    }

    private void txtcPreciovmKeyTyped(java.awt.event.KeyEvent
    evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumerosfloat(evt);        // TODO add your handli
ng code here:
    }

    private void txtPreciocmKeyTyped(java.awt.event.KeyEvent evt) {
//Validamos que solo se puedan insertar número
m.MetodoValidarNumerosfloat(evt);        // TODO add your handl
ing code here:
    }

    private void txfusuarioKeyTyped(java.awt.event.KeyEvent evt)
    {
//Validamos que solo se puedan incertar letras
m.MetodoValidarletras(evt);        // TODO add your handling c
ode here:
    }

    private void txfcorreoKeyTyped(java.awt.event.KeyEvent evt
    ) {
m.MetodoValidarcorreo(evt);        // TODO add your handling c
ode here:
    }

    private void jButton3ActionPerformed(java.awt.event.Action
    Event evt) {
   //Metodo que permite sellecionar un usuario
   //validamos que se aya seleccionado un usuario
     if (tablaUsuario.getSelectedRow()!=-1) {
             int u= (int) tablaUsuario.getValueAt(tablaUsuari
             o.getSelected
             Row(),0);
```

```
          System.out.println(u);
        //Eliminamos un usuario
          m.eliminarUsuario(bd,tableu,u);
         limpiaTabla(modelo4);
         consultarUsuarios();
    } else {

        JOptionPane.showMessageDialog(null,"Error no hay fila seleccionada.");
    }
}

private void btncancelaru1ActionPerformed(java.awt.event.Ac
tionEvent evt) {
    // TODO add your handling code here:
     //Metodo que despliega la ventana TablaAdministrador
     Modificarusuario.dispose();
     TablaAdministrador.setBounds(850, 535, 1400, 481);
     TablaAdministrador.setLocationRelativeTo(null);
     TablaAdministrador.setModal(true);
     TablaAdministrador.setVisible(true);
}

private void botonatrasRepActionPerformed(java.awt.event.Ac
tionEvent evt) {
    //Método que despliega la ventana  TablaAdministrador
    // TODO add your handling code here:
    if(usuario=="usuario"){
        Reporte.dispose();
        TablaEmpleado.setBounds(850, 535, 1400, 481);
        TablaEmpleado.setLocationRelativeTo(null);
        TablaEmpleado.setModal(true);
        TablaEmpleado.setVisible(true);
    }else{
        Reporte.dispose();
     TablaAdministrador.setBounds(850, 535, 1400, 481);
     TablaAdministrador.setLocationRelativeTo(null);
     TablaAdministrador.setModal(true);
     TablaAdministrador.setVisible(true);
    }

}

private void inventario1ActionPerformed(java.awt.even
t.ActionEvent evt) {
   //Método que despliega la ventana inventario1
    TablaEmpleado.dispose();
    Inventario1.setBounds(1237, 560, 1300, 560);
    Inventario1.setLocationRelativeTo(null);
    Inventario1.setModal(true);
```

```
    Inventario1.setVisible(true);

}

private void btnAgregar1ActionPerformed(java.awt.event.Actio
nEvent evt) {
    Inventario1.dispose();
   //Metodo que permite despliega la venta de Agregar producto

   //se limpian los campos de agregar producto
    txtMarcam.setText("");
    txtRamm.setText("");
    txtMemoriam.setText("");
    txtBateriam.setText("");
    txtPixelesm.setText("");
    txtcPreciovm.setText("");
    txtPreciocm.setText("");
    txtcantm.setText("");

    //despliega la venta agregar producto
    AgregarP.setBounds(552, 529, 414, 470);
    AgregarP.setLocationRelativeTo(null);
    AgregarP.setModal(true);
    AgregarP.setVisible(true);
}

private void btnModificar1ActionPerformed(java.awt.event.Act
ionEvent evt) {
    //Se validad que se aya seleccionado una fila para poster
    iormente desplegar la ventana Modifiacar producto
    if (tablaInventario3.getSelectedRow()!=-1) {
        Inventario1.dispose();
        ModificarP.setBounds(500, 518, 406, 450);
        ModificarP.setLocationRelativeTo(null);
        ModificarP.setModal(true);
        ModificarP.setVisible(true);

    } else {

       JOptionPane.showMessageDialog(null,"Error no hay fila seleccionada.");
    }
}

private void btnEliminar1ActionPerformed(java.awt.event.Acti
onEvent evt) {
   //Metdo que nos permite elinar un producto de la tabla tablaInventario3
   //se valida que la se aya seleccionado una fila
   if (tablaInventario3.getSelectedRow()!=-1) {
         String b= (String) tablaInventario3.getValueAt(tablaInventario3.getSele
```

```
            ctedRow(),0);
            m.eliminarModelo(bd, tablet, b);
           limpiaTabla(modelo6);
           consultarTelefono(modelo6,tablaInventario3);
        } else {

           JOptionPane.showMessageDialog(null,"Error no hay fila seleccionada.");
        }
    }

    private void btnAtrasinventario1ActionPerformed(java.awt.event
    .ActionEvent evt) {
      Inventario1.dispose();
      TablaEmpleado.setBounds(850, 535, 1400, 481);
      TablaEmpleado.setLocationRelativeTo(null);
      TablaEmpleado.setModal(true);
      TablaEmpleado.setVisible(true);
    }

    private void txtBuscar1ActionPerformed(java.awt.event.ActionEv
    ent evt) {
        // TODO add your handling code here:
    }



    private void jButton2ActionPerformed(java.awt.event.ActionE
    vent evt) {
       //Metodo que nos permite ir a la ventana de TablaAdmistrador
        Administrarusuario.dispose();
        TablaAdministrador.setBounds(850, 535, 1400, 481);
        TablaAdministrador.setLocationRelativeTo(null);
        TablaAdministrador.setModal(true);
        TablaAdministrador.setVisible(true);
    }

    private void txtBuscarKeyReleased(java.awt.event.KeyEv
    ent evt) {
        limpiaTabla(modelo1);
        consultarTelefono(modelo1,tablaInventario);
//      Creamos una variable y en ella guardamos lo que e
l usuario valla insertando
        String query=txtBuscar.getText().toLowerCase();
//      llamamos al metodo filter y le enviamos los para
metro
        filter(query,modelo1,tablaInventario);
    }

    private void txtBuscar1KeyReleased(java.awt.event.KeyEve
```

```
    nt evt) {
        limpiaTabla(modelo6);
        consultarTelefono(modelo6,tablaInventario3);
//      Creamos una variable y en ella guardamos lo que el usu
ario valla insertando
        String query=txtBuscar1.getText().toLowerCas
        e();
//      llamamos al metodo filter y le enviamos los param
etro
        filter(query,modelo6,tablaInventario3);
    }

    private void buscarUKeyReleased(java.awt.event.KeyEvent
    evt) {
        limpiaTabla(modelo4);
        consultarUsuarios();
//      Creamos una variable y en ella guardamos lo que el usuar
io valla insertando
        String query=buscarU.getText().toLowerCase();
//      llamamos al metodo filter y le enviamos los parametro
        filter(query,modelo4,tablaUsuario);
    }

    private void buscarUActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
    }

    private void buscarRActionPerformed(java.awt.event.ActionEven
    t evt) {
        // TODO add your handling code here:
    }

    private void buscarRKeyReleased(java.awt.event.KeyEvent evt) {
       // consultarReportes();
        limpiaTabla(modelo5);
        consultarReportes();
//      Creamos una variable y en ella guardamos lo que el usuario
valla insertando
        String query=buscarR.getText().toLowerCase();
//      llamamos al metodo filter y le enviamos los parametro
        filter(query,modelo5,tablaReporte);
    }


    private void tablaUsuarioMouseClicked(java.awt.event.MouseEvent
    evt) {
        int seleccionar = tablaUsuario.rowAtPoint(evt.getPoint());
        txfusuario.setText(String.valueOf(tablaUsuario.getValueAt(s
```

```
        eleccionar,1)));
        txfcorreo.setText(String.valueOf(tablaUsuario.getValueAt(se
        leccionar,2)));
    }

    private void tablaInventarioMouseClicked(java.awt.event.MouseEv
    ent evt) {
        int seleccionar = tablaInventario.rowAtPoint(evt.getPoin
        t());
        txtMarcam.setText(String.valueOf(tablaInventario.getValue
        At(seleccionar,1)));
        txtRamm.setText(String.valueOf(tablaInventario.getValueAt(
        seleccionar,2)));
        txtMemoriam.setText(String.valueOf(tablaInventario.getVal
        ueAt(seleccionar,3)));
        txtBateriam.setText(String.valueOf(tablaInventario.getValu
        eAt(seleccionar,4)));
        txtPixelesm.setText(String.valueOf(tablaInventario.getVa
        lueAt(seleccionar,5)));
        txtcPreciovm.setText(String.valueOf(tablaInventario.getVa
        lueAt(seleccionar,6)));
        txtPreciocm.setText(String.valueOf(tablaInventario.getValu
        eAt(seleccionar,7)));
        txtcantm.setText(String.valueOf(tablaInventario.getValueA
        t(seleccionar,8)));

    }

    private void tfCodigoKeyReleased(java.awt.event.KeyEvent evt) {
        limpiaTabla(modelo2);
        consultarTelefonoventa(modelo2,tablaInventario1);
//      Creamos una variable y en ella guardamos lo que el usuari
o valla insertando
        String query=tfCodigo.getText().toLowerCase();
//      llamamos al metodo filter y le enviamos los parametro
        filter(query,modelo2,tablaInventario1);
    }

    private void txtCodigoKeyReleased(java.awt.event.KeyEvent evt) {
        m.MetodoValidarletras(evt);
        limpiaTabla(modelo3);
        consultarTelefonoventa(modelo3,tablaInventario2);
//      Creamos una variable y en ella guardamos lo que el usua
rio valla insertando
        String query=txtCodigo.getText().toLowerCase();
//      llamamos al metodo filter y le enviamos los parametro
        filter(query,modelo3,tablaInventario2);
    }
```

```
private void btnCobrarpActionPerformed(java.awt.event.Actio
nEvent evt) {
   //Método que permite realizar una venta
    Sonido();

   String fecha=fd.getText();
    // se valida que no aya espacios vacíos
   if(txtRecibido.getText().isEmpty()){
      JOptionPane.showMessageDialog(null,"No ha insertado
      la cantidad recibida");
   }else{

   String Reci = txtRecibido.getText();
   double  Recibido = Double.parseDouble(Reci);
   if(Recibido >= total){
    Double cobrar=Recibido-total;
      //se validad de que sea usuario o administrador
       if(usuario=="usuario"){

           //Se valida si realmente se quiere realizar
           la venta del producto

           int confirmar=JOptionPane.showConfirmDialog(null, "CONFIRMAR VENTA",  "TELEFONOS", JOp
           ORMATION_MESSAGE);
           if(confirmar==0){
                JOptionPane.showMessageDialog(null," El cambio es:"+cobrar);
               for (int i = 0; i < rou; i++) {
                   m.modificarCantidad(bd, cd[i], mode[i]);
                   m.insertarVenta(bd, tablev, mode[i], cd[i],fecha);
               }
               total=0.0;
               cp1.setValue(0);
               Cobrar.dispose();
               JOptionPane.showMessageDialog(null, "Venta exitosa");
               limpiaTabla(modelo3);
               limpiaTabla(modelo8);
               txtTotal.setText("");
               TablaEmpleado.setBounds(850, 535, 1400, 481);
               TablaEmpleado.setLocationRelativeTo(null);
               TablaEmpleado.setModal(true);
               TablaEmpleado.setVisible(true);
           }else{
               Cobrar.dispose();
               JOptionPane.showMessageDialog(null, "Venta cancelada");
               total=0.0;
               cp1.setValue(0);
                txtTotal.setText("");
               TablaEmpleado.setBounds(850, 535, 1400, 481);
               TablaEmpleado.setLocationRelativeTo(null);
```

```
                TablaEmpleado.setModal(true);
                TablaEmpleado.setVisible(true);
            }
        }else{
            int confirmar=JOptionPane.showConfirmDialog(
            null, "CONFIRMAR VENTA",  "TELEFONOS", JOptionPane.OK_CANCEL_OPTION,JOptionPane.INF
            ORMATION_MESSAGE);
            if(confirmar==0){
                 JOptionPane.showMessageDialog(null," El cambio es:"+cobrar);
                for (int i = 0; i < rou; i++) {
                    m.modificarCantidad(bd, cd[i], mode[i]);
                    m.insertarVenta(bd, tablev, mode[i], cd[i],fecha);
                }
                JOptionPane.showMessageDialog(null, "Venta exitosa");
                limpiaTabla(modelo2);
                limpiaTabla(modelo7);
                textCobrar.setText("");
                txtRecibido.setText("");
                total=0.0;
                pc.setValue(0);
                Cobrar.dispose();
                TablaAdministrador.setBounds(850, 535, 1400, 481);
                TablaAdministrador.setLocationRelativeTo(null);
                TablaAdministrador.setModal(true);
                TablaAdministrador.setVisible(true);
            }  else{
                JOptionPane.showMessageDialog(null,"Venta cancelada");
                limpiaTabla(modelo2);
                limpiaTabla(modelo7);
                Cobrar.dispose();
                total=0.0;
                TablaAdministrador.setBounds(850, 535, 1400, 481);
                TablaAdministrador.setLocationRelativeTo(null);
                TablaAdministrador.setModal(true);
                TablaAdministrador.setVisible(true);

            }
        }
    }
  else{//En caso de que la cantodad recibi sea menor que la cantidad total del producto se le not
      m.mensajeActualizado(mensajeReci,"Cantidad incorrecta");
     }
}

}


public int validarinicio(){
  //Validamos que los datos incertaso en la ventana de LOGEO sean correctos
```

```
        validar=0;
        if(txtNombreu.equals("")||txtContrasenau.equals("")){
            validar=1;
        }
     JOptionPane.showMessageDialog(null, validar);
        return validar;
    }



    private void tablaInventario3MouseClicked(java.awt.event.Mou
    seEvent evt) {
// caon el metodo mauseclicked al sellecionar un producto se pi
nta dichos producto en  los texfield
        int seleccionar = tablaInventario3.rowAtPoint(evt.getP
        oint());
        txtMarcam.setText(String.valueOf(tablaInventario3.getV
        alueAt(seleccionar,1)));
        txtRamm.setText(String.valueOf(tablaInventario3.getVa
        lueAt(seleccionar,2)));
        txtMemoriam.setText(String.valueOf(tablaInventario3.g
        etValueAt(seleccionar,3)));
        txtBateriam.setText(String.valueOf(tablaInventario3.ge
        tValueAt(seleccionar,4)));
        txtPixelesm.setText(String.valueOf(tablaInventario3.ge
        tValueAt(seleccionar,5)));
        txtcPreciovm.setText(String.valueOf(tablaInventario3.g
        etValueAt(seleccionar,6)));
        txtPreciocm.setText(String.valueOf(tablaInventario3.ge
        tValueAt(seleccionar,7)));
        txtcantm.setText(String.valueOf(tablaInventario3.getVa
        lueAt(seleccionar,8)));
    }

    private void tfCodigoActionPerformed(java.awt.event.Action
    Event evt) {
        // TODO add your handling code here:
    }

    private void tablaInventario1MouseClicked(java.awt.event.M
    ouseEvent evt) {
         //Integer a=Integer.parseInt(cant.getText());// Obtene
         r la cantidad del producto
         int valor=(Integer)pc.getValue();
         int fila= Tablapre_venta.getRowCount();// fila de l
         a tabla de venta
         int columna= Tablapre_venta.getColumnCount(); // col
         umna de la tabla venta
         int seleccionar = tablaInventario1.rowAtPoint(evt.get
```

```
        Point());
        int canp=  Integer.parseInt(String.valueOf(tablaInven
        tario1.getValueAt(seleccionar,3)));
        modelo =  String.valueOf(tablaInventario1.getValueAt
        (seleccionar,0));
        String array[]=new String[fila];
      for(int i=0;i<fila;i++){// obtenemos los modelos
      array[i]= String.valueOf(Tablapre_venta.getValueAt(i,0));
    }
      for(int i=0;i<fila;i++){
          if(modelo.equals(array[i])){
              cant2=Integer.parseInt(String.valueOf(Tablapre_
              venta.getValueAt(i,2)));
          }
      }
      // se verica que la cantidad de producto sea menor mayor
      que cero y que sea menor que la cantidad existente
      if(valor==0||valor>canp){
        JOptionPane.showMessageDialog(null, "Verifique que la
        cantidad a insertar sea menor al producto existente y que sea di
        ferente de cero.");
      }else{
        if(canp>=(cant2+valor)){
        resta=Double.parseDouble(String.valueOf(tablaInventario1.
        getValueAt(seleccionar,2)))*valor;
        total = total + resta;
        textCobrar.setText(""+total);
         for(int i=0;i<fila;i++)
        if(modelo.equals(array[i])){
          // cantidad de la tabla principla
          cant2=Integer.parseInt(String.valueOf(Tablapre_venta
          .getValueAt(i,2)));//cantidad que tenemos almacena de ventas
          modelo7.removeRow(i);
        }
         cant1= valor;
         cant1=cant1+cant2;

    consultarTelefonoventa2(modelo7,Tablapre_venta,modelo);
    }else{
            JOptionPane.showMessageDialog(null,"Producto en ex
            istencia "+canp +"\nSeleccionados hasta el momento
            "+cant2);
            cant2=0;
        }
  }
}
  private void tablaInventario2MouseClicked(java.awt.event.Mo
  useEvent evt) {
      // tabla empleado
```

```
 int valor=(Integer)cp1.getValue();
   int fila= Tablapre_venta1.getRowCount();// fila de la
   tabla de venta
   int columna= Tablapre_venta1.getColumnCount(); // colum
   na de la tabla venta
   int seleccionar = tablaInventario2.rowAtPoint(evt.getPoin
   t());
   int canp=  Integer.parseInt(String.valueOf(tablaInventari
   o2.getValueAt(seleccionar,3)));
   modelo =  String.valueOf(tablaInventario2.getValueAt(selec
   ionar,0));
   String array[]=new String[fila];
  for(int i=0;i<fila;i++){// obtenemos los modelos
  array[i]= String.valueOf(Tablapre_venta1.getValueAt(i,0
  ));

}
  for(int i=0;i<fila;i++){// Obtenos la cantidad del producto
      if(modelo.equals(array[i])){
          cant2=Integer.parseInt(String.valueOf(Tablapre_v
          enta1.getValueAt(i,2)));
      }
  }


  if(valor==0||valor>canp){

    JOptionPane.showMessageDialog(null, "Verifique que
    la cantidad a insertar sea menor al producto existent
    e y que sea diferente de cero.");

  }else{
    if(canp>=(cant2+valor)){

    resta=Double.parseDouble(String.valueOf(tablaInventario2.g
    etValueAt(seleccionar,2)))*valor;
    total = total + resta;
    txtTotal.setText(""+total);
     for(int i=0;i<fila;i++)
    if(modelo.equals(array[i])){
      // cantidad de la tabla principla
      cant2=Integer.parseInt(String.valueOf(Tablapre_vent
      a1.getValueAt(i,2)));//cantidad que tenemos almacena de ventas
      modelo8.removeRow(i);
    }
     cant1= valor;
     cant1=cant1+cant2;
consultarTelefonoventa2(modelo8,Tablapre_venta1,modelo);
}else{
```

```
          JOptionPane.showMessageDialog(null,"Producto en existencia "+canp +"\nSeleccionados has
          cant2=0;
      }
}


}

private void btnCancelarproductorActionPerformed(java.awt.e
vent.ActionEvent evt) {
    Sonido();
    // Metodo (boton) eliminar de la tabla empleado, perimi
    te eliminar un producto al seleccionar el producto
    int selecion = Tablapre_venta1.getSelectedRow();
     int mul,cantidad;

    //Seleccionamos la fila con getSelectedRow() para despue
    s eliminarla.
     if(Tablapre_venta1.getSelectedRow()!=-1){
        mul= Integer.parseInt(((String.valueOf(Tablapre_vent
        a1.getValueAt(selecion,2)))));
        if(mul!=1){//Validamos que sea diferente de  para p
        odteriormente perdir los productos a eliminar
           cantidad= Integer.parseInt(JOptionPane.showInputDialog("
           Cuántos producto desea eliminar"));
           if(cantidad >=mul){//Se valida que el producto
           sea mayor o igual
              if(cantidad>mul){//si es mayor se pide se s
              e desea eliminar todos
                confirmar=JOptionPane.showConfirmDialog(n
                ull, "La cantidad a eliminar es mayor al exietent
                e, desea eliminar todos los productos ",  "TELEFONOS", JOptionPane.OK_CANCEL_OPTI
                .INFORMATION_MESSAGE);

              }

              if(confirmar==0){// confirmamos la eliminacion
                obtener = Double.parseDouble((String.valueOf(Tabla
                pre_venta1.getValueAt(selecion,1))))*mul;
                total= total-obtener;
                txtTotal.setText(""+total);
                modelo8.removeRow(selecion);
                }else{
                 cantidad= Integer.parseInt(JOptionPane.showInputD
                 ialog("Cuántos producto desea eliminar"));

                 if(cantidad!=mul){
                 int a=mul-cantidad;
```

```
                obtener = Double.parseDouble((String.valueOf(Tabla
                pre_venta1.getValueAt(selecion,1))))*cantidad;
                total= total-obtener;
                txtTotal.setText(""+total);
                Tablapre_venta1.setValueAt(""+a, selec
                ion, 2);
                    }else{
                obtener = Double.parseDouble((String.valueOf(Tabl
                apre_venta1.getValueAt(selecion,1))))*mul;
                total= total-obtener;
                txtTotal.setText(""+total);
                modelo8.removeRow(selecion);
                }
            }

        }else{
            int a=mul-cantidad;
            obtener = Double.parseDouble((String.valueOf(Tablapre
            _venta1.getValueAt(selecion,1))))*cantidad;
            total= total-obtener;
            txtTotal.setText(""+total);
            Tablapre_venta1.setValueAt(""+a, selecion, 2);
           }

        }else{
            obtener = Double.parseDouble((String.valueOf(Tablapre
            _venta1.getValueAt(selecion,1))))*mul;
            System.out.println(obtener);
            total= total-obtener;
            txtTotal.setText(""+total);
            modelo8.removeRow(selecion);
        }
    }else{
    // Si la tabla no contiene datos le indicamos al usu
    ario que la tabla está vacía
        JOptionPane.showMessageDialog(null,"Tabla vacía");
    }



}

private void historialvActionPerformed(java.awt.event.
ActionEvent evt) {
    TablaAdministrador.dispose();
    Reporte.setBounds(384, 388, 675, 517);
    Reporte.setLocationRelativeTo(null);
    Reporte.setModal(true);
    Reporte.setVisible(true);
```

```
  }

  private void formWindowClosing(java.awt.event.WindowEvent
  evt) {
    cerrar();
  }

  private void txtNombreuMouseClicked(java.awt.event.MouseEv
  ent evt) {
     this.txtNombreu.setText("");
  }

  private void txtContrasenauMouseClicked(java.awt.event.Mouse
  Event evt) {
   this.txtContrasenau.setText("");
  }

  private void aceptarAdministradorMouseClicked(java.awt.event
  .MouseEvent evt) {
  Sonido();
  }

  private void txtConfirmarKeyTyped(java.awt.event.KeyEvent
  evt) {
     //Metodo que nos permite validar que se incerten solo nú
     mero
      m.MetodoValidarletrasNumeros(evt);        // TODO add y
      our handling code here:
  }

  private void txtConfirmarActionPerformed(java.awt.event.Ac
  tionEvent evt) {
      // TODO add your handling code here:
  }

  private void btnExcelActionPerformed(java.awt.event.ActionE
  vent evt) {
//Metodo que nos permite extraer el registro de los producto
s vendidos y que nos lo envia a un archivo excel
      javax.swing.JFileChooser fc=new javax.swing.JFileChooser
      ();
      java.io.OutputStream out=null;
      java.io.File file=null;
      if(tablaReporte.getRowCount()>0){
      try {
          if(fc.showSaveDialog(this)==javax.swing.JFileChooser.AP
          PROVE_OPTION){
```

```
            file = fc.getSelectedFile();
            out = new java.io.FileOutputStream(file.getAbsolut
            ePath() + ".xlsx");
            XSSFWorkbook libro = new XSSFWorkbook();
            XSSFSheet hoja=libro.createSheet("Reporte venta de teléfonos");
            XSSFRow registro=null;
           javax.swing.table.DefaultTableModel tm=(javax.swing.table.DefaultTableModel)tablaRepor
           getModel();
            registro=hoja.createRow(0);
            registro=hoja.createRow(1);
            registro.createCell((short)5).setCellValue(" "+"SIS
            TEMA DE VENTAS SFH");
            registro=hoja.createRow(2);
            registro.createCell((short)5).setCellValue(" "+"RE
            PORTE DE VENTAS");
            registro=hoja.createRow(3);
            for (int i=0 ; i<tm.getColumnCount() ; i++) {
                registro.createCell((short)i).setCellValue(tm.ge
                tColumnName(i));
            }
            for (int i=0; i<tm.getRowCount() ; i++) {
                registro = hoja.createRow(i+4);
                for (int j=0 ; j<tm.getColumnCount() ; j++) {
                    Object v1=tm.getValueAt(i,j);
                    if (v1== null){
                        v1=" ";
                    }
                    registro.createCell((short)j).setCellValue(
                    ""+v1);
                }
            }
            try {
                libro.write(out);
                if(out != null){
                    javax.swing.JOptionPane.showConfirmDialog(nu
                    ll,"Archivo creado"," ",javax.swing.JOptionPane.CLOSED_OPTION);
                }
            } catch (java.io.IOException ex) {
            }
        }
        else {}
    } catch (FileNotFoundException ex) {
         System.out.println("No hay archivo");
    }
}
else
    javax.swing.JOptionPane.showMessageDialog(this,"No tiene da
    tos para exportar","Advertencia",javax.swing.JOptionPane.ERRO
    R_MESSAGE);
```

```
}

private void txtTotalKeyTyped(java.awt.event.KeyEvent evt) {
    m.MetodoValidarNumeros(evt);// validamos que en el campo se
    ingresen solo números
}

private void textCobrarKeyTyped(java.awt.event.KeyEvent evt) {
     m.MetodoValidarNumeros(evt);
}

//Metodo que nos ayuda a verificar si el usuario realmente quiera
salir
public void cerrar(){
    String botones[]={"Cerrar","Cancelar"};
    int eleccion=JOptionPane.showOptionDialog(this,"¿Desea usted
    cerrar la aplicación?","Confirmación", 0, 0, null, botones, this);
    if (eleccion==JOptionPane.YES_OPTION){
        System.exit(0);
    }else if(eleccion==JOptionPane.NO_OPTION){
        System.out.println("Se canceló el cierre");
    }
}

public void horaActual(){
  //Método que nos permite realizar un reloj utilizando la clase Jcalendar
    Calendar ca =new GregorianCalendar();
   Date da = new Date();
   ca.setTime(da);

   hora=ca.get(Calendar.HOUR_OF_DAY)>9?""+ca.get(Calendar.HOUR_OF_DAY):"0"+ca.get(Calendar.HOUR_O
   minutos=ca.get(Calendar.MINUTE)>9?""+ca.get(Calendar.MINUTE):"0"+ca.get(Calendar.MINUTE);
   segundos=ca.get(Calendar.SECOND)>9?""+ca.get(Calendar.SECOND):"0"+ca.get(Calendar.SECOND);
}
public void limpiar(){
    //Metodo que nos permite limpiar la los campos Rexfield
    txtMarcam.setText("");
    txtRamm.setText("");
    txtMemoriam.setText("");
    txtBateriam.setText("");
    txtPixelesm.setText("");
    txtcPreciovm.setText("");
    txtPreciocm.setText("");
    txtcantm.setText("");
}

public void limpiarisertar(){
```

```
    //Metodo que nos permite limpiar la los campos Rexfield
    txtModelo.setText("");
    txtMarca.setText("");
    txtRam.setText("");
    txtMemoria.setText("");
    txtBateria.setText("");
    txtPixeles.setText("");
    txtPreciov.setText("");
    txtPrecioc.setText("");
    txtCantidadm.setText("");
}

//Metodo que nos ayuda a buscar y mostrar un dato en la tabla
public void filter(String query,DefaultTableModel modelo, JTable Tabla ){
    TableRowSorter<DefaultTableModel> tr= new TableRowSorter<DefaultTableModel>(modelo);
    Tabla.setRowSorter(tr);
    tr.setRowFilter(RowFilter.regexFilter(query));
}

public void Sonido(){
   //Metodo que nos permite darle sonido cada véz que le demos clic a un boton
    try {
     AudioClip sound;
     sound=java.applet.Applet.newAudioClip(getClass().getResource("/audio/BtnSelect.wav"));
     sound.play();

    } catch (Exception e) {
        System.err.println("Error: "+e);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    try {
        UIManager.setLookAndFeel(new com.jtattoo.plaf.mint.MintLookAndFeel());
    } catch (Exception e) {
        System.out.println("");
    }

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Ventas().setVisible(true);
        }
    });
```

```
    RelojModeloUtil modelo = new RelojModeloUtil();
     modelo.addObserver (new Observer()
     {
         public void update (Observable unObservable, Object dato)
         {



                 try{


                     if (ban){
                         letrero.setText(a2.substring(0, a2.length()-cont)+ a.substring(0, cont));
                         letrero2.setText(a2.substring(0, a2.length()-cont)+ a.substring(0, cont))
                         cont++;
                      }
                     else{
                         letrero.setText(a.substring(cont, a.length()) + a2.substring(0, cont) );
                         letrero2.setText(a2.substring(0, a2.length()-cont)+ a.substring(0, cont))
                         cont++;
                     }

                     if (cont==a.length()){
                         if (cont==a.length() && ban== false){
                             cont=0;
                             ban=true;
                         }else{
                             cont=0;
                             ban=false;
                         }
                     }
                 }catch(Exception e){}
         }
     });
}

// Variables declaration - do not modify
private javax.swing.JDialog Administrarusuario;
private javax.swing.JDialog AgregarP;
private javax.swing.JDialog Cobrar;
private javax.swing.JDialog Inventario;
private javax.swing.JDialog Inventario1;
private javax.swing.JMenuBar Menu;
private javax.swing.JDialog ModificarP;
private javax.swing.JDialog Modificarusuario;
private javax.swing.JTextField NoP;
private javax.swing.JDialog Registrar;
private javax.swing.JDialog Reporte;
private javax.swing.JDialog TablaAdministrador;
```

```
private javax.swing.JDialog TablaEmpleado;
public javax.swing.JTable Tablapre_venta;
public javax.swing.JTable Tablapre_venta1;
private javax.swing.JDialog Usuario;
private javax.swing.JButton aceptarAdministrador;
private javax.swing.JCheckBox agregarProducto;
private javax.swing.JButton atrasBtn;
private javax.swing.JButton botonCobrar;
private javax.swing.JButton botonEliminarp;
private javax.swing.JButton botonatrasRep;
private javax.swing.JButton btnAgregar;
private javax.swing.JButton btnAgregar1;
private javax.swing.JButton btnAtras;
private javax.swing.JButton btnAtrasinventario;
private javax.swing.JButton btnAtrasinventario1;
private javax.swing.JButton btnCamcelarpro;
private javax.swing.JButton btnCamcelarpro1;
private javax.swing.JButton btnCancelarp;
private javax.swing.JButton btnCancelarproductor;
private javax.swing.JButton btnCobrar;
private javax.swing.JButton btnCobrarp;
private javax.swing.JButton btnEliminar;
private javax.swing.JButton btnEliminar1;
private javax.swing.JButton btnExcel;
private javax.swing.JButton btnGuardar;
private javax.swing.JButton btnGuardarp;
private javax.swing.JButton btnGuardarp1;
private javax.swing.JButton btnModificar;
private javax.swing.JButton btnModificar1;
private javax.swing.JButton btncancelaru1;
private javax.swing.JButton btnguardaru;
private javax.swing.JTextField buscarR;
private javax.swing.JTextField buscarU;
private javax.swing.JLabel c;
private javax.swing.JPanel colorPanel;
private javax.swing.JPasswordField contraAdmin;
private javax.swing.JSpinner cp1;
private javax.swing.JCheckBox editarProducto;
private javax.swing.JCheckBox eliminarProducto;
private javax.swing.JLabel fd;
private javax.swing.JButton historialv;
private javax.swing.JLabel hora2;
private javax.swing.JLabel imagenAgregar;
private javax.swing.JLabel imagenAgregar1;
private javax.swing.JLabel imagenEliminar;
private javax.swing.JLabel imagenEliminar1;
private javax.swing.JLabel imagenLogo;
private javax.swing.JLabel imagenModificar;
private javax.swing.JLabel imagenModificar1;
```

```
private javax.swing.JPanel imagenPanel;
private javax.swing.JLabel imagenUsuario;
private javax.swing.JLabel imgContrasena;
private javax.swing.JLabel imgNombre;
private javax.swing.JCheckBox ingresarHistorial;
private javax.swing.JCheckBox ingresarInventario;
private javax.swing.JMenuItem iniciarSesion;
private javax.swing.JButton inventario1;
private javax.swing.JMenuItem itemInventario;
private javax.swing.JMenuItem itemReporte;
private javax.swing.JMenuItem itemUsuarios;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
```

```
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator11;
private javax.swing.JSeparator jSeparator12;
private javax.swing.JSeparator jSeparator13;
private javax.swing.JSeparator jSeparator14;
private javax.swing.JSeparator jSeparator15;
private javax.swing.JSeparator jSeparator16;
private javax.swing.JSeparator jSeparator17;
private javax.swing.JSeparator jSeparator18;
private javax.swing.JSeparator jSeparator19;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator20;
private javax.swing.JSeparator jSeparator21;
private javax.swing.JSeparator jSeparator22;
private javax.swing.JSeparator jSeparator23;
private javax.swing.JSeparator jSeparator24;
private javax.swing.JSeparator jSeparator25;
private javax.swing.JSeparator jSeparator26;
private javax.swing.JSeparator jSeparator27;
private javax.swing.JSeparator jSeparator28;
private javax.swing.JSeparator jSeparator29;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator30;
private javax.swing.JSeparator jSeparator31;
private javax.swing.JSeparator jSeparator32;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JSeparator jSeparator5;
private javax.swing.JSeparator jSeparator6;
private javax.swing.JSeparator jSeparator7;
private javax.swing.JSeparator jSeparator8;
private javax.swing.JSeparator jSeparator9;
private javax.swing.JToolBar jToolBar1;
private javax.swing.JLabel labelTotal;
private javax.swing.JLabel lblBuscador;
private javax.swing.JLabel lblBuscador1;
private javax.swing.JLabel lblCantidad;
private javax.swing.JLabel lblCantidadp;
private javax.swing.JLabel lblCantidadp1;
private javax.swing.JLabel lblCodigo;
private javax.swing.JLabel lblCodigop;
private javax.swing.JLabel lblConfirmar;
private javax.swing.JLabel lblContrasena;
```

```
private javax.swing.JLabel lblContraseñau;
private javax.swing.JLabel lblCorreo;
private javax.swing.JLabel lblImagen;
private javax.swing.JLabel lblLetrero;
private javax.swing.JLabel lblLetrero1;
private javax.swing.JLabel lblMarcap;
private javax.swing.JLabel lblMarcap1;
private javax.swing.JLabel lblNombre;
private javax.swing.JLabel lblNombrep;
private javax.swing.JLabel lblNombrep1;
private javax.swing.JLabel lblNombreu;
private javax.swing.JLabel lblPrecioc;
private javax.swing.JLabel lblPrecioc1;
private javax.swing.JLabel lblPreciov;
private javax.swing.JLabel lblPreciov1;
private javax.swing.JLabel lblRecibido;
private javax.swing.JLabel lblTotal;
private javax.swing.JLabel letrasLogo;
public static javax.swing.JLabel letrero;
public static javax.swing.JLabel letrero2;
private javax.swing.JLabel logo;
private javax.swing.JLabel mensajeA;
private javax.swing.JLabel mensajeReci;
private javax.swing.JMenu menuAtras;
private javax.swing.JMenu menuInicio;
private javax.swing.JMenu menuPermisos;
private javax.swing.JPanel panelAdministrador;
private javax.swing.JPanel panelAgregarp;
private javax.swing.JPanel panelAgregarp1;
private javax.swing.JPanel panelCambios;
private javax.swing.JPanel panelCambios1;
private javax.swing.JPanel panelCobrar;
private javax.swing.JPanel panelColor;
private javax.swing.JPanel panelEmpleado;
private javax.swing.JPanel panelImagen;
private javax.swing.JPanel panelImagen1;
private javax.swing.JPanel panelInicio;
private javax.swing.JPanel panelInventario;
private javax.swing.JPanel panelInventario1;
private javax.swing.JPanel panelProducto;
private javax.swing.JPanel panelProductos;
private javax.swing.JPanel panelRegistro;
private javax.swing.JPanel panelUsuario;
private javax.swing.JSpinner pc;
private javax.swing.JMenuItem registrarse;
private javax.swing.JLabel relog;
private javax.swing.JScrollPane scrollpaneInventario;
private javax.swing.JScrollPane scrollpaneInventario1;
private javax.swing.JScrollPane scrollpaneInventario2;
```

```
private javax.swing.JScrollPane scrollpaneInventario3;
public static javax.swing.JTable tablaInventario;
private javax.swing.JTable tablaInventario1;
private javax.swing.JTable tablaInventario2;
public static javax.swing.JTable tablaInventario3;
private javax.swing.JTable tablaReporte;
private javax.swing.JTable tablaUsuario;
private javax.swing.JTextField textCobrar;
private javax.swing.JTextField tfCodigo;
private javax.swing.JLabel totalLbl;
private javax.swing.JTextField txfcorreo;
private javax.swing.JTextField txfusuario;
private javax.swing.JTextField txtBateria;
private javax.swing.JTextField txtBateriam;
private javax.swing.JTextField txtBuscar;
private javax.swing.JTextField txtBuscar1;
private javax.swing.JTextField txtCantidadm;
private javax.swing.JTextField txtCodigo;
private javax.swing.JPasswordField txtConfirmar;
private javax.swing.JPasswordField txtContrasena;
private javax.swing.JPasswordField txtContrasenau;
private javax.swing.JTextField txtCorreo;
private javax.swing.JTextField txtMarca;
private javax.swing.JTextField txtMarcam;
private javax.swing.JTextField txtMemoria;
private javax.swing.JTextField txtMemoriam;
private javax.swing.JTextField txtModelo;
private javax.swing.JTextField txtNombre;
private javax.swing.JTextField txtNombreu;
private javax.swing.JTextField txtPixeles;
private javax.swing.JTextField txtPixelesm;
private javax.swing.JTextField txtPrecioc;
private javax.swing.JTextField txtPreciocm;
private javax.swing.JTextField txtPreciov;
private javax.swing.JTextField txtRam;
private javax.swing.JTextField txtRamm;
private javax.swing.JTextField txtRecibido;
private javax.swing.JTextField txtTotal;
private javax.swing.JTextField txtTotalp;
private javax.swing.JTextField txtcPreciovm;
private javax.swing.JTextField txtcantm;
// End of variables declaration


public void run() {
     for (int i = 0; i < 10; i++) {

    i=i%4;
    try{
```

```
            sleep(10);
        }catch(Exception ex){
            System.out.println("Error"+ex.getMessage());
        }
        horaActual();
        relog.setText(hora+":"+minutos+":"+segundos);
        hora2.setText(hora+":"+minutos+":"+segundos);
    }
    }


    public static class RelojModeloUtil extends Observable
 {
     /**
      * Lanza un timer cada segundo.
      */
     public RelojModeloUtil()
     {
         Timer timer = new Timer();
         timer.scheduleAtFixedRate(timerTask, 0, 100);

     }

      /**
      * Clase que se mete en Timer, para que se le avise cada segundo.
      */
     TimerTask timerTask = new TimerTask()
     {
         /**
          * Método al que Timer llamará cada segundo. Se encarga de avisar
          * a los observadores de este modelo.
          */
         public void run()
         {
             setChanged();
             notifyObservers(new Date());

         }
     };
}
}
```

Codigo de clase Metodos

```
package venta;

import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.KeyEvent;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTable;

/**
 *@author SFH
 *Almaraz Vargas Hermilo
 *González Ruiz Floriceli
 *Lagunas Gonzalez Stephanie Jacqueline
 */
public class Metodos {

    String pass = "milo";
    String user = "postgres";
    String host = "localhost";
    Connection con1 = null;
    String nom;
    ResultSet rs = null;
    int resu;
    public Connection conectaServer() {
    //se realiza la coneción a la base de dtos
        Connection con = null;
        try {
            Class.forName("org.postgresql.Driver");
            con = DriverManager.getConnection("jdbc:postgresql:
            //" + host + "/", user, pass);
            System.out.println("conexión establecida");

        } catch (Exception ex) {
            System.out.println("Error al conectar:" + ex.getMessage());
        }
        return con;
    }
  // se coneta a la base de datos
    public Connection conectaBase(String bd) {
        Connection con = null;
        try {
            Class.forName("org.postgresql.Driver");
            con = DriverManager.getConnection("jdbc:postgresql:
```

```
            //" + host + "/" + bd, user, pass);
            System.out.println("conexión establecida");

        } catch (Exception ex) {
            System.out.println("Error al conectar:" + ex.getMessage());
        }
        return con;
    }
// perimite almacenar el resultado de una consulta.
    public ResultSet ejecutaConsulta(String sql, String bd) {
        try {
            ResultSet rs = null;
            Connection con = null;
            con = conectaBase(bd);
            Statement st = con.createStatement();
            rs = st.executeQuery(sql);
        } catch (Exception e) {
            System.out.println("Error al conectar" + e.getMessage());

        }

        return rs;
    }
// Se incertan los modelos de los productos
    public void insertarModelo(String bd, String table, String id_modelo, String marca, String ram, S

        try {
            Statement s = conectaBase(bd).createStatement();
            Double prev = Double.parseDouble(pre_ven);
            Double prec = Double.parseDouble(pre_com);
            PreparedStatement ps = conectaBase(bd).prepareStatement
            ("INSERT INTO " + table + "(id_modelo,marca,ram,memoriaint,bateria,pixeles,pre_ven,
            pre_comp,cantidad)values (?,?,?,?,?,?,?,?,?)");
            ps.setString(1, id_modelo);
            ps.setString(2, marca);
            ps.setInt(3, Integer.parseInt(ram));
            ps.setInt(4, Integer.parseInt(memoriaint));
            ps.setInt(5, Integer.parseInt(bateria));
            ps.setInt(6, Integer.parseInt(pixeles));
            ps.setDouble(7, prev);
            ps.setDouble(8, prec);
            ps.setInt(9, Integer.parseInt(cantidad));
            ps.executeUpdate();
            conectaBase(bd).close();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

    }
```

```
// Con este método se insertan los producto a la base de datos
public void insertarUsuario(String bd, String table, String
nombre,
String correo, String contrasena) {

    try {
        //Con PrepareStantement Se prepara una unica sentencia especificando los campos de la tab
        //Para no realizar multiples consultas
        PreparedStatement ps = conectaBase(bd).prepareStatement
        ("INSERT INTO " + table +
                                "(usuario,correo,contrasena)values (?,?,?)");
        ps.setString(1, nombre);
        ps.setString(2, correo);
        ps.setString(3, contrasena);
        ps.executeUpdate();

        JOptionPane.showMessageDialog(null, "datos insertados correctamente");
        conectaBase(bd).close();

    } catch (SQLException ex) {
        System.out.println(ex);
    }

}

//  Método que permite realizar una venta el caul recibe
cuatro parametros
public void insertarVenta(String bd, String table, String
modelo, int cantidad) {

    try {
        //Se prepara una sentencia a ejecutar
        PreparedStatement ps = conectaBase(bd).prepareStatement
        ("INSERT INTO " + table + "(modelo,cantidad)values (?,?)");
        ps.setString(1, modelo);
        ps.setInt(2, cantidad);
        ps.executeUpdate();
        conectaBase(bd).close();

    } catch (SQLException ex) {
        System.out.println(ex);
    }

}
//Metodo que permite modificar un producto desde la base
de datos este metodo recibe dos parametros
public int modificarCantidad(String bd, int cantidad,String
modelot)
{
```

```
    try {
      Statement s=conectaBase(bd).createStatement();
      // se modifica la cantidad del produto
      String uno="Update telefonos set cantidad= cantidad-"+
      cantidad+" where id_modelo='"+modelot+"';";
      s.executeUpdate(uno);
      System.out.println(uno);
       conectaBase(bd).close();
    } catch (SQLException ex) {
        System.out.println(ex);
    }
  return resu;
}


//Metodo que perimite modificar un telefo, este metodo recibe las caracteristicas especificadas e
// Los cuales nos permitirá hacer la modificacion del telefono.
public void modificarModelo(String bd, String table, String marca,
String ram, String memoriaint, String bateria, String pixeles, String pre_ven, String pre_com, St

    try {
    Statement s=conectaBase(bd).createStatement();
    Double prev = Double.parseDouble(pre_ven);
    Double prec = Double.parseDouble(pre_com);
    //Se procede ha ahcer la modificación del producto
    String uno="update "+table+" set marca='"+marca+"',ram="+Integer.parseInt(ram)+" ,memoriaint=
    parseInt(bateria)+", pixeles="+Integer.parseInt(pixeles)+",pre_ven="+prev+" ,pre_comp="+prec+
    where id_modelo='"+tab.getValueAt(tab.getSelectedRow(), 0)+"';";
    s.executeUpdate(uno);
    System.out.println(uno);
    JOptionPane.showMessageDialog(null,"datos modificados
    correctamente");
    conectaBase(bd).close();
    } catch (SQLException ex) {
        System.out.println(ex);
    }
}
//Metodo que nos peromite modificar las caracteristicas de la
cuenta de un usuario como nombre de usuario, contraseña,
correo
public void modificarUsuarios(String bd, String table,
JTable u,
String usuario, String contrasena, String correo) {
    String[] datos = new String[3];
    String[] campo = new String[3];
    campo[0] = "usuario";
    campo[1] = "contrasena";
    campo[2] = "correo";
    datos[0] = usuario;
```

```
        datos[1] = contrasena;
        datos[2] = correo;
        String mod;
        try {
            Statement s = conectaBase(bd).createStatement();
            for (int i = 0; i < 3; i++) {//Se reliza un recorrido para realizar los cambios respectiv
                mod = "Update " + table + " set " + campo[i] + " ='" + datos[i] + "'" + " where idU="
                s.executeUpdate(mod);

            }
                JOptionPane.showMessageDialog(null, "Los datos se han modificado exitosamente.");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage());
        }
    }
    //Metodo que permite validar que solpo se admintan letras mayúculas
    así como munúsculas
    public void MetodoValidarletras(KeyEvent evt) {

        char c = evt.getKeyChar();
        if ((c < 'a' || c > 'z') && (c < 'A' || c > 'Z')) {
            evt.consume();
        }

    }
//Metodo que perimite Validar que solo se ingrecen números
    public void MetodoValidarNumeros(KeyEvent evt) {

        char c = evt.getKeyChar();
        if ((c < '0' || c > '9')) {
            evt.consume();
        }

    }
    //Método que permite que solo se ingresen  números de tipo
    flotante
    public void MetodoValidarNumerosfloat(KeyEvent evt) {
      char c=evt.getKeyChar();
       if((c<'0'||c>'9')&& (c!='.'))
           evt.consume();
        }


   //Método que nos permite validar que solo se ingresen letras y
   número
    public void MetodoValidarletrasNumeros(KeyEvent evt) {
        char c = evt.getKeyChar();
        if ((c < 'a' || c > 'z') && (c < 'A' || c > 'Z')&&
        (c < '0' || c > '9'))  {
```

```
        evt.consume();
    }
}

public void MetodoValidarcorreo(KeyEvent evt) {

    char c = evt.getKeyChar();
    if ((c < 'a' || c > 'z')&&(c < '0' || c > '9')&&
    (c =='@'))  {
        evt.consume();
    }

}


//Metodo con retorno de tipo entero nos ayuda a verificar
que existan
la contraseña introducida
// si dicha contraseña es correcta se retorna un 1
public int contraseña(String contraseña, String bd) {//Validar
contraseña
    int a = 0;
    String SQL = "select * from USUARIOS where contrasena ='" +
    contraseña + "' ";
    try {
        Statement s = conectaBase(bd).createStatement();
        ResultSet rs = s.executeQuery(SQL);
        if (rs.next()) {
            a = 1;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error contraseña " +
        e);
    }
    return a;
}
//Metodo con retorno de tipo entero nos ayuda a verificar que
existan
la el nombre de usuario introducida
// si dicha usuario es correcta se retorna un 1
public int nombre(String usuario, String bd) {// validar usuario
    Connection c;
    int a = 0;
    String SQL = "select * from USUARIOS where usuario='" + usuario +
    "'";
    try {
        Statement s = conectaBase(bd).createStatement();
        ResultSet rs = s.executeQuery(SQL);
        if (rs.next()) {
```

```
            a = 1;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error nombre" + e);
    }
    return a;
}

public String usuario(String usuario, String bd) {// determinamos
si es administrador

    String SQL = "select * from USUARIOS where usuario='" + usuario +
    "' ";

    try {

        Statement s = conectaBase(bd).createStatement();
        ResultSet rs = s.executeQuery(SQL);
        if (rs.next()) {

            nom = rs.getString(2);// con getSring estraemos el
            contenido de la consulta

        }

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error usuario" + e);
    }

    return nom;
}

// Eliminamos un producto de la base de datos
public void eliminarModelo(String bd, String nombreTabla, String a) {
    try {
        Statement s = conectaBase(bd).createStatement();
        String eliminar = "DELETE FROM " + nombreTabla + " WHERE id_modelo='" + a + "';";
        s.executeUpdate(eliminar);
        JOptionPane.showMessageDialog(null, "Se eliminó el producto exitosamente.");

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, ex);
    }
}
//Eliminamos al Usuario de la base de datos.
public void eliminarUsuario(String bd, String nombreTabla, int id) {
    try {
        Statement s = conectaBase(bd).createStatement();
        String eliminar = "DELETE FROM " + nombreTabla + " WHERE idU="
```

```
            + id+ ";";
            s.executeUpdate(eliminar);
            JOptionPane.showMessageDialog(null, "Se eliminó el usuario  exitosamente.");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
    }

 //Metodo que permite establecer una fecha
public String productoFecha(){
       Date f=new Date();
       SimpleDateFormat ff=new SimpleDateFormat("dd/MM/YYYY");

    return ff.format(f);
    }
//Metodo que nos permite ingresar un producto más a la base de datos
 public void insertarVenta(String bd, String table, String modelo, int cantidad, String fecha) {

       try {

            PreparedStatement ps = conectaBase(bd).prepareStatement
            ("INSERT INTO " + table + "(modelo,cantidad,fecha)values (?,?,?)");
            ps.setString(1, modelo);
            ps.setInt(2, cantidad);
            ps.setString(3, fecha);
            ps.executeUpdate();
          //  JOptionPane.showMessageDialog(null, "Datos guardados correctamente.");
            conectaBase(bd).close();

        } catch (SQLException ex) {
            System.out.println(ex);
        }

    }

   public void mensajeActualizado(JLabel label, String mensaje){
       Timer miTimer= new Timer();
       label.setText(mensaje);
       TimerTask miMensaje = new TimerTask(){
           @Override
           public void run() {
              label.setText("");
           }

       };
       miTimer.schedule(miMensaje, 3000);
   }

   public Image Seticono() {
```

```
    Image icon=Toolkit.getDefaultToolkit().getImage("logo.png");
    return icon;
    }
}
```

Capítulo 6

Conclusión

El desarrollo del proyecto de punto de venta que inicio a principios del semestre hasta el momento actual se logró concretar lo especificado durante el inicio del proyecto. En esta última entrega del proyecto se destacan las siguientes funcionalidades las cuales son: altas de productos, bajas de productos, actualizaciones de productos, así como también bajas y altas de usuarios. Lo más importante de este programa es que realiza la venta de manera correcta ya que se validad cada uno de los campos para que de esta manera no se genera error al momento de realizar una venta de producto. Cabe mencionar que se implementó una sección que genera un archivo de Excel donde se almacena todos los registros de los productos vendidos. Con este proyecto de sistema de punto de venta se logró obtener más conocimiento a la hora de elaborar un programa, se implementó todo lo aprendido durante el semestre en el proyecto. Con este proyecto se da una idea de lo que realmente se puede realizar afuera en el mundo laboral

6.1. Referencias

[1] S.W. Ambler. The Elements of UML 2.0 Style. 1st Edición,
Edición Kindle.9 Mayo 2005

[2] S. Martina, S. Mariano, H. Christian, K. Gerti. UML @ Classroom.
Edición 2015. Springer. 9 marzo 2015

[3] O. Antoni. Conceptual Modeling of Information Systems.
2007th Edición. Springer. 28 Agosto 200

[4]R. Pressman, B. Maxim and L. Medellin Serna, Ingenieria de
l software , 7th ed. Mexico [etc.]: McGraw-Hill, 2010, p. 20.

[5]I. SOMMERVILLE, Ingeniería del software , 7th ed.
Madrid(españa): Miguel Martín romo, 2005, p. 15.

[6]J. Pilataxi, "UML Gota una Gota de Martin Fowler
FEBRERO", Academia.edu de 1999. [En línea]. Disponible:https://www.academia.edu/9309108/
UML_Gota_a_Gota_Martin_Fowler_FEB. [Consultado El: 02 - Feb - 2022].

[7] Bruegge B, Dutoit A. Ingenieria de software orientado

a objetos.Mexico: Pearson Educacion; 2002.

[8] Booch G, Rumbaugh J, Jacobson I, Garcia Molina J,
Saez Martin ezJ. El lenguaje unificado de modelado. Madrid: Pearson Educacion; 2010.

[9]Pantaleo G, Rinaudo L, German Asenjo A.Ingenieria de software
. Mexico: Alfaomega; 2015.

[10]P. Rigby y M. Norris, Ingenieria de Software Explicada /
Software Engineering Explained. Limusa, 1994.

[11]McKinney, D. and Jimenez, A., n.d. Un dia de trabajo d
e desarrollador de software.

[12]T. Erler, UML. Bonn: Bhv, 2002.

6.2. Anexo 1.PSP

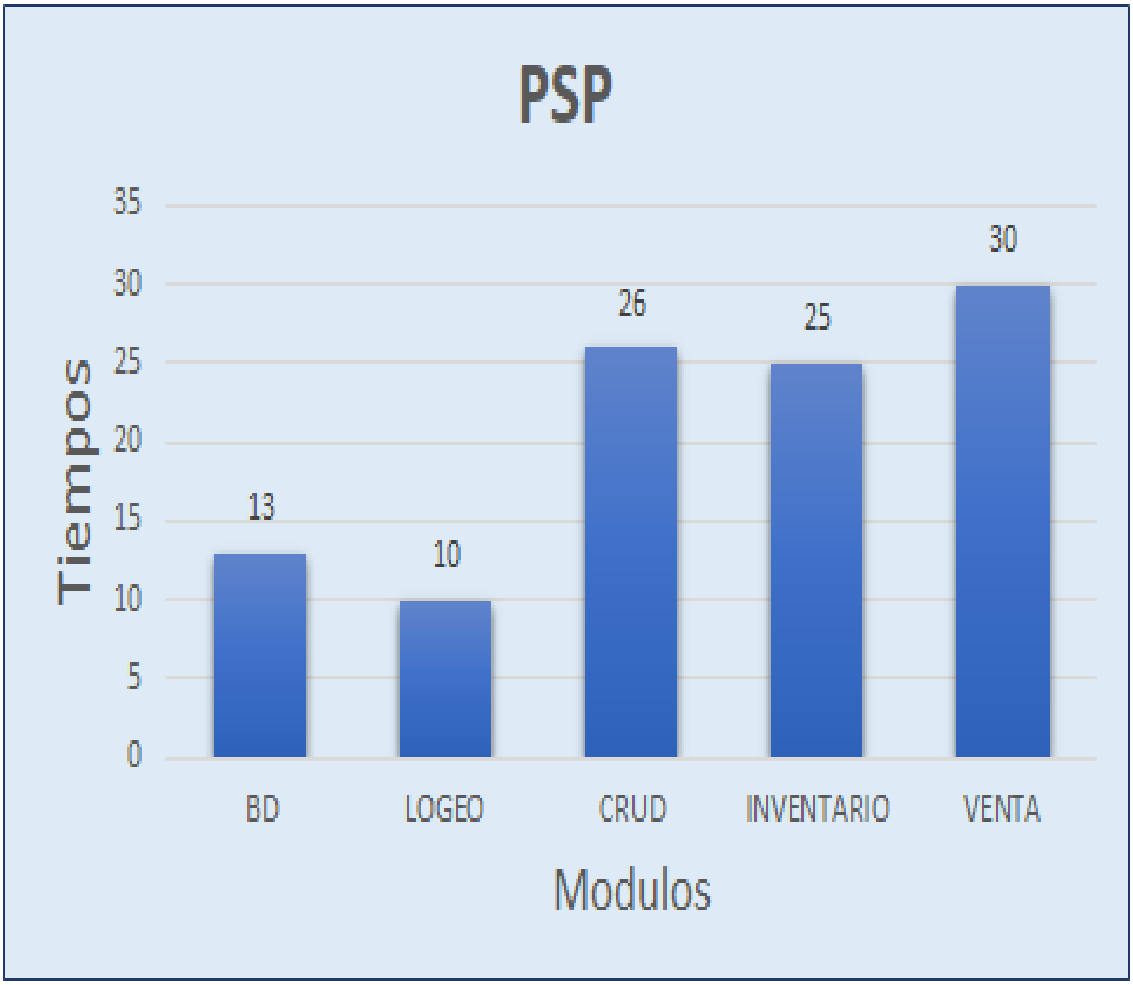


Figura 6.1: Gráfica del usu de PSP

En la gráfica se puede apreciar a ver que la parte de ventas nos llevó mucho tiempo aplicarlo.

6.3. Anexo 2.Galería de imagenes

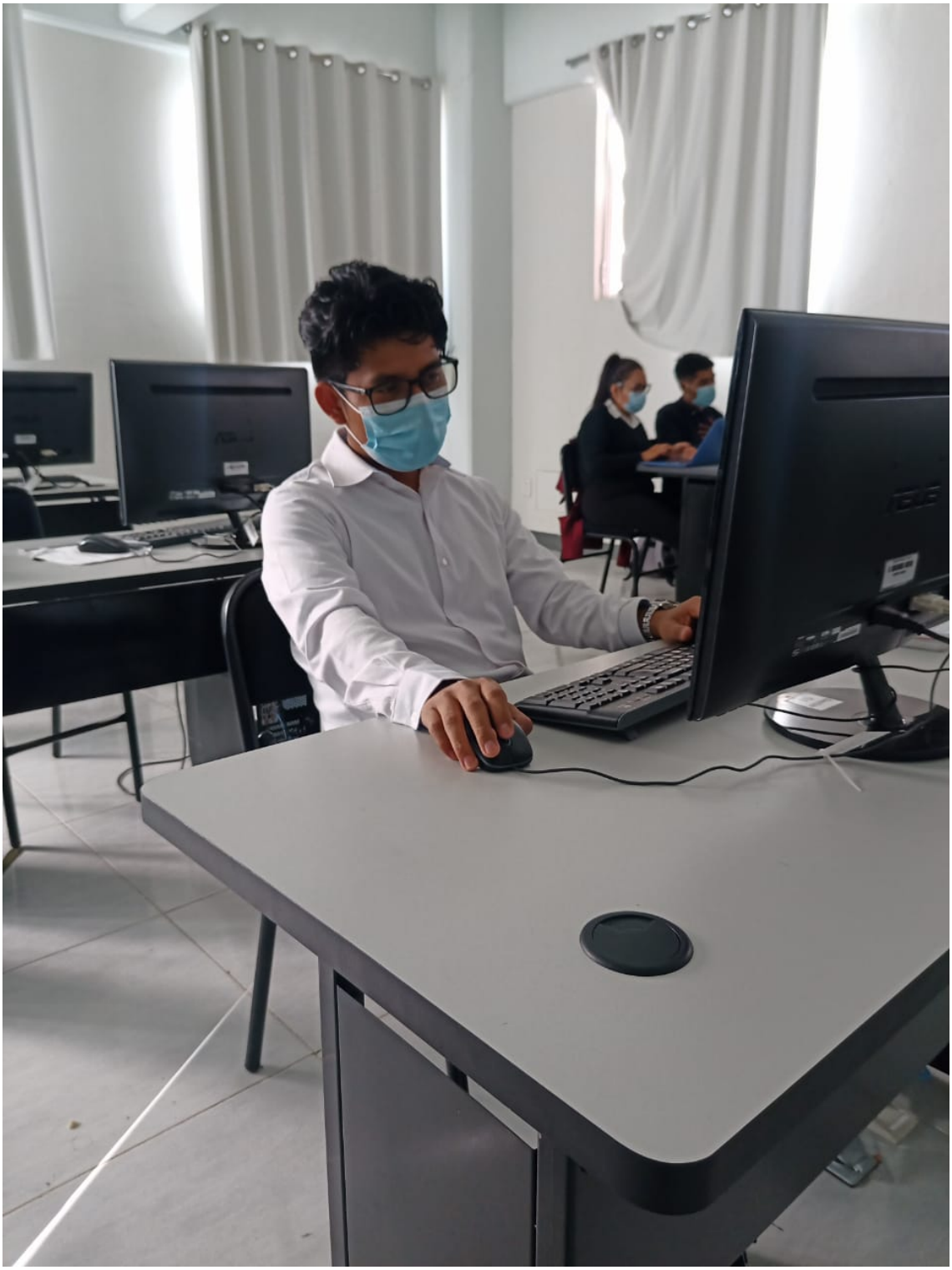


Figura 6.2: Evidencia la aportación de Hermilo

Figura 6.3: Evidencia la aportación de Hermilo

Figura 6.4: Evidencia la aportación de Floriceli

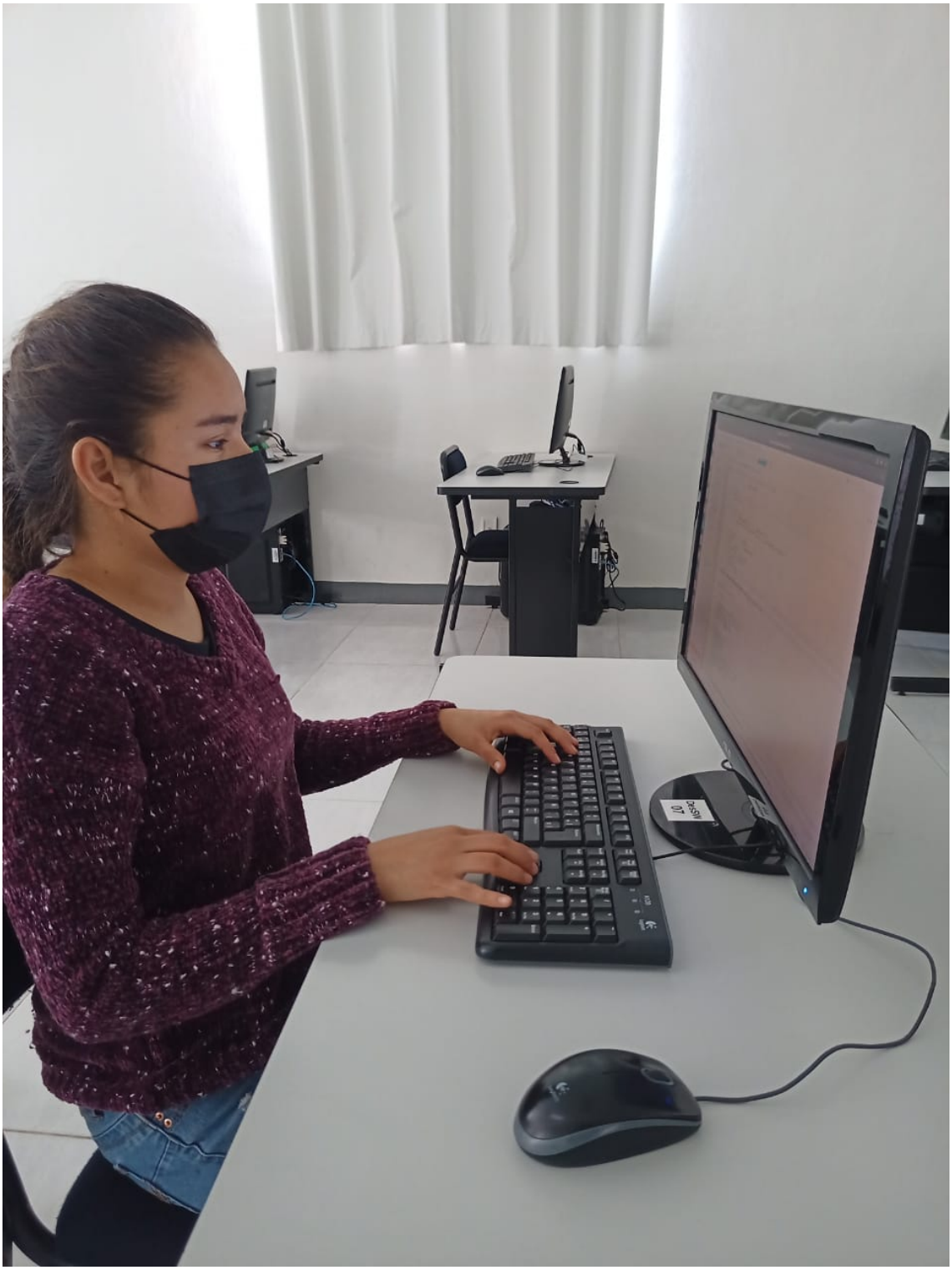


Figura 6.5: Evidencia la aportación de Floriceli

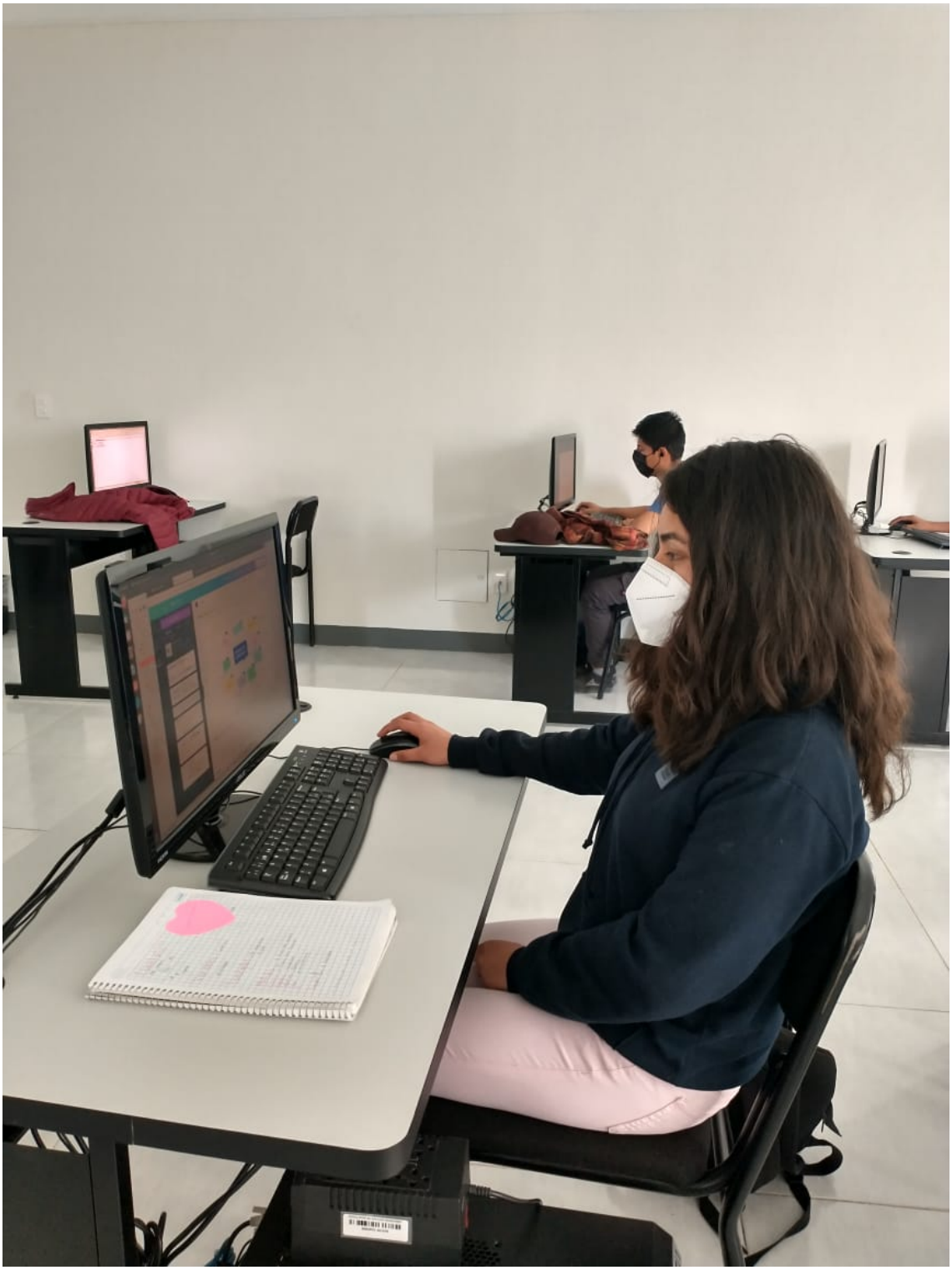


Figura 6.6: Evidencia la aportación de Stephanie

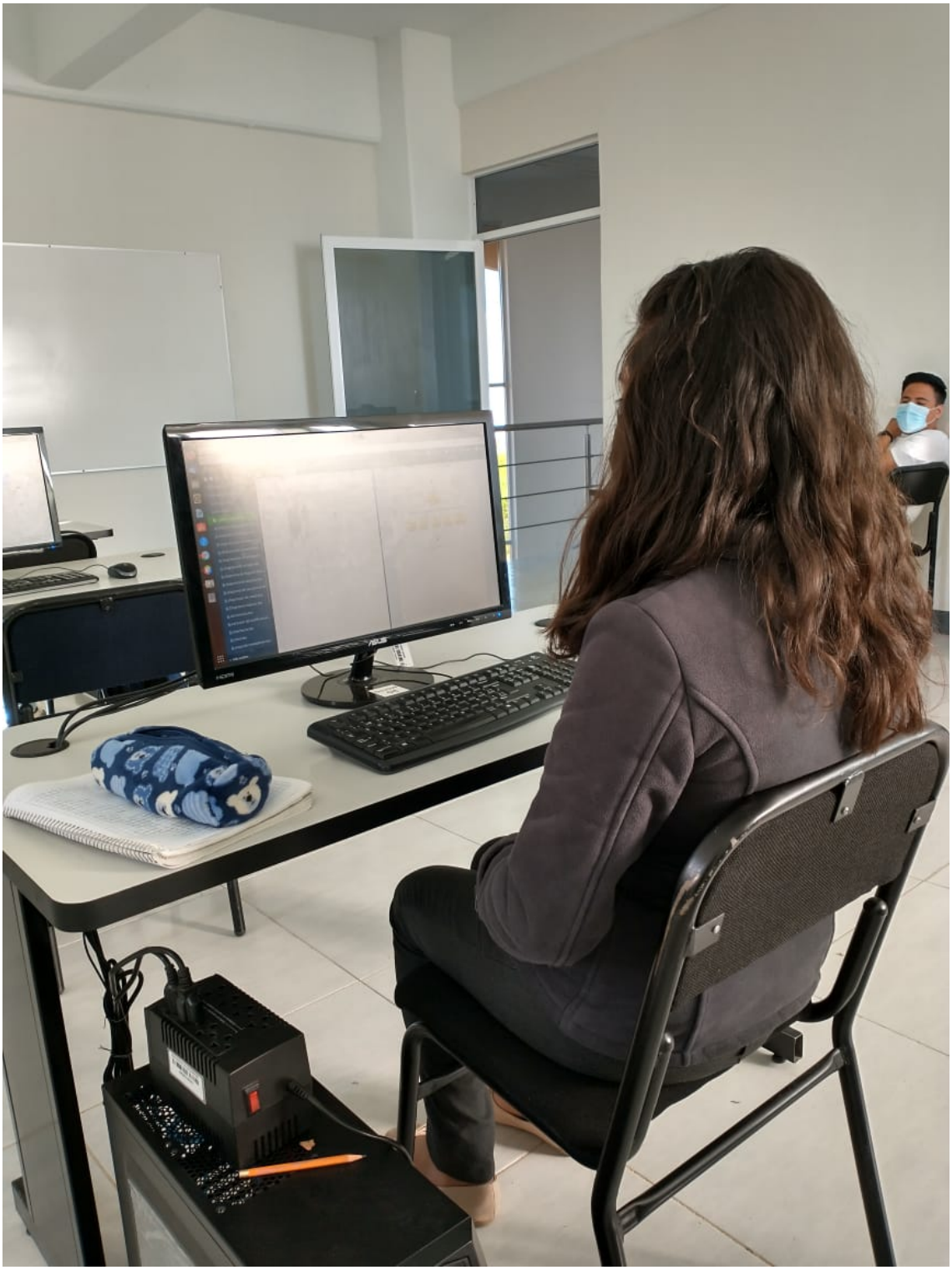


Figura 6.7: Evidencia la aportación de Stephanie

Figura 6.8: Evidencia de la colaboración del equipo