



CONCESIONARIO DE AUTOS

ASIGNATURA

PROGRAMACIÓN ORIENTADA A OBJETOS

INTEGRANTES

SANTIAGO FLOREZ AGUDELO

JONATAN ESTIVEN SANCHEZ VARGAS

JUAN JOSE ALZATE ROJAS

FELIPE VELEZ FERNANDEZ

2023

UNIVERSIDAD NACIONAL DE COLOMBIA

Características de POO

Clases abstractas

```
package gestorAplicacion.activos;

import java.io.Serializable;

public abstract class Transaccion implements Serializable{
    private static final long serialVersionUID = 1L;
    static ArrayList<Transaccion> transacciones = new ArrayList<Transaccion>();
    String tipo;
    int transfer;
    Mecanico mecanico;
    long idTransaccion;
}
```

Transacción es una clase abstracta, se definió como abstracta ya que en el programa tenemos diferentes tipos de transacciones, por lo que fue más eficiente tener una “plantilla” de transacciones.

Métodos abstractos

```
public abstract String info();
```

el método info() de la clase Transaccion lo utilizamos como una especie de toString, el cual invocamos en el momento en que una transacción independiente de su tipo, es creada al momento de usar una funcionalidad (ejemplo, TransaccionVenta en la funcionalidad venta autos)

Constante

```
public class Vendedor extends Trabajador implements Serializable{
    private static final long serialVersionUID = 1L;
    static ArrayList<Vendedor> vendedores = new ArrayList<Vendedor>();
    int ventas = 0;
    protected String puesto;
    final static double comision = 0.02;
}
```






Esta constante es el valor de la comisión que se pone en el concesionario sobre la venta de autos y artículos.

Interfaces

```
public interface Persona {  
    String getNombre();  
    long getCedula();  
    long getTelefono();  
    String getCorreo();  
    String getDireccion();  
    void setNombre(String nombre);  
    void setCedula(long cedula);  
    void setTelefono(long telefono);  
    void setCorreo(String correo);  
    void setDireccion(String direccion);  
}
```

Decidimos crear la interfaz Persona ya que antes la habíamos definido como una clase abstracta, pero figuraba mejor como interfaz ya que realmente solo tenía gets y sets como métodos, nada complejo que se herede.

Herencia

- >  Transaccion.java
- >  TransaccionModificacion.java
- >  TransaccionTaller.java
- >  TransaccionVenta.java
- >  TransaccionVentaTaller.java

```
public class TransaccionVenta extends Transaccion {  
    // Atributos de Transacción  
    // Atributos de Transacción Venta  
    // Métodos de Transacción  
    // Métodos de Transacción Venta  
}
```

Todos los tipos de transacciones heredan de la clase abstracta Transacción, cada tipo de transacción tiene atributos específicos adicionales, por lo que fue mejor organizar las transacciones de esta manera.

Ligadura dinámica

```
public static ArrayList<Vendedor> selectorVend(Auto o) {  
    String vende=null;  
    ArrayList<Vendedor> vendedores = new ArrayList<Vendedor>();  
    vende="Vitrina";  
    for (Vendedor vendedor : getVendedores()) {  
        if (vende.equals(vendedor.getPuesto())) {  
            vendedores.add(vendedor);  
        }  
    }  
    return vendedores;  
}
```

Atributos de clase

```
static ArrayList<TransaccionVenta> transaccionesven = new ArrayList<TransaccionVenta>();
```


El tipo de atributo de clase que más usamos en todo el programa fueron los arraylist de clase, ya que con estos podemos tener de manera ordenada todos los objetos dentro de cada arraylist correspondiente a cada clase.

métodos de clase

```
public static ArrayList<Auto> getAutosDisponibles() {  
    ArrayList<Auto> disponibles = new ArrayList<Auto>();  
    for (Auto auto : autos) {  
        if (auto.isDisponible()) {  
            disponibles.add(auto);  
        }  
    }  
    return disponibles;  
}
```

Algunas funcionalidades requerían el uso sin llamar a una instancia del objeto, o bien, también porque se utilizaban atributos de clase.

Uso de constante

```
public class Vendedor extends Trabajador implements Serializable{  
    private static final long serialVersionUID = 1L;  
    static ArrayList<Vendedor> vendedores = new ArrayList<Vendedor>();  
    int ventas = 0;  
    String puesto;  
    final static double comision = 0.02; 
```

```
@Override  
int calcularSalario() {  
    return (int) (getSalario() + (ventas * comision));  
}
```

Encapsulamiento

Sobrecarga de métodos

```
public static void pago(Vendedor vend, Auto a) {  
    vend.setSalario(vend.getSalario() + ((a.getPrecio()) * Vendedor.comision));  
}  
public static void pago(Vendedor vend, Artículo a) {  
    vend.setSalario(vend.getSalario() + ((a.getPrecio()) * Vendedor.comision));  
}
```

El método pago se usa en diferentes funcionalidades, pero no con los mismos objetos, por lo que usamos sobrecarga de métodos para no redundar en el uso del código.

Desambiguación

```
@Override
public void setCedula(long cedula) {
    this.cedula=cedula;
}

@Override
public void setTelefono(long telefono) {
    this.telefono=telefono;
}
```

```
public Trabajador(String nombre, long cedula, long telefono, String correo, double salario, String banco,
    long cuentaBanco) {
    this(nombre,cedula,telefono,correo,"Medellin",salario,banco,cuentaBanco);
}
```

```
public Cliente(String nombre, long cedula, long telefono, String correo, String modeloInteres, long presupuesto) {
    this(nombre,cedula,telefono,correo,"Medellin",modeloInteres,presupuesto);
}
```

Enumeración

```
public enum MarcaAuto {
    TOYOTA,
    CHEVROLET,
    MAZDA
}
```

Se utilizó para darle la marca a los carros.

EXPLICACIÓN TÉCNICA.

- **PROCESO VENTAS**

```
Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas de ventas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
Ingrese el número de la opción que va a utilizar:
```

Esta funcionalidad inicia pidiéndole al cliente su cédula, este número se inserta en un método que busca que este número coincida con el de algún cliente registrado en el concesionario, para luego traer el objeto cliente que encuentre relacionado a este número de identificación. El método usado es:

- ***Cliente.getClientPorCedula(cedula)***

```
Ingrese el número de la opción que va a utilizar: 1
Escriba la cédula del comprador:
1234567890
```

Luego, con el objeto cliente encontrado, se muestra en pantalla el nombre del cliente, el presupuesto que ha registrado en el concesionario y por último se

muestra una lista de los carros de la marca de interés del cliente, esto se hace primero utilizando el método ***info()*** del objeto cliente que vamos a llamar en esta funcionalidad ***comprador***, se utiliza este método que hace un return de su nombre, y se utiliza el método ***comprador.getPresupuesto()*** que devuelve el presupuesto del comprador, luego para mostrar la lista se toman solo los autos que corresponden a la marca de interés del comprador, comprobando a través del array que devuelve el método ***InventarioAuto.getAutosDisponibles()*** El cual devuelve todos los autos disponibles, luego, se le pide al comprador que elija una opción, alguno de los carros listados, o más opciones de búsqueda.

```

Nombre del cliente: Mafalda González
Su presupuesto es: 35000000
Estos son los autos de la marca de interés para el cliente disponibles en este momento:
Los carros de la marca Mazda disponibles son:

    Modelo          Precio          Color
1  Cx-3             500000000      azul
2  Cx-5             120000000      verde
3  Cx-7             240000000      rosa
4  Cx-9             340000000      azul
5  Miata            440000000      rosa

Seleccione el numero del carro[1-5]
0. Más opciones de búsqueda...
Seleccione el Auto en el que está interesado o use las otras opciones de búsqueda:

```

Si se selecciona alguno de los carros, el objeto queda asignado al nombre de auto, el cual en pantalla se mostrará su información completa y se le pide al usuario que confirme su elección, luego se crea un array de tipo vendedor llamado vendedores que se le asigna el valor a través del método **Vendedor.selectorVend(auto)** el cual recorre un array que se obtiene por el método **getVendedores** que retorna un array de todos los vendedores, entonces con este array se compara el atributo Puesto de cada vendedor y los que sean del puesto Vitrina se añaden en el array que devuelve el método **Vendedor.selectorVend(auto)** y los vendedores dentro de este array se ordenan como se muestra en pantalla.

```

El auto elegido es Datos del Carro: Modelo Cx-3, Marca Mazda, Precio 500000000, Color azul, Full equipo
¿Desea confirmar?: (si/no)
si
Los vendedores de Vitrina disponibles son:

    Vendedor          Tipo de venta
1  Juan Guaido        Vitrina
2  María De la Paz    Vitrina
3  Ana Gabriel        Vitrina
4  Carla Jimenez      Vitrina

Seleccione el numero del vendedor[1-4]:
1

```

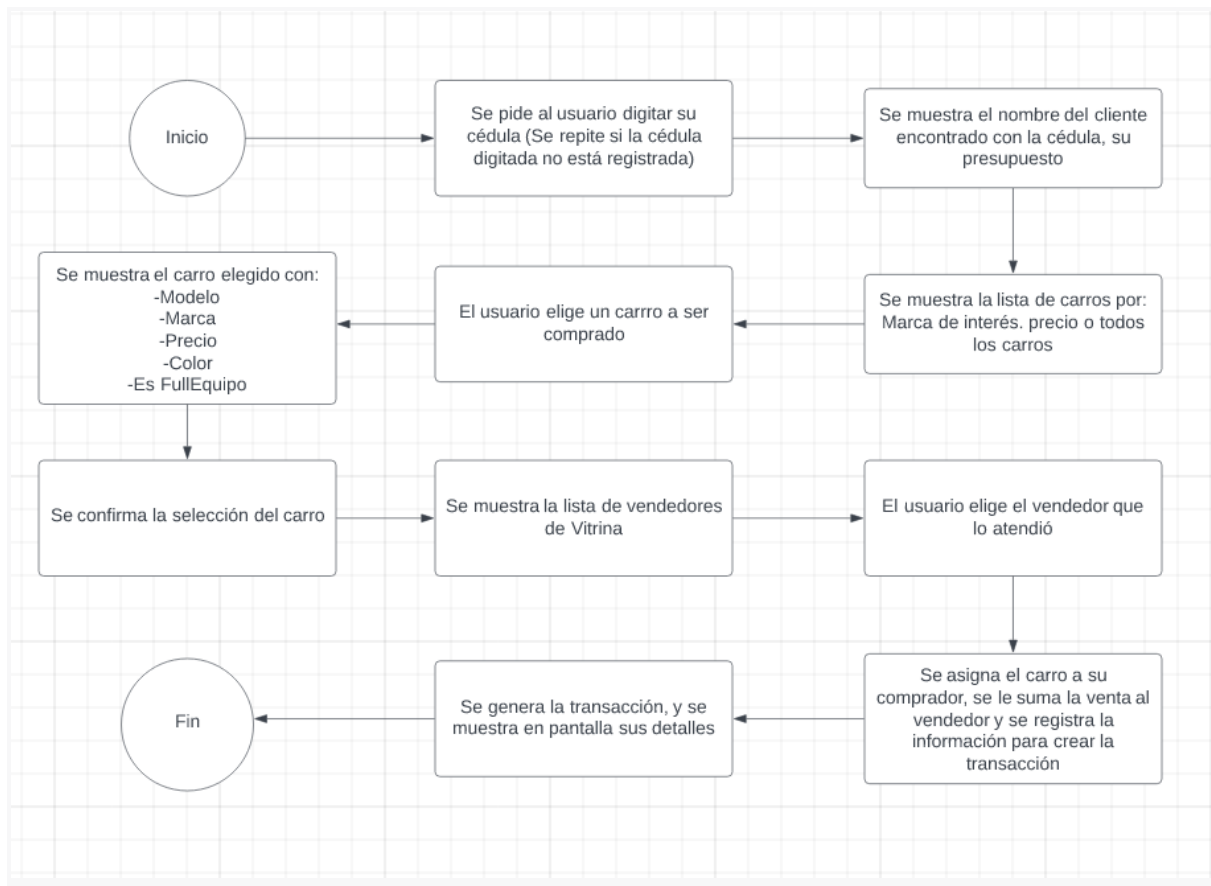
Se le pide al usuario que seleccione el vendedor que le ha atendido para que finalmente se cree la compra, el sistema crea un nuevo objeto de tipo

TransaccionVenta en el cual sus atributos son el método de pago, el comprador que es tipo Cliente, el auto que ha elegido y el vendedor asociado, todo esto se muestra en pantalla para dar a entender que la compra se ha completado.

```

Transacción #00000659: venta realizada por Juan Guaido para el cliente María Conchita Alonso por un total de $50000000 por el auto Cx-3

```



- **PROCESO TALLER**

```

Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas de ventas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
Ingresa el número de la opción que va a utilizar:
  
```

La funcionalidad taller, es una funcionalidad que interactúa con diferentes clases y métodos especificados en ellas.

Esta funcionalidad empieza pidiéndole al usuario su identificación, este número de identificación ingresa a dos métodos en los cuales se comprueba que la identificación, tenga una transacción de compra de vehículo (Ya que nuestro concesionario solo le presta servicios de taller a vehículos comprados allí); y que también el usuario tenga un auto en su poder, esto se hace mediante los métodos:

- ***TransaccionVenta.getClientPorCedula(cedula);***
- ***TransaccionVenta.getTransaccionPorCedula(cedula);***

Estos métodos trabajan buscando si en el Array de clientes y en el Array de Transacciones está este comprador, y estos nos van a devolver el primero un objeto tipo cliente y el segundo un objeto tipo Auto y un mensaje con la información del cliente y se le pregunta si desea confirmar. Si alguno de los métodos nos da un resultado de **null**, Sacará un mensaje que el usuario no posee transacciones en nuestro concesionario y se le pedirá que vuelva a ingresar la cédula

```
Introduzca la cédula del propietario: 1234991492
Nombre del cliente: Felipe Velez
¿Confirmar propietario? (si/no)si
```

```
Introduzca la cédula del propietario: 123456789
La cédula ingresada no se encuentra en Transaccion. Por favor, vuelva a ingresarla.
Introduzca la cédula del propietario:
```

(*Img 3*)

Luego de esto, teniendo ya un cliente registrado y con un auto se le preguntara que le desea hacer a su vehículo, y mediante la opción que se escoja, se va a usar el método:

- ***Mecanico.mecanicoDisponible(auto)***

Este método recibe el auto que arrojó el método anterior y va a mostrarnos los mecánicos disponibles para nuestro vehículo con ese tipo de especialidad (*Img 5*), en caso de no tener mecánicos disponibles, sacará un mensaje diciendo que no hay mecánicos disponibles para nuestro vehículo.

```
System.out.println("\n\nQue deseeas hacerle al Vehiculo");
System.out.println("1. Latoneria y pintura");
System.out.println("2. Cambio de Llantas y alineacion");
System.out.println("3. Cambio de Aceite");
System.out.println("4. Cambio de Frenos");
System.out.print("Ingrese el número de la opción que va a utilizar: ");
//Se ingresa el auto que se devolvio anteriormente y este entrega una lista de mecanicos
ArrayList<Mecanico> mecanicos=Mecanico.mecanicoDisponible(auto);
```

```
Que deseeas hacerle al Vehiculo
1. Latoneria y pintura
2. Cambio de Llantas y alineacion
3. Cambio de Aceite
4. Cambio de Frenos
Ingrese el número de la opción que va a utilizar: 2
Los mecanicos que atienden Toyota disponibles son:

Nombre          Cedula          Atiende    Especialidad
1 Daniela Ramirez 1234567893      Toyota     Llantas

Selecione el numero del mecanico[1-1]:
```

Este método nos permite seleccionar un mecánico y **este nos arroja un objeto tipo mecánico** usando el objeto auto arrojado anteriormente.

Luego, con el mecánico arrojado anteriormente se hace un llamado al método:

- ***InventarioArticulo.articuloDispo(mecanico)***

```
//Ingresa este mecanico que se selecciono y se devuelve una lista de productos  
ArrayList<Articulo> producto=InventarioArticulo.articuloDispo(mecanico);
```

Este método nos permite seleccionar el producto el cual quiero ponerle a mi vehículo, mostrándonos los artículos específicamente para la especialidad que escogí.

Los productos Llantas disponibles son:

Referencia	Producto	Tipo Vehiculo	Marca	Precio	Cantidad
1 1004	Llanta todo terreno	automovil y camioneta	GoodYear	745000	100
2 1005	Llanta todo terreno	automovil y camioneta	Michelin	1150000	99
3 1006	Llanta terreno de barro	automovil y camioneta	Bridgestone	750000	100
4 1007	Llanta terreno de barro	automovil y camioneta	Yokohama Geolanda	1899000	100
5 1008	Llanta terreno de asfalto	automovil y camioneta	Goodyear	650000	100
6 1009	Llanta terreno de asfalto	automovil y camioneta	Michelin	1000000	100

Selecione el numero del producto[1-6]:

Este método nos permite seleccionar un Producto y este nos arroja un objeto tipo Artículo usando el objeto Mecánico obtenido anteriormente.

Con el artículo seleccionado se procede a dar una información que reúne todo lo seleccionado para que el usuario pueda confirmar si desea hacerlo

```
?Confirmar producto? (si/no)si  
El procedimiento a realizar es: Cambio de Llantas con Llanta terreno de barro, y su mecanico será Daniela Ramirez  
?Confirmar Transaccion? (si/no)
```

Después de haber confirmado los productos, el mecánico y tener el auto, se le pide al usuario que seleccione la hora en la cual desea ser atendido, le muestra la agenda del mecánico que seleccionó.

```
Selecione la hora para su servicio  
Esta es la agenda de: Daniela Ramirez  
Las horas disponibles son:  
  
Horario disponible  
1 9:00-11:00  
2 11:00-1:00  
3 2:00-4:00  
4 4:00-6:00  
  
Selecione la hora[1-4]:
```

Al seleccionar la hora, se da por terminada la transacción y se crea un objeto nuevo, recogiendo los objetos dados anteriormente, tales como Auto, Mecanico, Producto, Ingreso, acá en esta fase del código también se da el precio final que es el precio del producto más el precio mano de obra del mecánico, igualmente se le reducen las unidades usadas al Artículo, se le modifica el artículo al auto **(Si aplica)(Llantas, Pintura)**, Y luego de esto se hace un llamado al método Trabajador.pago para modificarle el salario al mecánico, sumándole el pago por la mano de obra.

```
Trabajador.pago(mecanico);
```

```
public static void pago(Mecanico mec) {  
    mec.setSalario(mec.getManoObra()+mec.getSalario());  
}
```

Se le confirma la transacción, precio, mecánico y hora al cliente y se le informa que la factura se le enviará a su correo **(Img 11)**.

```
La hora de su cita es4:00-6:00  
El precio total por su Servicio es:1949000  
Transacción #00000661: taller realizado por Daniela Ramirez para el cliente Felipe Velez por un total de $1949000  
La Factura le llegara a el correo: fevelezf@unal.edu.co  
¿Desea volver al menú principal? (si/no):
```

(Img 11)

Al darle que si desea volver al menú principal, muestra nuevamente el menú principal, en caso de decir que no, se despide diciendo hasta luego **(Img 12)**, y se serializan nuevamente todos los objetos, más los creados en el tiempo de ejecución **(Img 13)**.

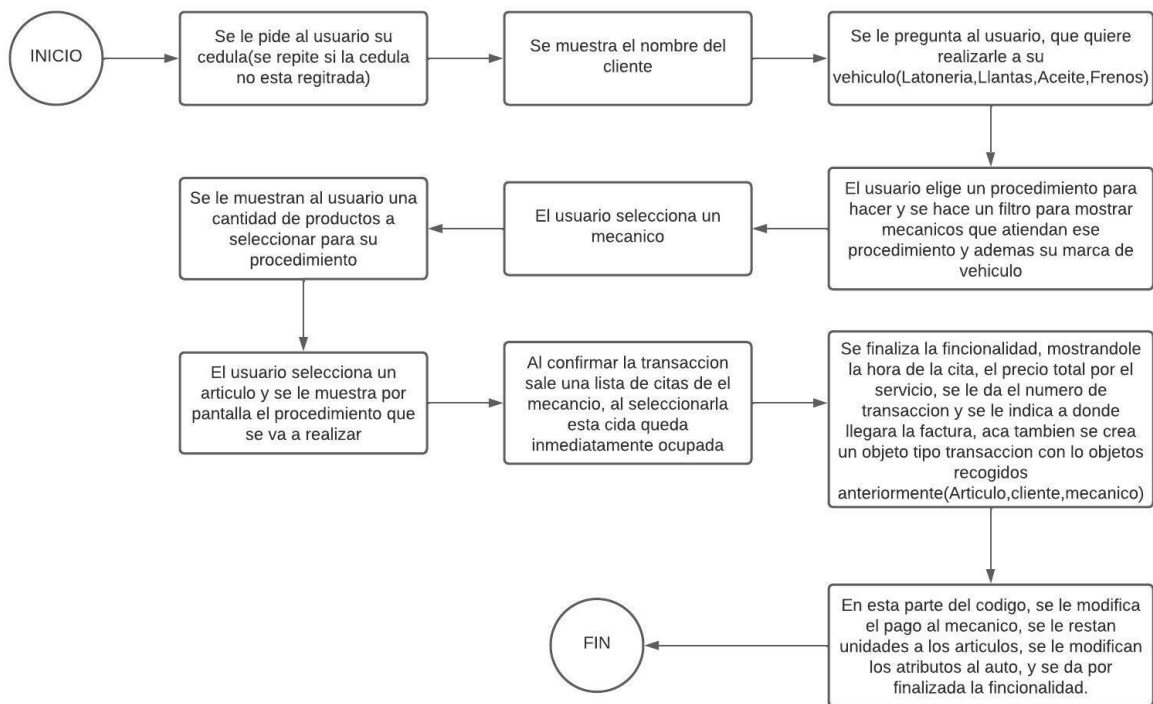
```
¿Desea volver al menú principal? (si/no): si  
  
Menú principal Concesionario  
1. Venta de Autos  
2. Venta de Repuestos  
3. Taller  
4. Consultar estadísticas de ventas  
5. Personalizar su auto  
6. Crear nuevo usuario (Comprador)  
7. Administración  
8. Salir  
Ingrese el número de la opción que va a utilizar:
```

```

case 3:
    procesoTaller();
    System.out.print("\n¿Desea volver al menú principal? (si/no): ");
    respuesta = sc.next();
    if(respuesta.equals("no")){
        Serializador.serializeArrays();
        volverAlMenuPrincipal = false;
    }
    break;

```

(Img 13)



• VENTA DE REPUESTOS

```

Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas / finanzas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
Ingrese el número de la opción que va a utilizar:

```

La funcionalidad Venta de repuestos, es una funcionalidad que interactúa con diferentes clases y métodos especificados en ellas.

Esta funcionalidad empieza pidiéndole al usuario su identificación, este número de identificación se ingresa a un método llamado:

- **Cliente.getClientPorCedula(cedula)**

```
//Devuelve un cliente que va debe estar previamente registrado  
comprador = Cliente.getClientPorCedula(cedula);
```

Este método trabaja buscando si en el Array de clientes está este comprador, y este nos va a devolver un objeto tipo cliente y un mensaje con la información del cliente y se le pregunta si desea confirmar. Si el método nos da un resultado de **null**, Sacará un mensaje que el usuario no está registrado en nuestro concesionario y se le pedirá que vuelva a ingresar la cédula.

```
Escriba la cédula del comprador:  
1234991492  
Nombre del cliente: Felipe Velez  
¿Confirmar cliente? (si/no)
```

Luego de esto se pregunta qué tipo de repuesto quiere comprar y mediante la opción que escoja se ingresa al método:

- **InventarioArticulo.selectorEspecial();**

```
¿Que repuesto desees Comprar?  
1. Repuestos de Motor  
2. Escapes  
3. Sistema de Sonido  
4. Suspension  
Ingrese el número de la opción que va a utilizar: 1  
Usted Va a comprar un repuesto de: Motor  
¿Confirmar Tipo de repuesto? (si/no)
```

```
repuesto=InventarioArticulo.selectorEspecial();
```

Este método hace un filtrado según la opción que escogí para los diferentes tipos de repuestos que tenemos, **Este método nos devuelve un objeto tipo Articulo.**

Con el articulo que me devuelve este método y despues de haber confirmado el tipo de repuesto, ingresamos el articulo al método:

- **InventarioArticulo.selectorMarca(repuesto)**

```
//Recibe la lista anterior y filtra por marcas de autos y devuelve un array de articulos filtrados  
marca=InventarioArticulo.selectorMarca(repuesto);
```

Según la marca que escoja el usuario, el hace un respectivo filtro de los artículos compatibles con dicha marca y **nos arroja un objeto tipo Artículo**.

```
¿Que Marca de auto posee?
1. Toyota
2. Mazda
3. Chevrolet
Ingrese el número de la opción que va a utilizar: 1
Usted Va a comprar un repuesto de: Toyota
¿Confirmar Tipo de Vehiculo? (si/no)
```

Con el Artículo del método anteriormente mencionado y después de haber confirmado el tipo de vehículo, se hace un llamado al metodo:

- `InventarioArticulo.selectorCalidad(marca)`

```
System.out.println("Ingrese el número de la opción que va a utilizar: ");
//Con la lista anterior se hace un refiltrado, por articulos premium o basico y devuelva una lista de articulos
ArrayList<Articulo> articulos=InventarioArticulo.selectorCalidad(marca);
```

Según la calidad que escoja el usuario, el hace un respectivo filtro de los artículos de dicha calidad y **nos arroja un objeto tipo Artículo**.

```
¿Que calidad desea?
1. Premium
2. Basico
Ingrese el número de la opción que va a utilizar:
```

Una vez seleccionado , nos permite escoger el tipo de artículo que queremos, mediante un selector:

```
//Selector de articulo despues de los filtros y devuelve un articulo.
String resultp = String.format("%-40s%-25s%-20s%-15s\n", " Producto", " Tipo Vehiculo", " Marca", " Precio");
byte i=0;
for (Articulo articuloi:articulos) {
    i++;
    String articuloinfo = String.format("%-40s%-25s%-20s%-15s\n", articuloi.getTipoArticulo(), articuloi.getTipoVehiculo(), articuloi.getMarca(), articuloi.getPrecio());
    resultp += String.format("%-3ds\n", i, articuloinfo );
}if (articulos.size() >= 1) {
    System.out.println("Los productos " + articulos.get(0).getEspecialidad() + " disponibles son:\n");
    System.out.println(resultp);
    int num = 0;
    while (num <= 0 || num > articulos.size()) {
        System.out.println("Seleccione el numero del producto" + "[1-" + articulos.size() + "]: ");
        if (sc.hasNextInt()) {
            num = sc.nextInt();
        } else {
            System.out.println("Entrada invalida. Introduzca un numero entre 1 y " + articulos.size() + ".");
            sc.nextLine(); // limpiar la entrada no válida
        }
    }
    articulo = articulos.get(num - 1);
}
```

Y esto es lo que nos muestra por pantalla

```
Los productos Motor disponibles son:
```

	Producto	Tipo Vehiculo	Marca	Precio
1	Bomba de agua	automóvil	Airtex	350000

```
Seleccione el numero del producto[1-1]:
```

```
1
```

```
Bomba de agua Airtex
```

```
¿Confirmar repuesto? (si/no)
```

Una vez confirmado el producto mediante el selector que nos arroja un **objeto tipo articulo** , y una vez confirmado el repuesto, procedemos a escoger el vendedor el cual nos asesoró en la compra mediante el método:

```
ArrayList<Vendedor> vendedores= Vendedor.selectorVend(articulo);
```

Este método toma el Articulo anteriormente arrojado y nos arroja un objeto tipo Vendedor, donde se hace la escogencia del vendedor.

```
Los vendedores de Repuestos disponibles son:
```

	Vendedor	Tipo de venta
1	Pedro Mojica	Repuestos
2	Luis Suarez	Repuestos
3	Jorge Elias	Repuestos

```
Seleccione el numero del vendedor[1-3]:
```

```
1
```

```
¿Confirmar vendedor? (si/no)si
```

```
¿Confirmar Transaccion? (si/no)si
```

```
Transacción #00000521: venta realizada por Pedro Mojica para el cliente Felipe Velez por un total de $350000 por el articulo Airtex
```

```
La Factura le llegara a el correo: felipe@gmail.com
```

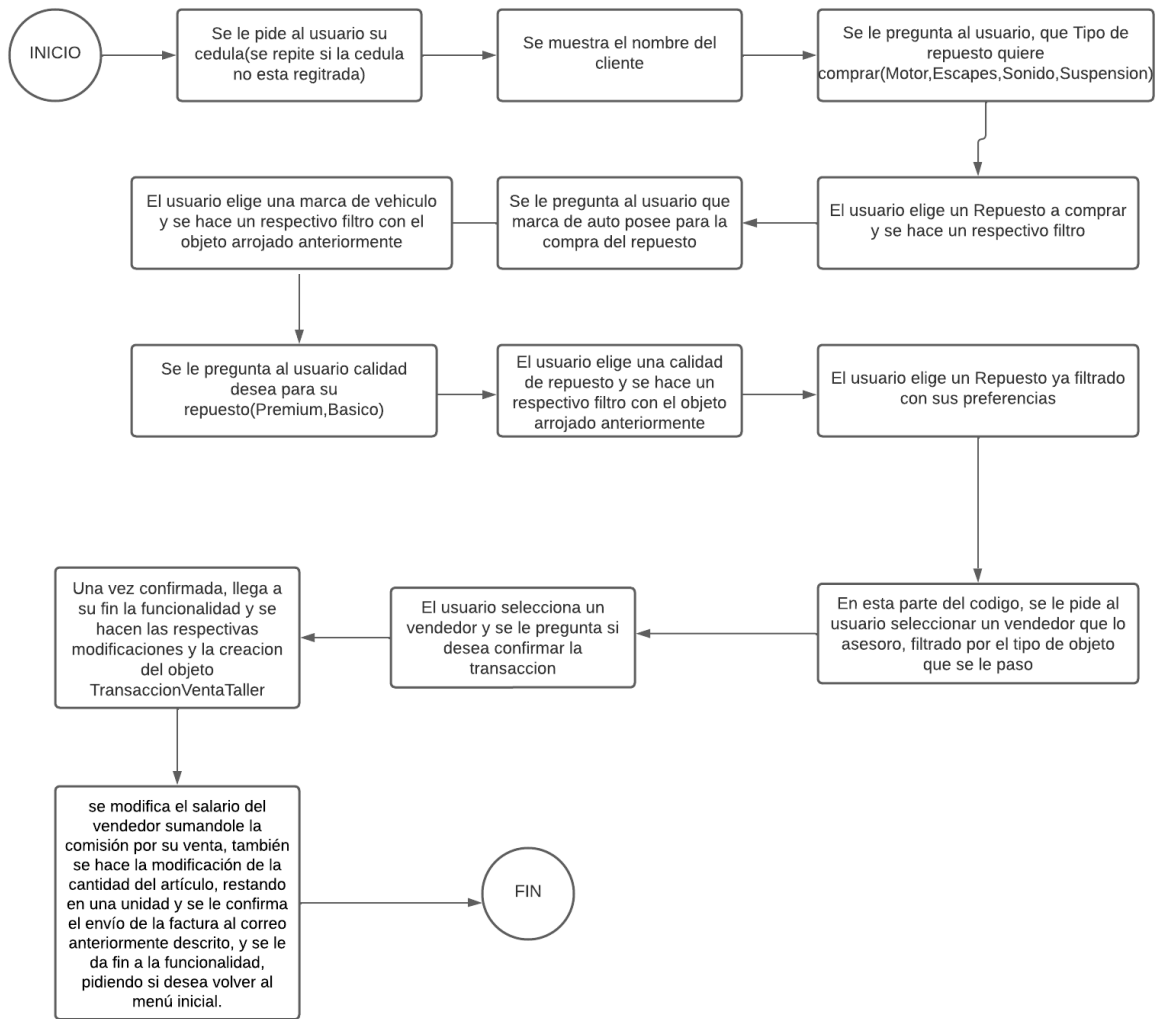
```
¿Desea volver al menú principal? (si/no):
```

Una vez escogido el vendedor se pide la confirmación tanto del vendedor como de transacción y posteriormente se crea el objeto tipo TransaccionVentaTaller, tomando los objetos anteriormente arrojados, se le hace un llamado al método:

- Trabajador.pago(vendedor,articulo);

donde esta modifica el salario del vendedor sumandole la comisión por su venta, también se hace la modificación de la cantidad del artículo, restando en una unidad y se le confirma el envío de la factura al correo anteriormente descrito, y se le da fin a la funcionalidad, pidiendo si desea volver al menú inicial.

```
if(!confirmarVendedor("no")) {  
    vendedor.confirmarVenta();  
    articulo.cantidad--;  
    Trabajador.pago(vendedor,articulo);  
    int transfer = (int) (Math.random() * 1000);  
    System.out.println(new TransaccionVentaTaller("efectivo", articulo.getPrecio(), comprador, articulo, vendedor,transfer).info());  
    System.out.print("La Factura le llegara a el correo: "+ comprador.getCorreo());  
    System.out.print("");  
}
```



- **CONSULTAR ESTADÍSTICAS / FINANZAS**

NOTA: Para acceder a esta funcionalidad, se debe ingresar la cédula del administrador: 3355479

```
Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas / finanzas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
Ingrese el número de la opción que va a utilizar:
```

Cuando el usuario desee consultar información relacionada a las estadísticas de ventas del concesionario, o información financiera como lo es el estado de resultados de la empresa, podrá acceder a ella eligiendo la opción 4.

En este caso, tendrá otras 4 opciones, estas son:

```
¿Qué estadísticas / info financiera quiere consultar?
1. Estado de Resultados
2. Estado de Resultados Detallado
3. Ventas - Vendedor
4. Ventas - Autos
Selecciona: [1-4]
```

Dado que el día actual y el nombre mes son variables que se utilizan en las 4 partes de la funcionalidad, procederé a explicar esta sección de código:

Se utiliza la clase `LocalDate` para obtener la fecha actual. Se crea un objeto fecha de tipo `LocalDate` y se obtiene el día del mes actual utilizando el método `getDayOfMonth()`. Además, se obtiene el nombre completo del mes actual utilizando el método `getMonth().getDisplayName(TextStyle.FULL, Locale.getDefault())`. Estos valores se almacenan en las variables `día` y `nombreMes`, respectivamente.

Opción 1:

En el caso que el usuario quiera ver el estado de resultados de la empresa y elija el 1, se crearán primero dos objetos de tipo `Array`, ('rubros' y 'utilidades') ambos que contendrán 4 valores de tipo `String`, la primera lista almacenará los rubros que alimentan el estado de resultados, como lo son las ventas totales, el costo de ventas, los gastos operacionales y de ventas y el impuesto de renta; por otro lado, la segunda lista tendrá los valores que resultan de las operaciones aritméticas hechas

a partir de los resultados de los rubros anteriores, estos son: utilidad bruta, utilidad operativa, utilidad antes de impuestos y utilidad neta.

Después se creará otro objeto de tipo Array vacío, llamado 'listaFinanzas', que almacenará 4 objetos de tipo long, y se pasará al método 'estResults' de la clase Transaccion, como se muestra a continuación:

```
listaFinanzas = Transaccion.estResults(listaFinanzas);
```

Posteriormente, este método lo que hará es, primeramente realizar la suma del total de ingresos del concesionario, es decir, empieza a sumar el atributo 'Ingreso' por medio del método 'getIngreso' de la clase Transaccion de todos los objetos de clase Transaccion, o de clases hijas de ella, como lo son: TransaccionVenta, TransaccionTaller, TransaccionVentaTaller y TransaccionModificacion. Esta sumatoria se almacena en el primer dato del Array que arrojará este método.

Después, se realiza la suma de los costos del concesionario, dato en el que incide el pago a los empleados del concesionario, es decir, vendedores y mecánicos, y para calcular este dato se realiza la suma del salario de cada uno de los vendedores, por medio del método getSalario de la clase trabajador, que extrae el atributo salario, y posteriormente se hace el mismo proceso con los mecánicos, la sumatoria se almacena en el segundo espacio del objeto de tipo Array.

En el tercer lugar del Array, estará el dato de la sumatoria de los gastos del concesionario, los cuales son las comisiones, que representan el 20% del total de ingresos por concepto de venta de autos, también está el gasto por pago de servicios y gastos fijos.

Y por último, está el valor correspondiente al impuesto de renta que el concesionario debe pagar, que se obtiene restando los costos y gastos a las ventas, y este valor se multiplica por 0.33, y para ingresarlo al Array se realiza un casteo explícito para que sea de tipo long.

Después de que el método estResults retornó el Array con estos valores, este resultado se asigna nuevamente a la variable 'listaFinanzas'

Después, se inicializa la variable 'res' de tipo long en 0. Luego, se inicia un bucle 'for' que se ejecuta 4 veces. Dentro del bucle, se realiza lo siguiente:

- En la primera iteración (cuando 'n' es igual a 0), se imprimen los rubros y los valores correspondientes de 'listaFinanzas'. Luego, se calcula la diferencia entre las ventas totales y el costo de ventas y se asigna a la variable 'res'.
- En las iteraciones siguientes, se imprime el nombre de la utilidad correspondiente y su valor actual, seguido de la descripción del rubro y su valor correspondiente de 'listaFinanzas'. Después de cada iteración, se resta

el valor del rubro actual de 'listaFinanzas' de la variable 'res'. En la última iteración, se imprime el nombre de la utilidad neta y su valor calculado.

Caso de uso:

Al realizar 3 compras por medio de la funcionalidad Ventas de Autos, 1 compra de un artículo de taller por medio de la funcionalidad Venta de Repuestos, y un servicio del taller por medio de la funcionalidad Taller, se imprimirá la siguiente información:

```
Selecciona: [1-4]
1
Estado de resultados desde el 1 hasta el 24 de mayo:
+ Ventas Totales: 350398900
- Costo de Ventas: 61122000
UTILIDAD OPERATIVA: 289276900
- Gastos Operacionales y de Ventas: 24000000
UTILIDAD ANTES DE IMPUESTOS: 265276900
- Impuesto de Renta: 87541377
UTILIDAD NETA: 177735523
```

Opción 2:

En el caso en el que el usuario elija la opción 2, que corresponde al estado de resultados detallado, el código imprime el "Estado de resultados detallado" desde el día 1 hasta el valor de "dia"(en el "nombreMes").

Primero, se crean dos objetos de tipo Array: "rubros2" y "utilidades2". Ambos contienen 4 valores de tipo String. "rubros2" almacena los rubros relacionados con el estado de resultados, como las ventas totales, el costo de ventas, los gastos operacionales y de ventas, y el impuesto de renta. Mientras tanto, "utilidades2" contiene las utilidades derivadas de los rubros anteriores, como utilidad bruta, utilidad operativa, utilidad antes de impuestos y utilidad neta.

A continuación, se crea un objeto de tipo Array llamado "listaFinanzas2" que almacena 4 valores de tipo long. Luego, se llama al método "estResults" de la clase Transaccion y se pasa el objeto "listaFinanzas2" como argumento.

```
listaFinanzas = Transaccion.estResults(listaFinanzas);
```

este método lo que hará es, primeramente realizar la suma del total de ingresos del concesionario, es decir, empieza a sumar el atributo 'Ingreso' por medio del método 'getIngreso' de la clase Transaccion de todos los objetos de clase Transaccion, o de clases hijas de ella, como lo son: TransaccionVenta, TransaccionTaller, TransaccionVentaTaller y TransaccionModificacion. Esta sumatoria se almacena en el primer dato del Array que arrojará este método.

Después, se realiza la suma de los costos del concesionario, dato en el que incide el pago a los empleados del concesionario, es decir, vendedores y mecánicos, y para calcular este dato se realiza la suma del salario de cada uno de los vendedores, por medio del método `getSalario` de la clase `trabajador`, que extrae el atributo `salario`, y posteriormente se hace el mismo proceso con los mecánicos, la sumatoria se almacena en el segundo espacio del objeto de tipo `Array`.

En el tercer lugar del `Array`, estará el dato de la sumatoria de los gastos del concesionario, los cuales son las comisiones, que representan el 20% del total de ingresos por concepto de venta de autos, también está el gasto por pago de servicios y gastos fijos.

Y por último, está el valor correspondiente al impuesto de renta que el concesionario debe pagar, que se obtiene restando los costos y gastos a las ventas, y este valor se multiplica por 0.33, y para ingresarlo al `Array` se realiza un casteo explícito para que sea de tipo `long`.

El resultado se asigna a la variable `"listaFinanzas"`.

Después, se inicializa la variable `"res2"` de tipo `long` en 0. Luego, se inicia un bucle `"for"` que se ejecuta 4 veces. Dentro del bucle, se realiza lo siguiente:

- En la primera iteración (cuando `"n"` es igual a 0), se imprimen los rubros y los valores correspondientes de `"listaFinanzas"`. Luego, se imprime información detallada sobre los ingresos por concepto de venta de autos, servicios del taller, venta de artículos del taller y modificaciones realizadas. También se muestra el pago a empleados, incluyendo vendedores y mecánicos. Finalmente, se calcula la diferencia entre las ventas totales y el costo de ventas y se asigna a la variable `"res2"`.
- En las iteraciones siguientes, se imprime el nombre de la utilidad correspondiente y su valor actual, seguido de la descripción del rubro y su valor correspondiente de `"listaFinanzas"`. Después de cada iteración, se resta el valor del rubro actual de `"listaFinanzas"` de la variable `"res2"`. En la tercera iteración, se imprime información detallada sobre las comisiones, servicios y gastos fijos. Por último, en la cuarta iteración, se imprime el nombre de la utilidad neta y su valor calculado.

Caso de uso:

Al realizar 3 compras por medio de la funcionalidad `Ventas de Autos`, 1 compra de un artículo de taller por medio de la funcionalidad `Venta de Repuestos`, y un servicio del taller por medio de la funcionalidad `Taller`, se imprimirá la siguiente información:

Selecciona: [1-4]

2

Estado de resultados detallado desde el 1 hasta el 24 de mayo:

+ Ventas Totales: 350398900

Ingresos por concepto de venta de autos:

Auto de marca Mazda, modelo 323: 30000000

Auto de marca Toyota, modelo Hilux: 230000000

Auto de marca Chevrolet, modelo Captiva: 90000000

Ingresos por concepto de servicios taller:

Mecánico(a) Sofia Hernandez, con valor de: 48900

Ingresos por concepto de venta de articulos taller:

Venta de Airtex con valor de: 350000

Ingresos por concepto de modificaciones:

- Costo de Ventas: 61122000

Pago a empleados:

Vendedores:

Juan Guaído, salario: 1800000.0

Pedro Mojica, salario: 1200000.0

María De la Paz, salario: 7600000.0

Luis Suarez, salario: 1207000.0

Ana Gabriel, salario: 1200000.0

Jorge Elias, salario: 1200000.0

Carla Jimenez, salario: 1200000.0

Oscar Velez, salario: 2200000.0

Mecanicos:

Carlos Martinez, salario: 1450000.0

Laura Hernandez, salario: 1450000.0

Mario Gonzalez, salario: 1450000.0

Daniela Ramirez, salario: 1450000.0

Manuel Torres, salario: 1450000.0

Lucia Perez, salario: 1450000.0

Andres Castro, salario: 1450000.0

Sofia Hernandez, salario: 1465000.0

Javier Diaz, salario: 1450000.0

Monica Rodriguez, salario: 1450000.0

Juadsadn Perez, salario: 1450000.0

Juaco Gomez, salario: 1450000.0

Pedro Rodriguez, salario: 1450000.0

Maria Martinez, salario: 1450000.0

Juan Lopez, salario: 1450000.0

Daniel Hurtado, salario: 1450000.0

Valentina Sanchez, salario: 1450000.0

Diego Jaramillo, salario: 1450000.0

Juan David Alzate, salario: 1450000.0

Miranda Catrillon, salario: 1450000.0

Nicolas Diaz, salario: 1450000.0

Carlos Alzate, salario: 1450000.0

Virginia Maldonado, salario: 1450000.0

Alejandro Narvaez, salario: 1450000.0

Argemiro Sanchez, salario: 1450000.0

Pink Floyd, salario: 1450000.0

Sara Bolivar, salario: 1450000.0

Juan Corrales, salario: 1450000.0

Sara Orrego, salario: 1450000.0

Josue Saenz, salario: 1450000.0

UTILIDAD OPERATIVA: 289276900

```

- Gastos Operacionales y de Ventas: 24000000
  Comisiones:
    Juan Guaido: 600000.0
    María De la Paz: 4600000.0
    María De la Paz: 1800000.0
  Servicios:
    Servicio de agua: 2000000
    Servicio de electricidad: 1500000
    Servicio de gas: 2500000
    Servicio de telefonía e internet: 1200000
    Servicio de seguridad: 1800000
    Servicio de limpieza: 1000000
  Gatos Fijos:
    Alquiler del local: 1500000
    Mantenimiento y reparaciones: 1000000
    Seguros: 2000000
    Publicidad y marketing: 1200000
    Costos financieros: 1300000
UTILIDAD ANTES DE IMPUESTOS: 265276900
  - Impuesto de Renta: 87541377
UTILIDAD NETA: 177735523

```

Opción 3:

En el caso de que se elija la opción 3, se imprimirá en pantalla información relacionada con las estadísticas de ventas, enfocadas en los vendedores del concesionario, de la siguiente manera:

Se declara un ArrayList llamado transaccionesActuales que almacena objetos de tipo TransaccionVenta. Estos objetos se obtienen llamando al método estático getTransaccionesven() de la clase TransaccionVenta.

Se declara otro ArrayList llamado vendedores1 que almacenará objetos de tipo Vendedor. Estos objetos se obtienen llamando al método vendedoresVentas(transaccionesActuales) de la clase TransaccionVenta. Este método recibe como argumento la lista de transacciones actuales y devuelve una lista filtrada de vendedores que han realizado ventas:

```
ArrayList<Vendedor> vendedores1 = TransaccionVenta.vendedoresVentas(transaccionesActuales);
```

Se realiza un cálculo para obtener el porcentaje de vendedores que han realizado ventas en relación al total de vendedores. Se utiliza la fórmula: $(\text{vendedores1.size()} / \text{Vendedor.getVendedores().size()}) * 100$. El resultado se almacena en la variable num, que es de tipo float. Luego, se redondea el valor utilizando el método Math.round(num), y el resultado se guarda en la variable roundedNum, de tipo int.

Se imprime un mensaje en la consola que muestra la cantidad de vendedores que han logrado ventas en el mes, utilizando los métodos `System.out.println()` y concatenación de cadenas. Se accede a las propiedades de los objetos `Vendedor` y se utiliza el método `size()` de `vendedores1` para obtener la cantidad de vendedores.

Se utiliza un bucle `for` para iterar sobre la lista de vendedores (`vendedores1`) y se imprime el nombre de cada vendedor utilizando el método `getNombre()`.

Se utiliza otro bucle `for` para iterar sobre la lista de transacciones de ventas (`TransaccionVenta.getTransaccionesven()`) y se realiza lo siguiente: Se acumula la suma total de ingresos en la variable `sumaTotal` y se incrementa el contador `contadorTotal` para obtener el número total de ventas.

Se imprime un mensaje que muestra la suma de dinero en ventas por cada vendedor. Se utiliza un bucle `for` para iterar sobre la lista de vendedores (`vendedores1`) y otro bucle `for` para iterar sobre la lista de transacciones de ventas (`TransaccionVenta.getTransaccionesven()`). Se utilizan condiciones (`if`) para verificar si el vendedor de una transacción coincide con el vendedor actual en el bucle externo. Se acumulan las sumas de ingresos y los contadores correspondientes, y se calcula el porcentaje de ingresos por ventas utilizando la fórmula $(\text{suma} / \text{sumaTotal}) * 100$. Los resultados se redondean utilizando `Math.round()` y se imprimen en la consola.

Se imprime un mensaje que muestra el número de ventas y el promedio de ingreso por venta de cada vendedor. Se utilizan bucles `for` similares al paso anterior para iterar sobre los vendedores y las transacciones de ventas.

Se acumulan las sumas de ingresos y los contadores correspondientes. Se calcula el promedio de ingreso por venta dividiendo la suma total entre el contador.

Caso de uso:

Al realizar 3 compras por medio de la funcionalidad Ventas de Autos, se imprimirá la siguiente información:

```
Selecciona: [1-4]
3

----- Los 8 vendedores, han logrado 5 ventas en el mes, promediando 0.625 ventas por vendedor -----
Juan Guaido: 30000000 $, el 9 % del total de ingresos por ventas de autos.
Maria De la Paz: 230000000 $, el 66 % del total de ingresos por ventas de autos.
María De la Paz: 90000000 $, el 26 % del total de ingresos por ventas de autos.

----- Suma de ingresos total, y promedio de ingresos diarios en lo corrido del mes de mayo: -----
Suma total de ingresos: 350000000, promedio de ingresos diarios en lo corrido del mes de mayo: 14583333 $.
```

Opción 4:

Se declara un ArrayList llamado autosIniciales que almacena objetos de tipo Auto. Estos objetos se obtienen llamando al método estático getAutosV() de la clase TransaccionVenta, que retorna los autos vendidos hasta el momento.

Se declara otro ArrayList llamado marcasVentas que almacenará cadenas de texto representando las marcas de los autos vendidos. Estas marcas se obtienen llamando al método marcasVentas(autosIniciales) de la clase TransaccionVenta.

```
ArrayList<String> marcasVentas = TransaccionVenta.marcasVentas(autosIniciales);
```

Este método recibe como argumento la lista de autos iniciales y devuelve una lista de marcas únicas de los autos vendidos.

Se realiza un cálculo para obtener el porcentaje de autos vendidos en relación al total de autos iniciales. Se utiliza la fórmula: $(\text{autosIniciales.size() / InventarioAuto.getAutos().size()}) * 100$. El resultado se almacena en la variable numA, que es de tipo float. Luego, se redondea el valor utilizando el método Math.round(numA), y el resultado se guarda en la variable roundedNumA, de tipo int.

Se imprime un mensaje en la consola que muestra la cantidad de autos vendidos y su porcentaje en relación al total de autos iniciales. Se accede a las propiedades de los objetos Auto y se utiliza el método info() para obtener información detallada de cada auto.

Se realizan cálculos para obtener la suma total de ingresos por ventas y el número total de ventas. Se utiliza un bucle for para iterar sobre la lista de transacciones de ventas (TransaccionVenta.getTransaccionesven()), acumulando los valores correspondientes en las variables sumaTotal2 y contadorTotal2.

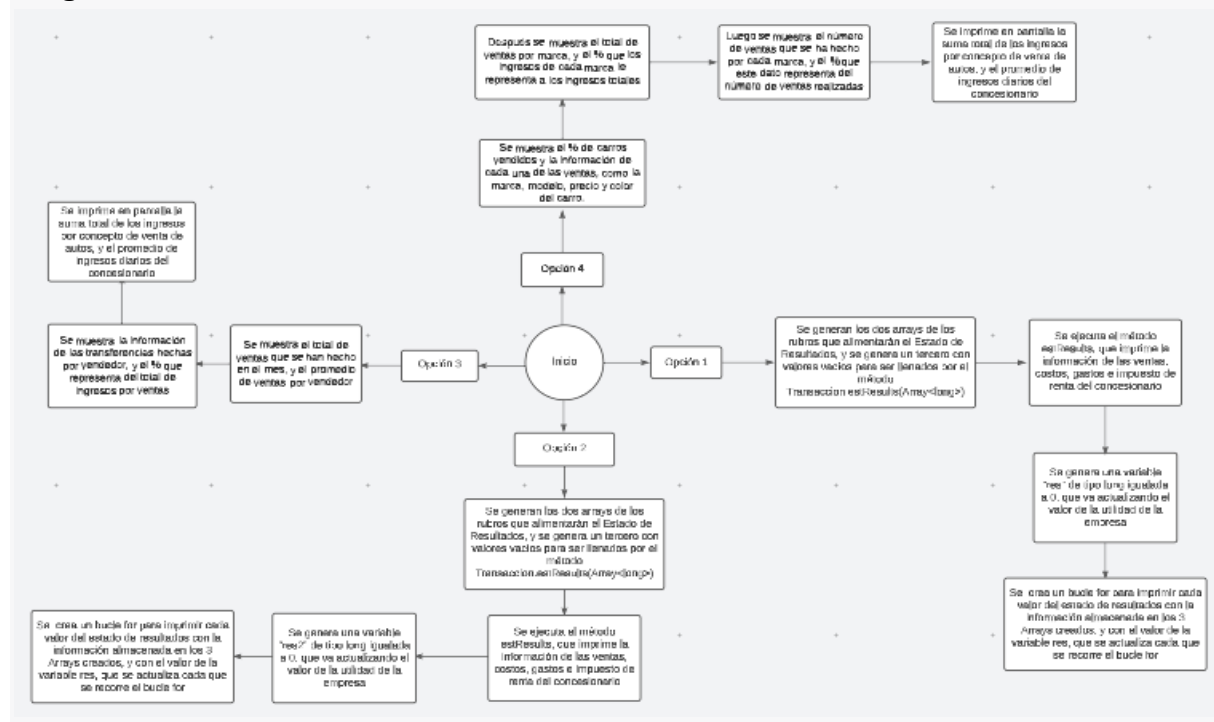
Se imprime un mensaje que muestra las ventas en términos de dinero por cada marca de auto. Se utiliza un bucle for para iterar sobre la lista de marcas de autos vendidos (marcasVentas) y otro bucle for para iterar sobre la lista de autos iniciales (autosIniciales). Se utilizan condiciones (if) para verificar si la marca de un auto coincide con la marca actual en el bucle externo. Se acumulan las sumas de precios de venta y se calcula el porcentaje en relación a la suma total de ingresos utilizando la fórmula $(\text{suma} / \text{sumaTotal2}) * 100$. Los resultados se redondean utilizando Math.round() y se imprimen en la consola.

Se imprime un mensaje que muestra el número de ventas por cada marca de auto. Se utilizan bucles for similares al paso anterior para iterar sobre las marcas de autos vendidos y los autos iniciales. Se utiliza un contador para contar las ocurrencias de una marca en la lista de autos iniciales. Se calcula el porcentaje en relación al

Se calcula el promedio de ingresos diarios dividiendo la suma total de ingresos entre el número de días (dia) y se redondea el resultado. Se imprime un mensaje que muestra la suma total de ingresos y el promedio de ingresos diarios en el mes actual.

Al realizar 4 compras por medio de la funcionalidad Ventas de Autos, se imprimirá la siguiente información:

Diagrama de Fases:



Link:

https://lucid.app/lucidspark/c9314b07-98bb-41e9-8676-b07cf188dbc4/edit?beaconFlowId=49E3BFE4E202C934&invitationId=inv_35369939-56c6-4f1b-ac94-f11bbf94fccf&page=0_0#

- **PERSONALIZAR SU AUTO**

La funcionalidad permite realizar modificaciones y mejoras en tu automóvil de manera personalizada. Con esta característica, podrás seleccionar entre diversas opciones de personalización, como cambios en la pintura, llantas, sistema de sonido, frenos o escape, según tus preferencias y necesidades, teniendo la opción de usar el taller del concesionario para hacer las modificaciones o solo hacer la compra de los artículos que desees para tu modificación

```
Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas / finanzas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
Ingrese el número de la opción que va a utilizar:
```

El programa comienza con un mensaje de bienvenida a la sección de personalización de automóviles.

El usuario debe ingresar la cédula del propietario del automóvil.

```
Bienvenido a nuestra sección de personalización de automóviles.
Introduzca la cédula del propietario: 1234567890
Nombre del cliente: María González
```

Se utiliza el método **"getClienteporCedula(cedula)"** de la clase **"TransaccionVenta"** para obtener el objeto Cliente correspondiente a la cédula ingresada. También se utiliza el método **"getTransaccionporCedula(cedula)"** para obtener el objeto Auto asociado al propietario.

Si tanto el propietario como el auto son nulos, se muestra un mensaje de error y se solicita nuevamente la cédula.

```
Nombre del cliente: Mafalda González
```

Si se encuentra un propietario y un auto asociado a la cédula ingresada, se muestra la información del propietario mediante el método **"info()"** de la clase **Cliente**.

```
¿Confirmar propietario? (si/no): si
¿Desea utilizar el taller con mecánicos o solo desea asignar un vendedor? (taller/vendedor):
```

El programa solicita al usuario que confirme si el propietario es correcto. Si el usuario responde "no", se repite el proceso de ingreso de cédula y búsqueda de propietario y auto y a continuación, se le pregunta al usuario si desea utilizar el taller con mecánicos o solo asignar un vendedor. Esta selección se almacena en la variable **"opcionTaller"**.

```
Ha seleccionado utilizar nuestro menu de ventas de articulos

Que desees hacerle al Vehiculo
1. Modificacion de pintura
2. Modificacion de Llantas
3. Modificacion del sonido
4. Modificacion de frenos
5. Modificacion del escape
Ingrese el número de la opción que va a utilizar:
```

Si la opción elegida es "taller", se muestra un mensaje de bienvenida y se presenta un menú de opciones de modificación disponibles para el vehículo. Las opciones incluyen la modificación de pintura, llantas, sonido, frenos y escape. y se le pide al usuario que ingrese el numero de la modificacion que desee utilizar

```
Los mecanicos que atienden Mazda disponibles son:
```

	Nombre	Cedula	Atiende	Especialidad
1	Virginia Maldonado	1234567906	Mazda	ModificacionLlantas

El programa utiliza el método **"mecanicoModif(auto)"** de la clase **"Mecanico"** para obtener una lista de mecánicos que pueden realizar modificaciones en el automóvil proporcionado. La lista de mecánicos se muestra por pantalla, mostrando información como el nombre, cédula, autos atendidos y especialidad de cada mecánico.

```
Seleccione el numero del mecanico[1-1]:
1
Los productos ModificacionLlantas disponibles son:
```

	Referencia	Producto	Tipo Vehiculo	Marca	Precio	Cantidad
1	1004	Llanta todo terreno	automovil y camioneta	GoodYear	745000	100
2	1005	Llanta todo terreno	automovil y camioneta	Michelin	1150000	99
3	1006	Llanta terreno de barro	automovil y camioneta	Bridgestone	750000	100
4	1007	Llanta terreno de barro	automovil y camioneta	Yokohama Geolanda	1899000	100
5	1008	Llanta terreno de asfalto	automovil y camioneta	Goodyear	650000	100
6	1009	Llanta terreno de asfalto	automovil y camioneta	Michelin	1000000	100
7	4567	Llantas todoterreno	automovil	Michelin	200000	50

El usuario debe seleccionar un mecánico de la lista ingresando el número correspondiente.

Luego, se utiliza el método **"articuloDispo(mecanico)"** de la clase **"InventarioArticulo"** para obtener una lista de productos disponibles que el mecánico puede utilizar para la modificación del vehículo. La lista de productos se muestra por pantalla, mostrando información como la referencia, producto, tipo de vehículo, marca, precio y cantidad disponible de cada artículo.

El usuario debe seleccionar un producto de la lista ingresando el número correspondiente.

```
¿Confirma el procedimiento, el mecánico y el producto? (si/no): si
Horario disponible:
1. 9:00-11:00
2. 11:00-1:00
3. 2:00-4:00
4. 4:00-6:00
```

Se solicita al usuario que confirme la selección del procedimiento, el mecánico y el producto. Si el usuario responde "si", se procede con la personalización.

```
Seleccione el número de hora [1-4]: 4
La hora de su cita es: 4:00-6:00
```

Se muestra el horario disponible del mecánico seleccionado y se solicita al usuario que seleccione la hora de la cita.

Se muestra la hora de la cita seleccionada y se elimina esa hora del horario disponible del mecánico con el método **"mecanico.getHorario().remove(num1 - 1);"** de la clase **"Mecanico"**

```
El costo total de la transacción es: 680000
```

Se calcula el costo total de la transacción sumando la mano de obra del mecánico y el precio del artículo seleccionado.

Se genera un número de transaccion aleatorio y se crea un objeto **"TransaccionModificacion"** con la información relevante de la personalización del automóvil

Se muestra la información de la transacción realizada, incluyendo el propietario, el auto, el mecánico, el producto, el horario de la cita y el costo total.

```
Transacción #00000376: taller realizado por Virginia Maldonado para el cliente Mafalda González que adquirio el producto:
```

```
Llanta terreno de asfalto de la marca: Goodyear , con precio total $650000
```

```
¿Desea utilizar el taller con mecánicos o solo desea asignar un vendedor? (taller/vendedor): vendedor
Ha seleccionado utilizar nuestro menu de ventas de articulos
```

```
Que deseas hacerle al Vehiculo
1. Modificacion de pintura
2. Modificacion de Llantas
3. Modificacion del sonido
4. Modificacion de frenos
5. Modificacion del escape
Ingrese el número de la opción que va a utilizar:
```

Si la opción elegida es "vendedor", se muestra un mensaje de bienvenida y se presenta el mismo menú de opciones de modificación disponibles para el vehículo.

```

Los productos Llantas disponibles son:

Referencia  Producto                                Tipo Vehiculo  Marca                                Precio  Cantidad
1  1004      Llanta todo terreno                        automovil y camioneta  Goodyear  745000  100
2  1005      Llanta todo terreno                        automovil y camioneta  Michelin  1150000  99
3  1006      Llanta terreno de barro                    automovil y camioneta  Bridgestone  750000  100
4  1007      Llanta terreno de barro                    automovil y camioneta  Yokohama Geolanda  1899000  100
5  1008      Llanta terreno de asfalto                  automovil y camioneta  Goodyear  650000  100
6  1009      Llanta terreno de asfalto                  automovil y camioneta  michelin  1000000  100
7  4567      Llantas todoterreno                        automovil          Michelin  200000  50

Seleccione el numero del producto[1-7]:
5
Los vendedores de Repuestos disponibles son:

```

Se muestra la lista de productos disponibles para la modificación seleccionada y el tipo de automóvil. El usuario selecciona un producto de la lista.

```

Los vendedores de Repuestos disponibles son:

Vendedor                                Tipo de venta
1  Pedro Mojica                          Repuestos
2  Luis Suarez                            Repuestos
3  Jorge Elias                            Repuestos

Seleccione el numero del vendedor[1-3]:
1

```

Se utiliza el método **"Vendedor.selectorVend(articulo)"** de la clase **"Vendedor"** para obtener una lista de vendedores que pueden ayudar con la modificación del automóvil proporcionado. La lista de vendedores se muestra por pantalla, mostrando información como el nombre, cédula y especialidad de cada vendedor.

```

Seleccione el numero del vendedor[1-3]:
1
¿Confirmar vendedor? (si/no)si
El costo total de la transacción es: 650000

```

El usuario debe seleccionar un vendedor de la lista ingresando el número correspondiente.

Se solicita al usuario que confirme la selección del vendedor. Si el usuario responde "si", se procede con la personalización.

Se muestra el costo total de la transacción, que corresponde al precio del artículo seleccionado.

```
El costo total de la transacción es: 650000
```

```
¿Desea confirmar la transaccion? (si/no)si
```

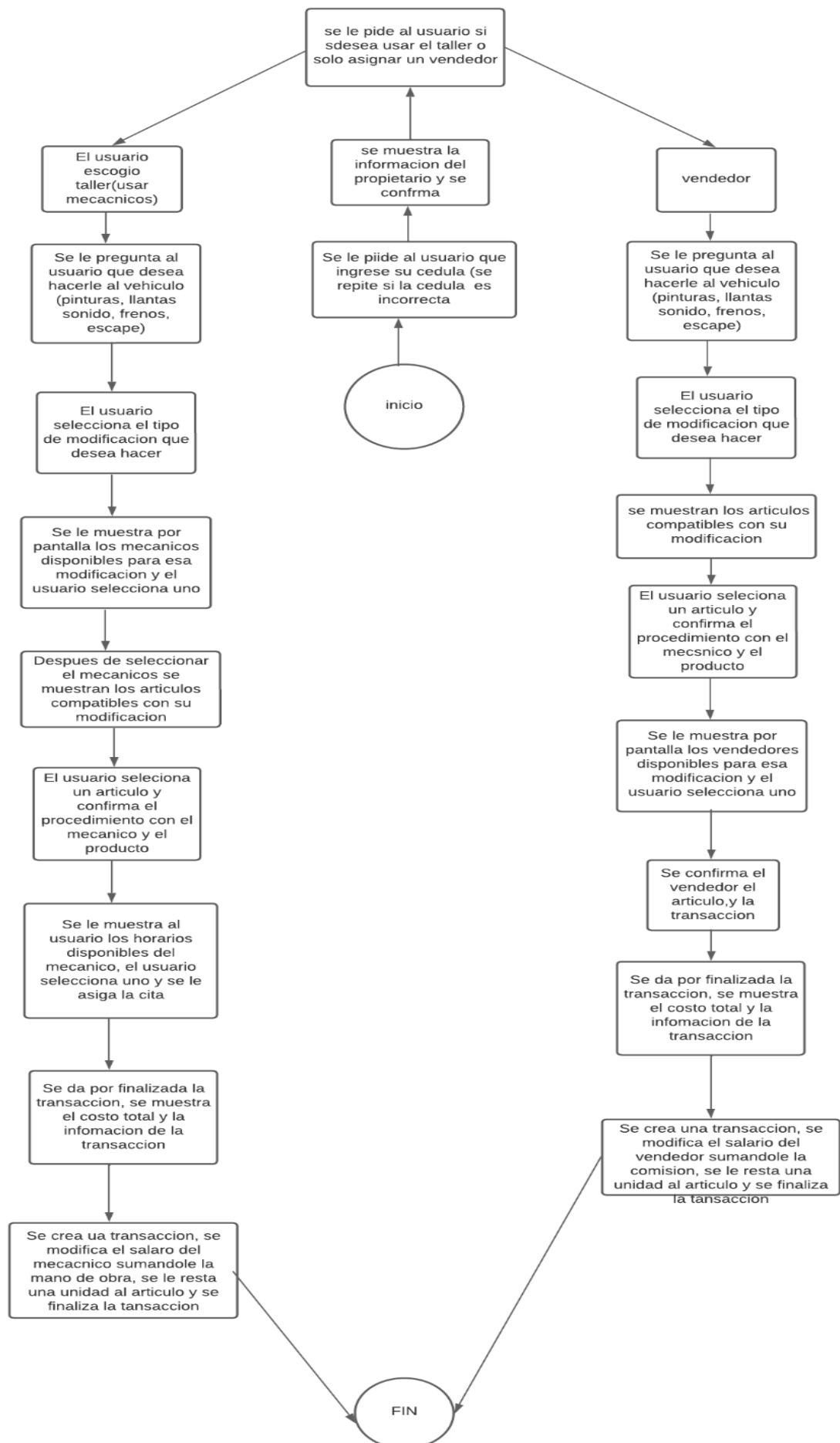
```
Transacción #00000881: venta realizada por Pedro Mojica para el cliente Mafalda González que adquirio el producto: L
```

```
Llanta terreno de asfalto de la marca: Goodyear , con precio total $650000
```

Se solicita al usuario que confirme la transacción. Si el usuario responde "si", se procede con la actualización de las ventas del vendedor y el stock de los productos modificados.

Se genera un número de transferencia aleatorio y se crea un objeto **"TransaccionModificacion"** con la información relevante de la personalización del automóvil.

Se muestra la información de la transacción realizada, incluyendo el propietario, el auto, el vendedor, el producto y el costo total.



- **ADMINISTRACIÓN**

Importante: para ingresar a esta funcionalidad la cédula que debe de poner es 3355479.

La funcionalidad de administración se usa más que todo para agregar nuevos registros de objetos y para reiniciar los horarios de los mecánicos. Esta funcionalidad empieza entrando desde el menú principal eligiendo la opción:

```
Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas de ventas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
7. Administración
8. Salir
```

Cabe destacar que el apartado de administración solo puede ser usado por trabajadores con el cargo de administrador, por lo que el sistema le pide al usuario su cédula para entrar, por lo que se crea una nueva variable llamada admin, la cual se le asigna un objeto en caso de que al usar el método ***Vendedor.getVendedorPorCedula(cedula)*** retorne el objeto Vendedor con cargo de administrador, de ser exitoso, en pantalla se muestra el siguiente menú:

```
Introduzca su cedula
3355479
¿Que labor administrativa desea hacer?
1. ¿Asignar Horarios a mecanicos?
2. Añadir articulo (Nuevo)
3. Añadir Auto (Nuevo)
4. Añadir Vendedor (Nuevo)
5. Añadir Mecánico (Nuevo)
6. Salir del Portal administrativo
```

1. asignar horarios a mecánicos.

Si elegimos la primera opción de asignar horarios a mecánicos se nos va a pedir la cédula del mecánico al cual queremos volver a asignarle sus horarios, luego de introducir la cédula el sistema va a crear un apuntador “mech” de tipo Mecanico, y su valor será asignado por ***Mecanico.getMecanicoPorCedula(cedulamecanico)***,

luego, a mech se le agrega a un arraylist que tiene en sus atributos nuevamente la disponibilidad de sus horarios a través del método ***mech.setHorario(arrayList<String> horario)***, el cual reemplaza el arraylist actual por uno nuevo con el horario del mecánico reiniciado. Luego en pantalla muestra que ha podido reasignar el horario y le pregunta al usuario si quiere volver al menú:

```
¿Que labor administrativa desea hacer?
1. ¿Asignar Horarios a mecanicos?
2. Añadir artículo (Nuevo)
3. Añadir Auto (Nuevo)
4. Añadir Vendedor (Nuevo)
5. Añadir Mecánico (Nuevo)
6. Salir del Portal administrativo
1
¿Introduzca la cedula del mecanico?
1234567899
Horario Reasignado a Daniel Hurtado

¿Desea volver al menú principal? (si/no): si

Menú principal Concesionario
1. Venta de Autos
2. Venta de Repuestos
3. Taller
4. Consultar estadísticas de ventas
5. Personalizar su auto
6. Crear nuevo usuario (Comprador)
```

2. Añadir artículo

Si elegimos la opción de añadir un nuevo artículo el sistema nos va a pedir los siguientes datos uno a uno:

1. Nombre del artículo (String)
2. Referencia del artículo (long)
3. Calidad del artículo (String)
4. Tipo de artículo (String)
5. Especialidad (String)
6. Tipo del vehículo (String)
7. Marca del artículo (String)
8. Precio del artículo (long)
9. Número de unidades (int)
10. marca específica (String)

Observaciones: Cuando se ingresa la referencia del artículo el sistema comprueba que la referencia introducida no esté repetida, de lo contrario el proceso de adicción se detiene debido a que la referencia ya está registrada.

Cuando se pide el tipo de vehículo existen 3 opciones, automóvil, camioneta o ambos, es decir automóvil y camioneta.

Si se define que el artículo es para una marca en especial el sistema le va a pedir al usuario que escoja la marca para la cual el artículo está destinado a usar.

```
4. Añadir Mecánico (Nuevo)
5. Añadir Mecánico (Nuevo)
6. Salir del Portal administrativo
2
Nombre del articulo
Aceite sintetico
Referencia del articulo
10001
Calidad del articulo (premium/basico)
premium
Tipo del articulo (repuesto/taller)
taller
Especialidad (Llantas/Pintura/Motor/Frenos/Modificacion)
Motor
Tipo del Vehiculo
automovil y camioneta
Marca del articulo
Mobil SUP 1000 20W-50
precio del articulo
38700
Numero de unidades
50
¿Es para una Marca especial? (si/no)
si
¿Cual? (Mazda/Toyota/Chevrolet)
Mazda
Articulo creado con exito

¿Desea volver al menú principal? (si/no): si
```

3. Añadir auto

Esta opción permite añadir un nuevo objeto tipo auto, el sistema pide los siguientes datos:

1. Modelo del carro (String)
2. Marca del carro (String)
3. Precio del carro (long)
4. Cilindraje del carro (int)
5. Color del carro (String)
6. FullEquipo (String)

Observaciones: Las marcas que se manejan en nuestro concesionario son Mazda, Toyota y Chevrolet.

```
Modelo del carro:
mx-5
Marca del carro:
Mazda
Precio del carro:
2500000000
Cilindraje del carro:
2000
Color del carro:
Rojo
El carro es Full Equipo? (si/no):
si
El carro Mazda, modelo mx-5, FullEquipo se ha añadido con éxito
```

4. Añadir vendedor

La opción de añadir vendedor le pide al usuario los siguientes datos:

1. Nombre del vendedor (String)
2. Cédula del vendedor (long)
3. Teléfono/Celular del vendedor (long)
4. Correo del vendedor (String)
5. Dirección del vendedor (String)
6. Salario del vendedor (double)
7. Banco del vendedor (String)
8. Número de cuenta (long)
9. Puesto del vendedor (String)

Observaciones: Los puestos que se manejan en el concesionario de los vendedores son de administrador, vendedor de vitrina y vendedor de repuestos (admin, Vitrina, Repuestos).

```
Nombre del vendedor:
Emmanuel Barrientos
Cédula del vendedor:
45578866
Teléfono del vendedor:
317799466
Correo del vendedor:
emmaba@gmail.com
Dirección del vendedor:
Cl 1 #45-12
Salario del vendedor:
2800000
Banco del vendedor:
Bancolombia
Numero de cuenta de banco del vendedor:
9812277818
Puesto del vendedor(admin,Vitrina,Repuestos):
Repuestos
El vendedor Emmanuel Barrientos con Cédula 45578866, ha sido añadido con éxito.
```

5. Añadir mecánico

La opción de añadir mecánico le pide al usuario los siguientes datos:

1. Nombre del mecánico (String)
2. Cédula del mecánico (long)
3. Teléfono/Celular del mecánico (long)
4. Correo del mecánico (String)
5. Dirección del mecánico (String)
6. Salario del mecánico (double)
7. Banco del mecánico (String)
8. Número de cuenta (long)
9. Marca de carros que atiende (String)
10. Especialidad del mecánico (String)
11. Valor de mano de obra (long)

Observaciones: las especialidades del mecánico son las partes del carro o área específica en la que trabaja el mecánico (Llantas/Pintura/Motor/Frenos/Modificación)

```
Nombre del mecánico:
Santiago Florez
Cédula del mecánico:
4135121000
Teléfono del mecánico:
318387559
Correo del mecánico:
sflorezz@gmail.com
Dirección del mecánico:
Cr 45 # 23-2
Salario del mecánico:
3200000
Banco del mecánico:
Bancolombia
Número de cuenta de banco del mecánico:
9912371239
Marca de carros que atiende el mecánico:
Mazda
Especialidad del mecánico:
Motor
Valor de mano de obra del mecánico:
60000
El mecánico Santiago Florez con Cédula 4135121000, ha sido añadido con éxito.
```

Descripción de la app

La aplicación del concesionario de autos es una herramienta integral que ofrece múltiples servicios para satisfacer todas las necesidades de los clientes en relación con la compra y mantenimiento de vehículos de las marcas Mazda, Toyota y Chevrolet.

A través de la app, los usuarios pueden acceder al catálogo de vehículos disponibles, visualizar sus características, precios y realizar compras de forma segura y sencilla desde la plataforma. Además, aquellos clientes que ya hayan adquirido un vehículo con nosotros podrán gestionar el proceso de taller y solicitar servicios de mantenimiento.

La app también dispone de una sección dedicada a la venta de repuestos y accesorios para los vehículos de las marcas mencionadas, con la posibilidad de buscar y encontrar piezas específicas.

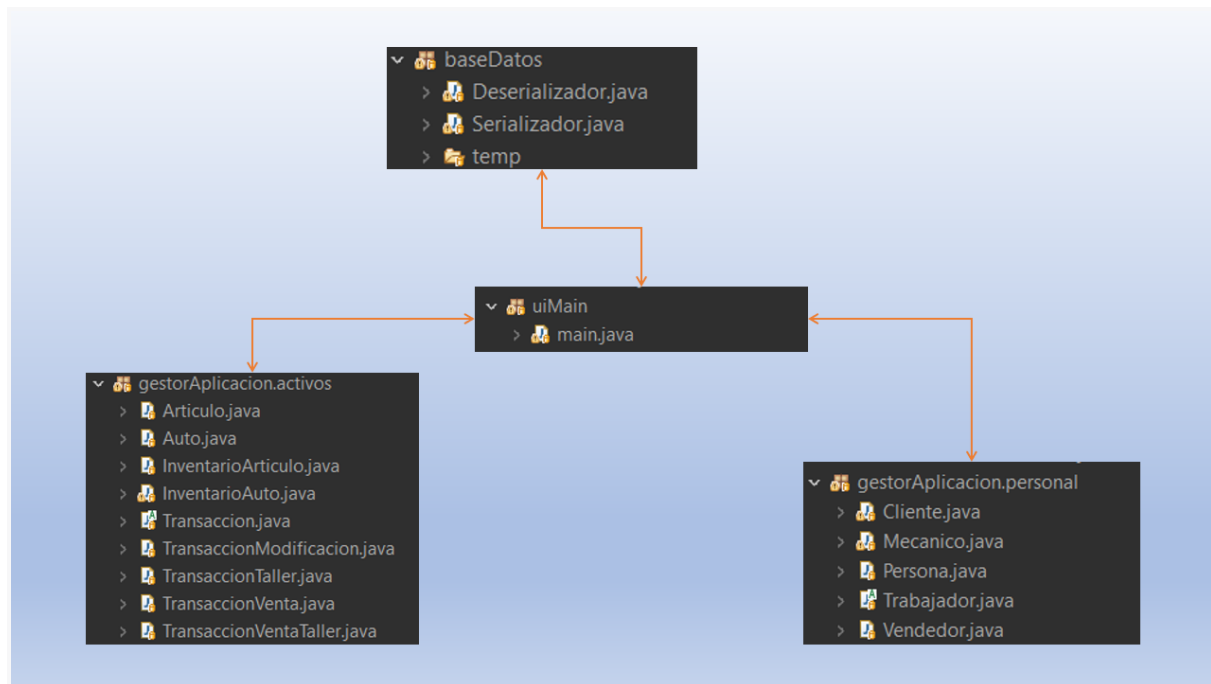
La sección de personalización de autos es una de las características más destacadas de la aplicación, permitiendo a los usuarios configurar y personalizar sus vehículos según sus necesidades y preferencias. Desde la app, los usuarios pueden seleccionar diferentes opciones de colores, equipamiento y características especiales para sus vehículos.

La sección de estadísticas permite a los usuarios visualizar las ventas del concesionario, estadísticas de los vehículos más vendidos y otras métricas relevantes. Además, la sección de administración permite a los gerentes y propietarios del concesionario administrar la aplicación y realizar cambios en el inventario, añadir nuevos vendedores y mecánicos, actualizar precios y mucho más.

En definitiva, la aplicación del concesionario de autos es una herramienta completa y útil para los clientes que desean adquirir vehículos, mantenerlos, personalizarlos y comprar repuestos y accesorios. Además, ofrece funciones de estadísticas y administración para los gerentes del concesionario, lo que se traduce en un servicio más completo y satisfactorio para nuestros clientes.

Encapsulamiento en paquetes:

El empaquetamiento del programa se ha hecho partido en dos paquetes, cada uno categorizado se ha categorizado a criterio de la similitud de tipos de objetos para ser manejados y encontrados con mejor organización.



Serialización y deserialización:

Al momento de ejecutar el programa se carga una serie de diferentes objetos para el funcionamiento correcto de la aplicación. Estos objetos al ser inicializados automáticamente se insertan en un ArrayList creado para cada clase de objeto, luego, estos arrays son serializados en diferentes archivos cada uno, con el fin de manejar una mejor organización y manipulación de los objetos en caso de necesitarlo.

La serialización se lleva a cabo cuando se termina el programa por alguna de las diferentes opciones para salir del programa, y deserializar cuando se inicializa el mismo.

```
public static <T> void deserializarArrays(ArrayList<T> array , String name) {  
    File archivo = new File("");  
    try {  
        File ruta = new File(archivo.getAbsolutePath()+"/src/baseDatos/temp/"+name+".txt");  
        ObjectInputStream recuperando_fichero = new ObjectInputStream(new FileInputStream(ruta));  
        array.addAll((ArrayList<T>) recuperando_fichero.readObject());  
        recuperando_fichero.close();  
    } catch (FileNotFoundException e) {  
        System.out.print("FileNotFoundException");  
        e.printStackTrace();  
    } catch (IOException e) {  
        System.out.print("IOException");  
        e.printStackTrace();  
    } catch (ClassNotFoundException e) {  
        System.out.print("ClassNotFoundException");  
        e.printStackTrace();  
    }  
}
```

```
public static void serializarArrays(ArrayList<? extends Object> array, String name) {  
    File archivo = new File("");  
  
    try{  
        File ruta = new File(archivo.getAbsolutePath()+"/src/baseDatos/temp/"+name+".txt");  
  
        ObjectOutputStream escribiendo_fichero = new ObjectOutputStream(new FileOutputStream(ruta));  
  
        escribiendo_fichero.writeObject(array);  
  
        escribiendo_fichero.close();  
    }  
    catch (FileNotFoundException e) {  
        System.out.print("FileNotFoundException");  
    } catch (IOException e) {  
        System.out.print("IOException");  
    }  
}
```


Manual de Usuario:

Funcionalidad VENTA DE AUTOS:

En esta funcionalidad se muestran los autos disponibles según la marca de interés del comprador, con sus respectivos precios y colores. Si solo hay un automóvil disponible, se le pregunta al usuario si desea seleccionarlo. Si hay más de un automóvil disponible, se muestra una lista numerada de autos y se le pide al usuario que seleccione uno.

Después de la selección del automóvil, se presentan varias opciones de búsqueda adicionales (por marca, precio, o una opción para mostrar todos los autos.) y se le pide al usuario que seleccione una opción. Si el usuario elige una búsqueda por marca, se muestran los autos disponibles de esa marca y se le pide al usuario que seleccione uno. Si el usuario elige una búsqueda por precio, se le mostrarán los autos que puede comprar en relación del presupuesto que han registrado. Si el usuario elige un automóvil, se le presenta una opción para confirmar su selección.

Finalmente, se selecciona el vendedor presente para la transacción, se muestra una lista de los vendedores disponibles y se le pide al usuario que seleccione el que le ha atendido. Luego, se asigna al carro seleccionado su dueño, al comprador se le asigna el carro designado, al vendedor se le atribuye una compra y se crea una transacción con todos estos datos, esto sí el comprador tiene el presupuesto suficiente para comprar el carro.

Funcionalidad TALLER:

Este método es el que lleva la función del proceso de taller en nuestro concesionario. El proceso se inicia solicitando la cédula del propietario y luego se busca las transacciones si el propietario tiene una transacción y un automóvil asociado a esa transacción. Si no se encuentra ningún registro, se solicita al usuario que ingrese nuevamente la cédula del propietario.

Una vez que se encuentra el propietario y su automóvil, se presenta la información del propietario y se solicita la confirmación del propietario. Si se confirma, se muestra un menú con las opciones de reparación disponibles para el vehículo, que incluyen latonería y pintura, cambio de llantas y alineación, cambio de aceite y cambio de frenos.

A continuación, se presenta una lista de mecánicos que pueden trabajar en el automóvil, ya que cada uno de ellos tiene una especialidad y una marca diferente a la cual atienden. Se muestra el nombre del mecánico, la cédula, y su especialidad. El usuario selecciona el mecánico que desea que repare su vehículo.

Si se encuentra al menos un mecánico disponible, se presenta una lista de productos disponibles para hacer la reparación. La lista incluye la referencia del

producto, su tipo, tipo de vehículo, marca, precio y cantidad. El usuario selecciona los productos que desea utilizar durante la reparación.

En general, este método maneja todo el proceso de reparación de un vehículo en un taller, desde la selección del propietario y el vehículo hasta la selección del mecánico y los productos necesarios.

Funcionalidad VENTA DE REPUESTOS:

Esta es la funcionalidad venta de repuestos. Al ejecutarse, le pide al usuario que ingrese la cédula del comprador y la busca en una lista de clientes preexistentes. Si encuentra un cliente con la cédula ingresada, procede a pedirle que seleccione el tipo de repuesto que desea comprar, la marca de su vehículo y la calidad que desea. Luego, muestra los artículos disponibles según los criterios seleccionados y le pide al usuario que seleccione el artículo que desea comprar. Una vez que el usuario ha confirmado la compra, el sistema le muestra el precio del artículo y le pide que confirme si desea comprarlo o no.

Si el usuario confirma la compra, el sistema calcula el total de la compra, le pide al usuario que seleccione un vendedor y le muestra la información de la compra, incluyendo el precio total y la fecha de la transacción. Finalmente, guarda la transacción.

Funcionalidad CONSULTAR ESTADÍSTICAS / FINANZAS:

NOTA: Para acceder a esta funcionalidad, se debe ingresar la cédula del administrador: 3355479

Esta funcionalidad le pide al usuario que elija, si quiere consultar las estadísticas de ventas haciendo énfasis en la marca de los carros vendidos, o si las quiere consultar haciendo énfasis en las ventas por vendedores; o si por el contrario, quiere ver el estado de resultados simple del concesionario, o el estado de resultados detallado, en el que se muestra de dónde salen cada uno de los rubros mostrados.

En caso de que el usuario desee observar las estadísticas de las ventas, enfocadas en las ventas de los vendedores, se desplegará la siguiente información:

- El número de vendedores que se encuentran registrados en el sistema, y el número de vendedores que han realizado ventas, acompañado del porcentaje que representa este dato (vendedores con ventas / total de vendedores). También se muestran los nombres de los vendedores con ventas.

- Se muestra también la suma de dinero que ha ingresado según las ventas de cada uno de los vendedores que ha logrado realizar ventas, y el porcentaje que estos ingresos le representan a los ingresos totales del concesionario.
- También se muestra el número de ventas que ha hecho cada vendedor, y el porcentaje que este dato representa del total de ventas del concesionario.
- Por último, se muestra la suma total de ingresos del concesionario, y el promedio de ingresos diarios del concesionario (para este dato se asume que las ventas realizadas solo corresponden al mes en el que se está ejecutando el programa)

En caso de que el usuario desee observar las estadísticas de las ventas, enfocadas en los ingresos por concepto de venta de autos, se desplegará la siguiente información:

- El número de autos que se tenían al comienzo del mes de mayo y el número de autos que se han vendido, acompañado del porcentaje que representa este dato (autos vendidos / total de autos). También se muestran los datos de los autos que se han vendido, incluyendo el modelo, la marca, el precio, el color y si es full equipo.
- Se muestra la suma de dinero que ha ingresado por la venta de cada marca de auto y el porcentaje que estos ingresos representan del total de ingresos por ventas de autos del concesionario.
- También se muestra el número de ventas por marca de auto y el porcentaje que este dato representa del total de ventas de autos del concesionario.
- Por último, se muestra la suma total de ingresos del concesionario por ventas de autos y el promedio de ingresos diarios en lo corrido del mes de mayo.

En caso de que el usuario desee consultar el estado de resultados simple del concesionario, se muestra lo siguiente:

- En esta información se puede ver el estado de resultados de la empresa, en la que se detalla el monto que ingresó a causa de las ventas/servicios que presta el concesionario, también se muestra el costo de ventas, y después el monto de la utilidad operativa.
- Posteriormente, se muestran los gastos operacionales y de ventas, rubro que incide directamente en la utilidad antes de impuestos de la empresa, monto que se muestra a continuación.
- Por último, se detalla el monto que la empresa debería pagar por concepto de impuesto de renta, el cual por ley representa el 33% de la utilidad antes de impuestos, y en última instancia la utilidad neta que obtuvo la empresa.

En caso de que el usuario desee consultar el estado de resultados detallado del concesionario, se muestra lo siguiente:

Se muestran las ventas totales generadas por el concesionario durante el período en consideración. Estas ventas comprenden los ingresos provenientes de la venta de autos, servicios del taller y venta de artículos del taller.

Posteriormente, se detalla el costo de ventas, que representa los gastos directamente asociados con la producción y venta de los productos y servicios ofrecidos por el concesionario.

La utilidad operativa se calcula restando los gastos operacionales y de ventas del total de ventas, lo que indica la ganancia obtenida exclusivamente de las operaciones comerciales.

En la sección de gastos operacionales y de ventas, se mencionan los diferentes rubros que inciden en la utilidad antes de impuestos del concesionario. Estos rubros incluyen comisiones pagadas a los vendedores, servicios como agua, electricidad, gas, telefonía, internet, seguridad y limpieza, así como gastos fijos como alquiler, mantenimiento, seguros, publicidad, marketing y costos financieros.

La utilidad antes de impuestos se calcula restando los gastos operacionales y de ventas del resultado operativo, lo que muestra la ganancia neta antes de considerar los impuestos.

El impuesto de renta se calcula aplicando un porcentaje (generalmente el 33%) a la utilidad antes de impuestos y representa el monto que la empresa debe pagar al gobierno por concepto de impuestos.

Finalmente, se muestra la utilidad neta, que es la ganancia final obtenida por el concesionario después de restar el impuesto de renta de la utilidad antes de impuestos.

Funcionalidad PERSONALIZAR AUTO

La funcionalidad "Personalizar Automóvil" te permite realizar modificaciones y mejoras en tu automóvil de manera personalizada. Con esta característica, podrás seleccionar entre diversas opciones de personalización, como cambios en la pintura, llantas, sistema de sonido, frenos o escape, según tus preferencias y necesidades.

Cuando accedes a la sección de personalización de automóviles, se te pedirá ingresar la cédula del propietario del vehículo. Una vez ingresada, el sistema buscará la información del propietario y del automóvil en una base de datos. Si la información es válida, podrás proceder con la personalización.

A continuación, se te mostrará una lista de mecánicos disponibles que atienden el tipo de automóvil que posees. Podrás seleccionar un mecánico de tu elección para realizar las modificaciones en tu vehículo.

Después de seleccionar un mecánico, se te mostrará una lista de productos disponibles para la personalización. Estos productos están relacionados con las distintas áreas de modificación que puedes elegir, como pintura, llantas, sonido,

frenos o escape. Podrás seleccionar el producto deseado para llevar a cabo la personalización. Una vez seleccionado el producto, se te presentará un resumen de la modificación que se realizará, incluyendo el tipo de modificación, el producto seleccionado y el mecánico asignado. Si estás de acuerdo, podrás confirmar la transacción y seleccionar una hora disponible para el servicio.

Finalmente, se mostrará el precio total del servicio, incluyendo la mano de obra del mecánico y el costo del producto seleccionado. Si aceptas, se creará una transacción de personalización de automóvil, y tanto el mecánico como el producto seleccionado se marcarán como ocupados en el sistema.

Esta funcionalidad te brinda la oportunidad de personalizar y mejorar tu automóvil según tus preferencias, contando con profesionales y productos especializados. ¡Disfruta de un vehículo a tu medida!

Funcionalidad Administrador

Importante: para ingresar a esta funcionalidad la cédula que debe de poner es 3355479.

Esta funcionalidad está diseñada para tener una mejor organización de los objetos y reutilizar el ciclo de ciertas características de algunas funcionalidades, es el apartado en donde el administrador tiene la capacidad de agregar artículos, vehículos, vendedores, mecánicos y a los mecánicos darle nuevamente su apartado de horarios para que la funcionalidad de taller funcione correctamente, solo los trabajadores con cargo de “admin” pueden utilizar esta funcionalidad, el sistema les pedirá su cédula para comprobar que sí se encuentren registrados como administradores.

CASOS DE PRUEBA.

- 333444 esta cédula puede usarse para la funcionalidad VENTA DE AUTOS
- 1234991492 esta cédula posee transacciones realizadas por tanto puede usarse para todas las funcionalidades.
- 3355479 RECUERDE que es el admin por tanto con esta cédula podrá ingresar a la funcionalidad Estadísticas y Administración
- Si desea añadir su usuario y crear una nueva transacción con él, lo puede hacer desde el número 6 en el menú inicial y así crear su usuario y hacer las transacciones que desee.

Aquí hay imágenes de el uso de algunas funcionalidades:

```
Bienvenido al portal de creación
Introduzca su Nombre y Apellido
Felipe Velez
Introduzca su Cedula
1234991492
Introduzca su Telefono
3232109673
Introduzca su Correo
felipe@gmail.com
Introduzca su Direccion
Medellin
Introduzca su Marca de interes
Toyota
Introduzca su presupuesto
9000000000
```

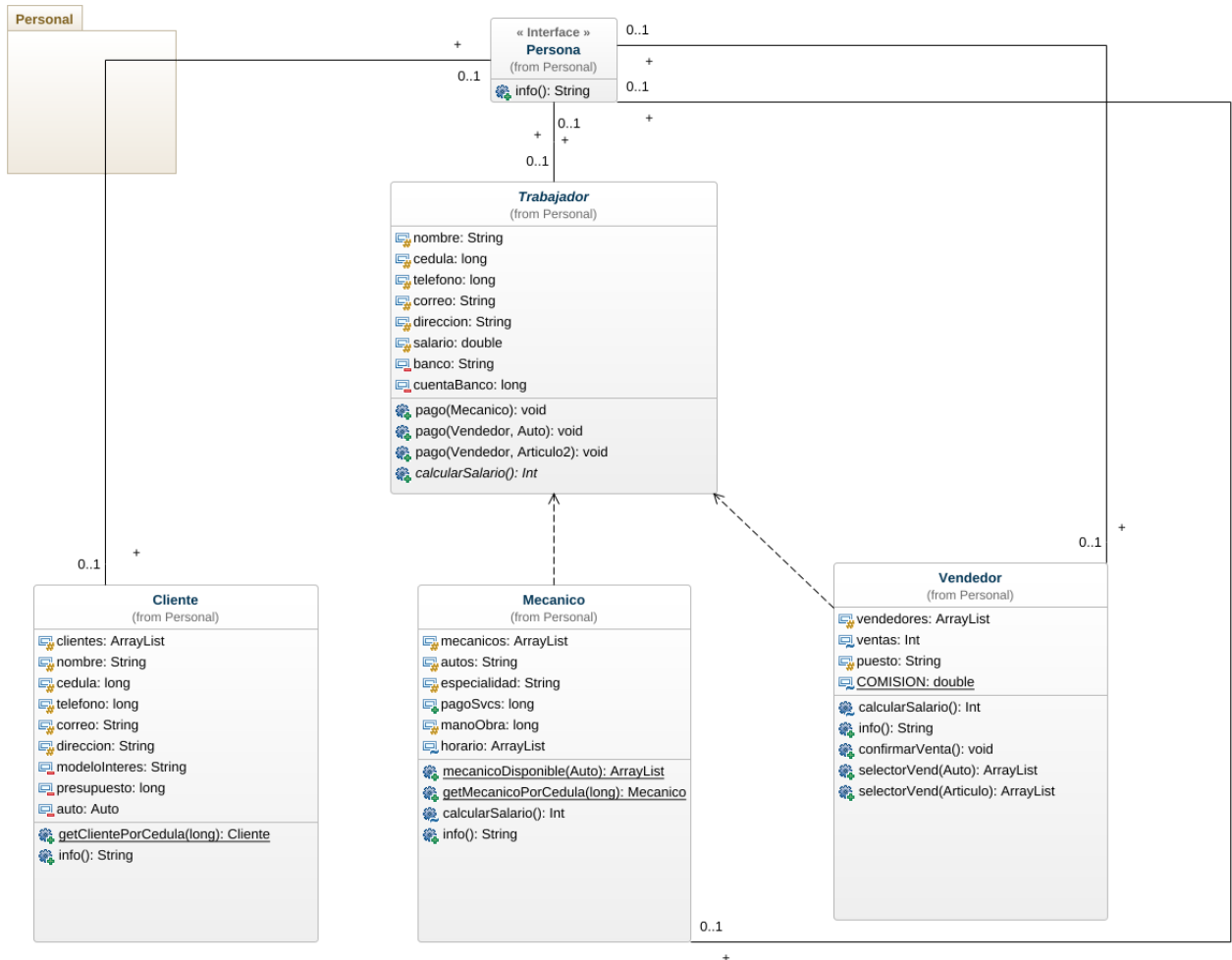
```
Transacción #00000818: venta realizada por María De la Paz para el cliente Felipe Velez por un total de $280000000 por el auto Prado
```

```
Transacción #00000976: taller realizado por Andres Castro para el cliente Felipe Velez por un total de $60000
La Factura le llegara a el correo: felipe@gmail.com
```

```
Transacción #00000592: venta realizada por Luis Suarez para el cliente Felipe Velez que adquirio el producto: Subwoofer de la marca: Kenwood , con p
recio total $100000
```

diagramas UML.

- PAQUETE gestoraplicacion.personal



- PAQUETE gestoraplicacion.activos

