//Trabajo Extra

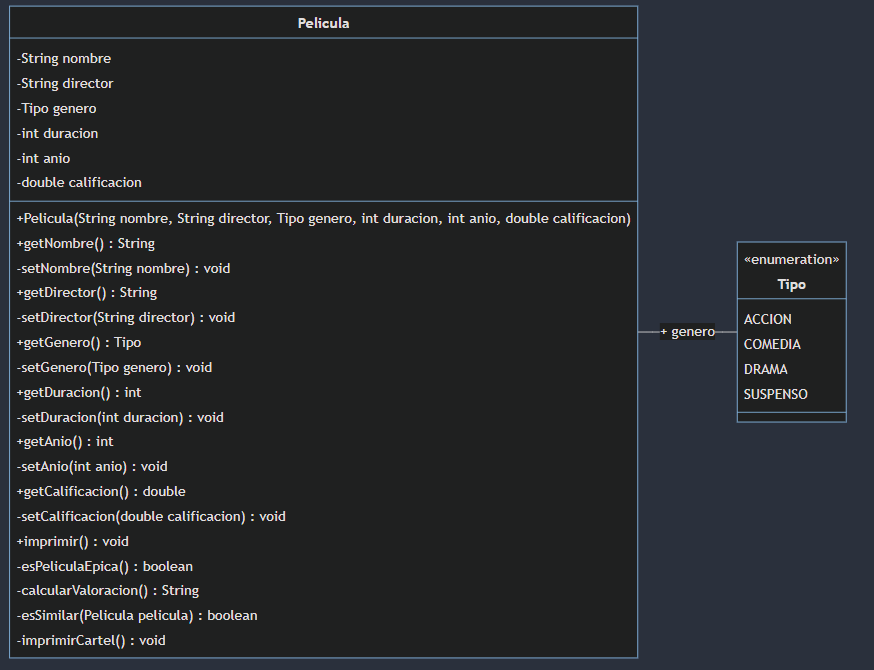
// Nombre: Alex Travez

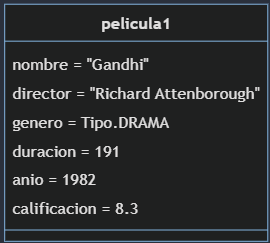
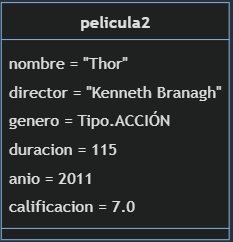
// Fecha de realizacion:30/01/2025

// Fecha de entrega: 31/01/2025

//Resultados:

PRIMER REQUERIMIENTO

 **Diagramas de Clase**

**Diagrama de Objetos:**

**Código :**

package paquete;

public class Pelicula {

private String nombre;

private String director;

private enum *Tipo* { ***ACCIÓN***, ***COMEDIA***, ***DRAMA***, ***SUSPENSO*** }

private *Tipo* genero;

private int duracion;

private int anio;

private double calificacion;

public Pelicula(String nombre, String director, *Tipo* genero, int duracion, int anio, double calificacion) {

this.nombre = nombre;

this.director = director;

this.genero = genero;

this.duracion = duracion;

this.anio = anio;

this.calificacion = calificacion;

}

public String getNombre() {

return nombre;

}

private void setNombre(String nombre) {

this.nombre = nombre;

}

public String getDirector() {

return director;

}

private void setDirector(String director) {

this.director = director;

}

public *Tipo* getGenero() {

return genero;

}

private void setGenero(*Tipo* genero) {

this.genero = genero;

}

public int getDuracion() {

return duracion;

}

private void setDuracion(int duracion) {

this.duracion = duracion;

}

public int getAnio() {

return anio;

}

private void setAnio(int anio) {

this.anio = anio;

}

public double getCalificacion() {

return calificacion;

}

private void setCalificacion(double calificacion) {

this.calificacion = calificacion;

}

public void imprimir() {

System.***out***.println("Nombre: " + nombre);

System.***out***.println("Director: " + director);

System.***out***.println("Género: " + genero);

System.***out***.println("Duración: " + duracion);

System.***out***.println("Año: " + anio);

System.***out***.println("Calificación: " + calificacion);

}

private boolean esPeliculaEpica() {

return duracion >= 180;

}

private String calcularValoracion() {

if (calificacion >= 0 && calificacion <= 2) {

return "Muy mala";

} else if (calificacion > 2 && calificacion <= 5) {

return "Mala";

} else if (calificacion > 5 && calificacion <= 7) {

return "Regular";

} else if (calificacion > 7 && calificacion <= 8) {

return "Buena";

} else if (calificacion > 8 && calificacion <= 10) {

return "Excelente";

} else {

return "No tiene asignada una valoración válida";

}

}

private boolean esSimilar(Pelicula pelicula) {

return pelicula.genero == genero && pelicula.calcularValoracion().equals(calcularValoracion());

}

// Método privado imprimirCartel

private void imprimirCartel() {

System.***out***.println("-------- " + nombre + " --------");

System.***out***.print("\t");

for (int i = 0; i < (int)calificacion; i++) {

System.***out***.print("\*");

}

System.***out***.println();

System.***out***.println("\t" + anio);

System.***out***.println("\t" + genero);

System.***out***.println(" " + director);

}

public static void main(String[] args) {

Pelicula pelicula1 = new Pelicula("Gandhi", "Richard Attenborough", *Tipo*.***DRAMA***, 191, 1982, 8.3);

Pelicula pelicula2 = new Pelicula("Thor", "Kenneth Branagh", *Tipo*.***ACCIÓN***, 115, 2011, 7.0);

pelicula1.imprimir();

System.***out***.println();

pelicula2.imprimir();

System.***out***.println();

System.***out***.println("La película " + pelicula1.getNombre() + " es épica: " + pelicula1.esPeliculaEpica());

System.***out***.println("La película " + pelicula2.getNombre() + " es épica: " + pelicula2.esPeliculaEpica());

System.***out***.println("La película " + pelicula1.getNombre() + " y la película " + pelicula2.getNombre() + " son similares: " + pelicula1.esSimilar(pelicula2));

System.***out***.println("\nCartel de la película 1:");

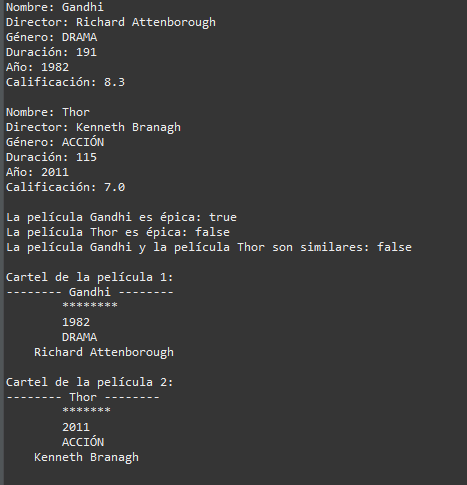
pelicula1.imprimirCartel();

System.***out***.println("\nCartel de la película 2:");

pelicula2.imprimirCartel();

}

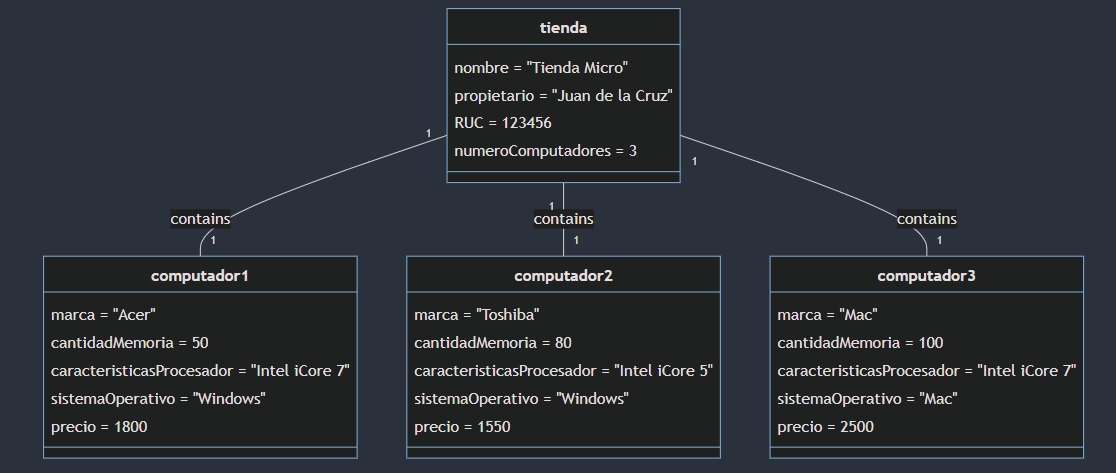
}

**Ejecución:**

SEGUNDO REQUERIMIENTO

 **Diagrama de clases :**

**Diagrama de Objetos:**



**Código:**

import java.util.Scanner;

public class Tienda {

String nombre, propietario;

int RUC;

Computador[] computadores;

static int *numeroComputadores*;

public Tienda(String nombre, String propietario, int RUC, int tamaño) {

if (tamaño < 1) {

System.***out***.println("Cantidad de computadores insuficiente.");

} else {

this.nombre = nombre;

this.propietario = propietario;

this.RUC = RUC;

computadores = new Computador[tamaño];

*numeroComputadores* = 0;

}

}

public boolean tiendaLlena() {

return *numeroComputadores* == computadores.length;

}

public boolean tiendaVacía() {

return *numeroComputadores* == 0;

}

public void añadir(Computador computador) {

if (tiendaLlena()) {

System.***out***.println("La tienda está llena. No se puede añadir elemento.");

} else {

computadores[*numeroComputadores*] = computador;

*numeroComputadores*++;

}

}

public boolean eliminar(String marcaComputador) {

int pos = buscar(marcaComputador);

if (pos < 0) {

return false;

}

for (int i = pos; i < *numeroComputadores* - 1; i++) {

computadores[i] = computadores[i + 1];

}

*numeroComputadores*--;

return true;

}

public int buscar(String marcaComputador) {

for (int i = 0; i < *numeroComputadores*; i++) {

if (computadores[i].marca.equals(marcaComputador)) {

return i;

}

}

return -1;

}

public void imprimir() {

for (int i = 0; i < *numeroComputadores*; i++) {

System.***out***.println("Computador " + i);

System.***out***.println("Marca = " + computadores[i].marca);

System.***out***.println("Cantidad de memoria = " + computadores[i].cantidadMemoria);

System.***out***.println("Características del procesador = " + computadores[i].caracteristicasProcesador);

System.***out***.println("Sistema operativo = " + computadores[i].sistemaOperativo);

System.***out***.println("Precio = " + computadores[i].precio);

}

}

}

class Computador {

String marca;

int cantidadMemoria;

String caracteristicasProcesador;

String sistemaOperativo;

double precio;

public Computador(String marca, int cantidadMemoria, String caracteristicasProcesador, String sistemaOperativo, double precio) {

this.marca = marca;

this.cantidadMemoria = cantidadMemoria;

this.caracteristicasProcesador = caracteristicasProcesador;

this.sistemaOperativo = sistemaOperativo;

this.precio = precio;

}

}

public class Main {

public static void main(String[] args) {

Tienda tienda = new Tienda("Tienda Micro", "Juan de la Cruz", 123456, 5);

Computador computador1 = new Computador("Acer", 50, "Intel iCore 7", "Windows", 1800);

Computador computador2 = new Computador("Toshiba", 80, "Intel iCore 5", "Windows", 1550);

Computador computador3 = new Computador("Mac", 100, "Intel iCore 7", "Mac", 2500);

tienda.añadir(computador1);

tienda.añadir(computador2);

tienda.añadir(computador3);

Scanner sc = new Scanner(System.***in***);

// Mostrar lista de marcas disponibles

System.***out***.println("Marcas disponibles:");

for (int i = 0; i < tienda.*numeroComputadores*; i++) {

System.***out***.println("- " + tienda.computadores[i].marca);

}

System.***out***.print("Ingrese la marca del computador a buscar: ");

String marca = sc.next();

int posicion = tienda.buscar(marca);

if (posicion != -1) {

System.***out***.println("El computador a buscar: " + marca + " se encuentra en la posición " + posicion);

} else {

System.***out***.println("El computador a buscar: " + marca + " no se encuentra en la tienda.");

}

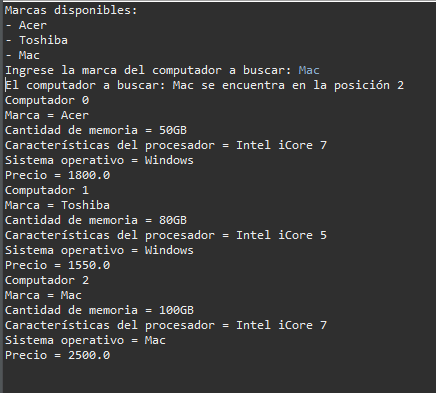
tienda.imprimir();

sc.close();

}

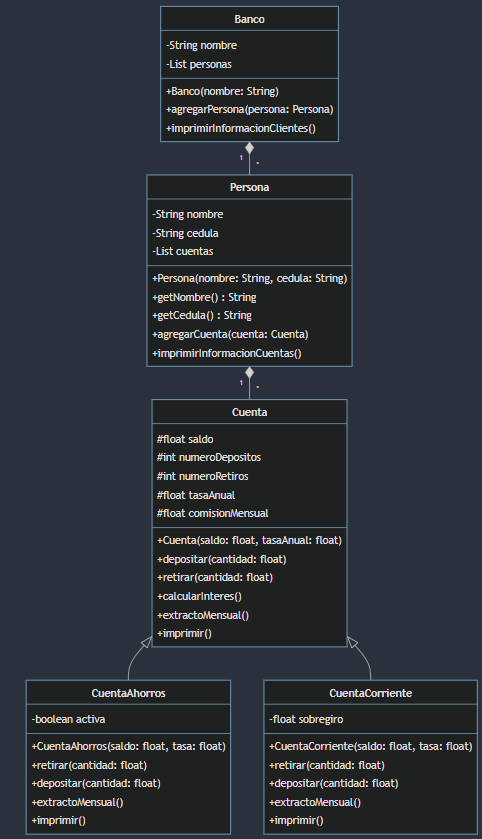
}

**Ejecución:**

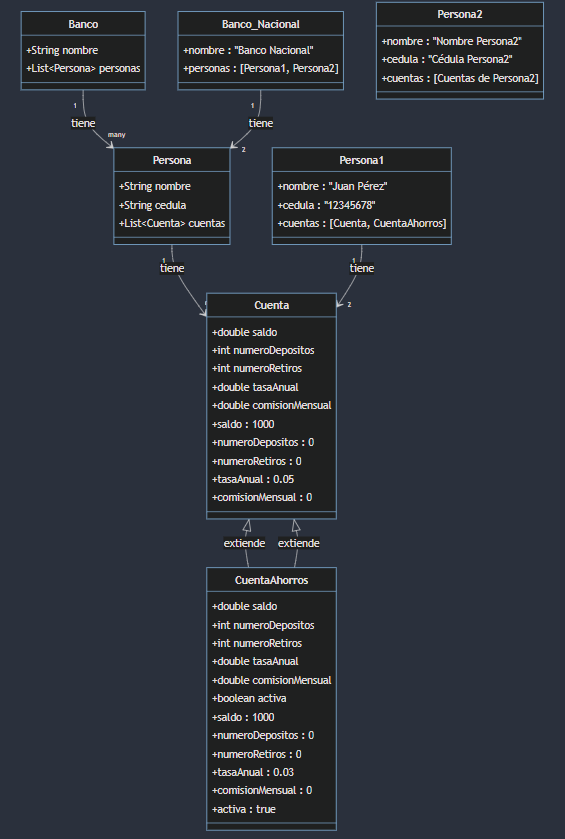


**TERCER REQUERIMIENTO**

**Diagramas de Clase:**



**Diagramas de Objetos:**



**Código:**

package paquete;

import java.util.ArrayList;

import java.util.List;

public class Banco {

private String nombre;

private List<Persona> personas;

public Banco(String nombre) {

this.nombre = nombre;

this.personas = new ArrayList<>();

}

public void agregarPersona(Persona persona) {

personas.add(persona);

}

public void imprimirInformacionClientes() {

System.***out***.println("Información de clientes del banco: " + nombre);

for (Persona persona : personas) {

System.***out***.println("Nombre: " + persona.getNombre() + ", Cédula: " + persona.getCedula());

persona.imprimirInformacionCuentas();

}

}

}

//Clases anteriores

public class Cuenta {

protected float saldo;

protected int numeroDepositos = 0;

protected int numeroRetiros = 0;

protected float tasaAnual;

protected float comisionMensual = 0;

public Cuenta(float saldo, float tasaAnual) {

this.saldo = saldo;

this.tasaAnual = tasaAnual;

}

public void depositar(float cantidad) {

saldo += cantidad;

numeroDepositos++;

}

public void retirar(float cantidad) {

float nuevoSaldo = saldo - cantidad;

if (nuevoSaldo >= 0) {

saldo -= cantidad;

numeroRetiros++;

} else {

System.***out***.println("La cantidad a retirar excede el saldo actual.");

}

}

public void calcularInteres() {

float tasaMensual = tasaAnual / 12;

float interesMensual = saldo \* tasaMensual;

saldo += interesMensual;

}

public void extractoMensual() {

saldo -= comisionMensual;

calcularInteres();

}

public void imprimir() {

System.***out***.println("Saldo = $" + saldo);

System.***out***.println("Tasa = $" + tasaAnual);

System.***out***.println("Comisión mensual = $" + comisionMensual);

System.***out***.println("Número de transacciones = " + (numeroDepositos + numeroRetiros));

}

}

package paquete;

public class CuentaAhorros extends Cuenta {

private boolean activa;

public CuentaAhorros(float saldo, float tasa) {

super(saldo, tasa);

activa = saldo >= 500;

}

*@Override*

public void retirar(float cantidad) {

if (activa) {

super.retirar(cantidad);

}

}

*@Override*

public void depositar(float cantidad) {

if (activa) {

super.depositar(cantidad);

}

}

*@Override*

public void extractoMensual() {

if (numeroRetiros > 4) {

comisionMensual += (numeroRetiros - 4) \* 10;

}

super.extractoMensual();

activa = saldo >= 500;

}

*@Override*

public void imprimir() {

super.imprimir();

System.***out***.println(activa ? "Cuenta activa" : "Cuenta no activa");

}

}

public class CuentaCorriente extends Cuenta {

private float sobregiro;

public CuentaCorriente(float saldo, float tasa) {

super(saldo, tasa);

sobregiro = 0;

}

*@Override*

public void retirar(float cantidad) {

float resultado = saldo - cantidad;

if (resultado < 0) {

sobregiro -= resultado;

saldo = 0;

} else {

super.retirar(cantidad);

}

}

*@Override*

public void depositar(float cantidad) {

if (sobregiro > 0) {

float residuo = sobregiro - cantidad;

if (residuo > 0) {

sobregiro = 0;

saldo = residuo;

} else {

sobregiro = -residuo;

saldo = 0;

}

} else {

super.depositar(cantidad);

}

}

*@Override*

public void extractoMensual() {

super.extractoMensual();

}

*@Override*

public void imprimir() {

super.imprimir();

System.***out***.println("Valor de sobregiro = $" + sobregiro);

}

}

public class Persona {

private String nombre;

private String cedula;

private List<Cuenta> cuentas;

public Persona(String nombre, String cedula) {

this.nombre = nombre;

this.cedula = cedula;

this.cuentas = new ArrayList<>();

}

import java.util.ArrayList;

public class Persona {

private String nombre;

private String cedula;

private List<Cuenta> cuentas;

public Persona(String nombre, String cedula) {

this.nombre = nombre;

this.cedula = cedula;

this.cuentas = new ArrayList<>();

}

// Métodos getter para los atributos privados

public String getNombre() {

return nombre;

}

public String getCedula() {

return cedula;

}

// Métodos para gestionar cuentas

public void agregarCuenta(Cuenta cuenta) {

cuentas.add(cuenta);

}

public void imprimirInformacionCuentas() {

System.***out***.println("Información de cuentas para la persona: " + nombre);

for (Cuenta cuenta : cuentas) {

cuenta.imprimir();

System.***out***.println();

}

}

}

package paquete;

//Clase principal

public class Main {

public static void main(String[] args) {

// Crear instancias de cuenta

Cuenta cuenta1 = new Cuenta(1000, 0.05f);

CuentaAhorros cuentaAhorros = new CuentaAhorros(1000, 0.03f);

CuentaCorriente cuentaCorriente = new CuentaCorriente(500, 0.02f);

// Crear personas

Persona persona1 = new Persona("Juan Pérez", "12345678");

Persona persona2 = new Persona("Ana Gómez", "87654321");

// Agregar cuentas a las personas

persona1.agregarCuenta(cuenta1);

persona1.agregarCuenta(cuentaAhorros);

persona2.agregarCuenta(cuentaCorriente);

// Crear banco

Banco banco = new Banco("Banco Nacional");

// Agregar personas al banco

banco.agregarPersona(persona1);

banco.agregarPersona(persona2);

// Imprimir información de clientes del banco

banco.imprimirInformacionClientes();

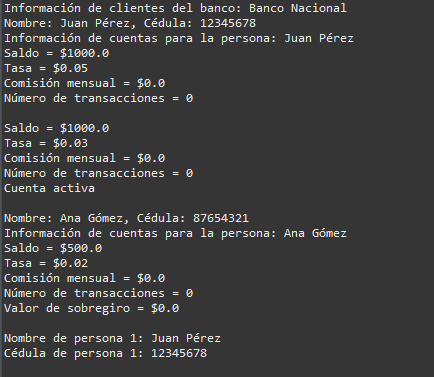
// Ejemplo de acceso usando getters

System.***out***.println("Nombre de persona 1: " + persona1.getNombre());

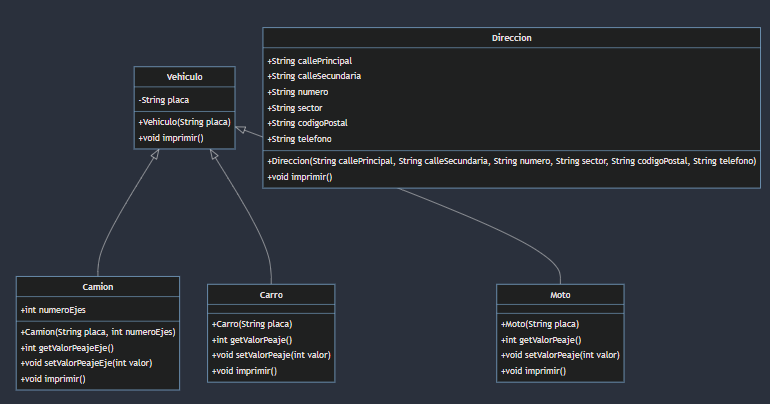
System.***out***.println("Cédula de persona 1: " + persona1.getCedula());

}

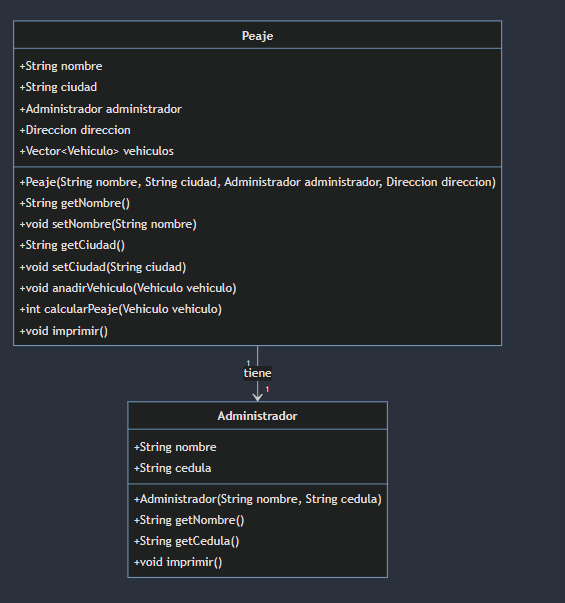
}

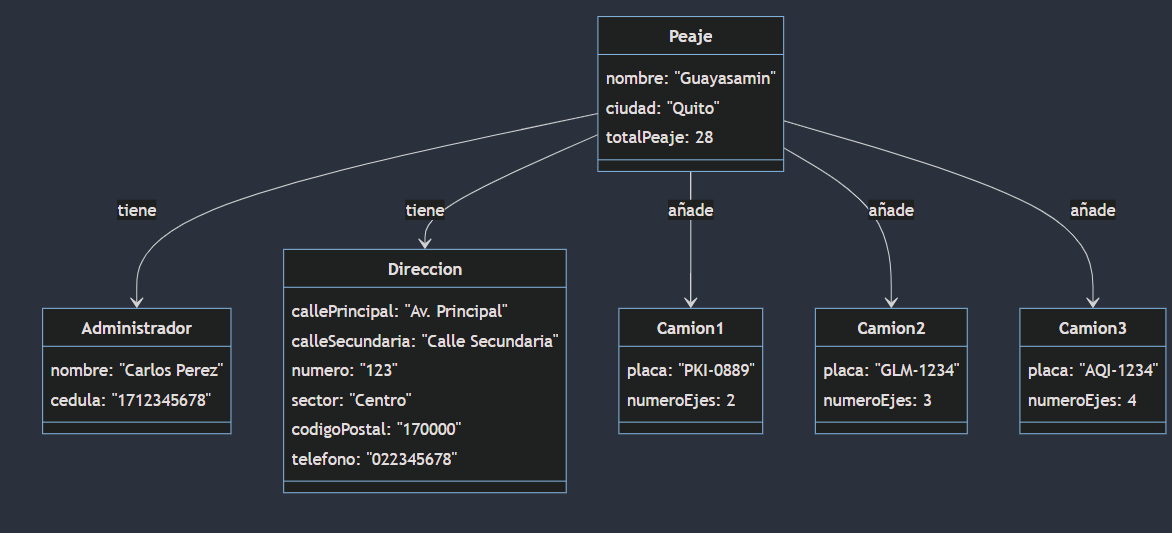
**Ejecución:**

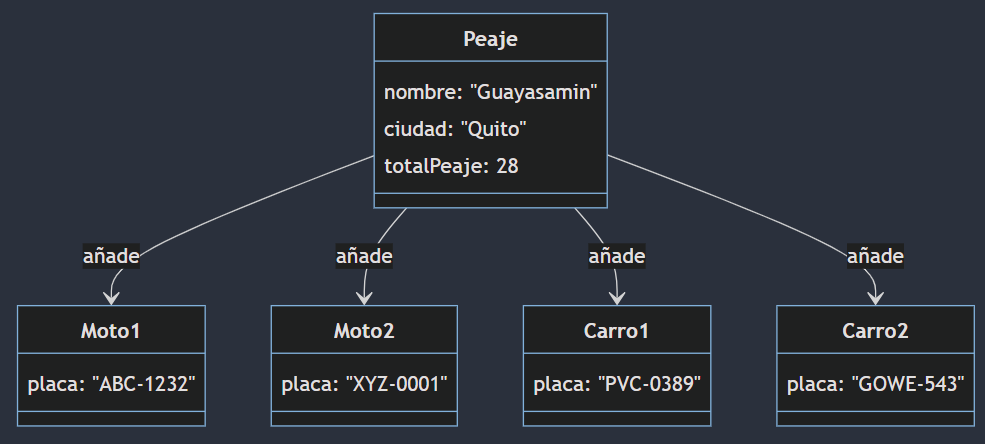
CUARTO REQUERIMIENTO

**Diagramas de Clase:**

**Continuación:**



**Diagramas de Objetos:**

**Continuación:**

**Código:**

package paquete;

class Administrador {

private String nombre;

private String cedula;

Administrador(String nombre, String cedula) {

this.nombre = nombre;

this.cedula = cedula;

}

public String getNombre() {

return nombre;

}

public String getCedula() {

return cedula;

}

void imprimir() {

System.***out***.println("Administrador: " + nombre + ", Cedula: " + cedula);

}

}

public class Camion extends Vehiculo {

static int *valorPeajeEje* = 2;

int numeroEjes;

Camion(String placa, int numeroEjes) {

super(placa);

this.numeroEjes = numeroEjes;

}

public int getValorPeajeEje() {

return *valorPeajeEje*;

}

static void setValorPeajeEje(int valor) {

*valorPeajeEje* = valor;

}

*@Override*

void imprimir() {

super.imprimir();

System.***out***.println("Valor del peaje = " + numeroEjes \* *valorPeajeEje*);

}

}

public class Carro extends Vehiculo {

static int *valorPeaje* = 5;

Carro(String placa) {

super(placa);

}

public int getValorPeaje() {

return *valorPeaje*;

}

static void setValorPeaje(int valor) {

*valorPeaje* = valor;

}

*@Override*

void imprimir() {

super.imprimir();

System.***out***.println("Valor del peaje = " + *valorPeaje*);

}

}

class Direccion {

private String callePrincipal;

private String calleSecundaria;

private String numero;

private String sector;

private String codigoPostal;

private String telefono;

Direccion(String callePrincipal, String calleSecundaria, String numero, String sector, String codigoPostal, String telefono) {

this.callePrincipal = callePrincipal;

this.calleSecundaria = calleSecundaria;

this.numero = numero;

this.sector = sector;

this.codigoPostal = codigoPostal;

this.telefono = telefono;

}

void imprimir() {

System.***out***.println("Direccion: " + callePrincipal + " y " + calleSecundaria + ", \nNumero: " + numero + ", \nSector: " + sector + ", \nCodigo Postal: " + codigoPostal + ", \nTelefono: " + telefono);

}

}

public class Moto extends Vehiculo {

static int *valorPeaje* = 2;

Moto(String placa) {

super(placa);

}

public int getValorPeaje() {

return *valorPeaje*;

}

static void setValorPeaje(int valor) {

*valorPeaje* = valor;

}

*@Override*

void imprimir() {

super.imprimir();

System.***out***.println("Valor del peaje = " + *valorPeaje*);

}

}

import java.io.Serializable;

import java.util.Vector;

public class Peaje implements Serializable {

private String nombre;

private String ciudad;

private Vector<Vehiculo> vehiculos;

private Administrador administrador;

private Direccion direccion;

private int totalPeaje = 0;

static int *totalCamiones* = 0;

static int *totalMotos* = 0;

static int *totalCarros* = 0;

Peaje(String nombre, String ciudad, Administrador administrador, Direccion direccion) {

this.nombre = nombre;

this.ciudad = ciudad;

this.administrador = administrador;

this.direccion = direccion;

vehiculos = new Vector<>(); // Crea el vector de vehículos

}

public String getNombre() {

return nombre;

}

private void setNombre(String nombre) {

this.nombre = nombre;

}

public String getCiudad() {

return ciudad;

}

private void setCiudad(String ciudad) {

this.ciudad = ciudad;

}

public void anadirVehiculo(Vehiculo vehiculo) {

vehiculos.add(vehiculo);

}

public int calcularPeaje(Vehiculo vehiculo) {

if (vehiculo instanceof Carro) {

*totalCarros*++;

totalPeaje += Carro.*valorPeaje*;

return Carro.*valorPeaje*;

} else if (vehiculo instanceof Moto) {

*totalMotos*++;

totalPeaje += Moto.*valorPeaje*;

return Moto.*valorPeaje*;

} else if (vehiculo instanceof Camion) {

*totalCamiones*++;

Camion camion = (Camion) vehiculo;

totalPeaje += camion.numeroEjes \* camion.*valorPeajeEje*;

return camion.numeroEjes \* camion.*valorPeajeEje*;

} else {

return -1;

}

}

public void imprimir() {

System.***out***.println("Peaje = " + getNombre());

System.***out***.println("Ubicacion = " + getCiudad());

System.***out***.println("Total de carros = " + *totalCarros*);

System.***out***.println("Total de motos = " + *totalMotos*);

System.***out***.println("Total de camiones = " + *totalCamiones*);

int totalVehiculos = *totalCarros* + *totalMotos* + *totalCamiones*;

System.***out***.println("Total de vehiculos = " + totalVehiculos);

System.***out***.println("Dinero recaudado = $" + totalPeaje);

direccion.imprimir();

administrador.imprimir();

}

}

import java.io.Serializable;

import java.util.Vector;

abstract public class Vehiculo {

private String placa;

Vehiculo(String placa) {

this.placa = placa;

}

void imprimir() {

System.***out***.println("Placa = " + placa);

}

}

public class Main {

public static void main(String[] args) {

Administrador admin = new Administrador("Carlos Perez", "1712345678");

Direccion direccion = new Direccion("Av. Principal", "Calle Secundaria", "123", "Centro", "170000", "022345678");

Peaje peaje = new Peaje("Guayasamin", "Quito", admin, direccion);

Camion camion1 = new Camion("PKI-0889", 2);

peaje.anadirVehiculo(camion1);

peaje.calcularPeaje(camion1);

Camion camion2 = new Camion("GLM-1234", 3);

peaje.anadirVehiculo(camion2);

peaje.calcularPeaje(camion2);

Camion camion3 = new Camion("AQI-1234", 4);

peaje.anadirVehiculo(camion3);

peaje.calcularPeaje(camion3);

Moto moto1 = new Moto("ABC-1232");

peaje.anadirVehiculo(moto1);

peaje.calcularPeaje(moto1);

Moto moto2 = new Moto("XYZ-0001");

peaje.anadirVehiculo(moto2);

peaje.calcularPeaje(moto2);

Carro carro1 = new Carro("PVC-0389");

peaje.anadirVehiculo(carro1);

peaje.calcularPeaje(carro1);

Carro carro2 = new Carro("GOWE-543");

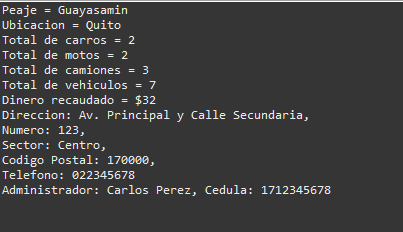
peaje.anadirVehiculo(carro2);

peaje.calcularPeaje(carro2);

peaje.imprimir();

}

}

**Ejecución:**