



# Universidad Distrital Francisco José de caldas

## Tecnología en Sistematización de Datos

### Vehículo



### ArrayList

### Polimorfismo

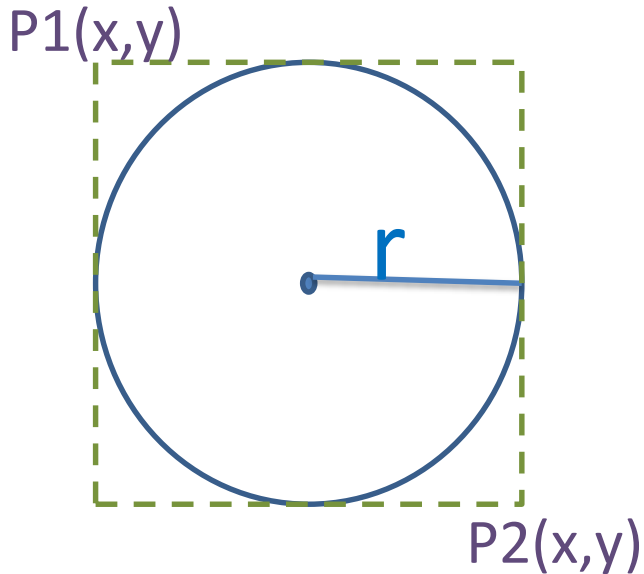
Programación Multinivel  
Sonia Alexandra Pinzón Nuñez



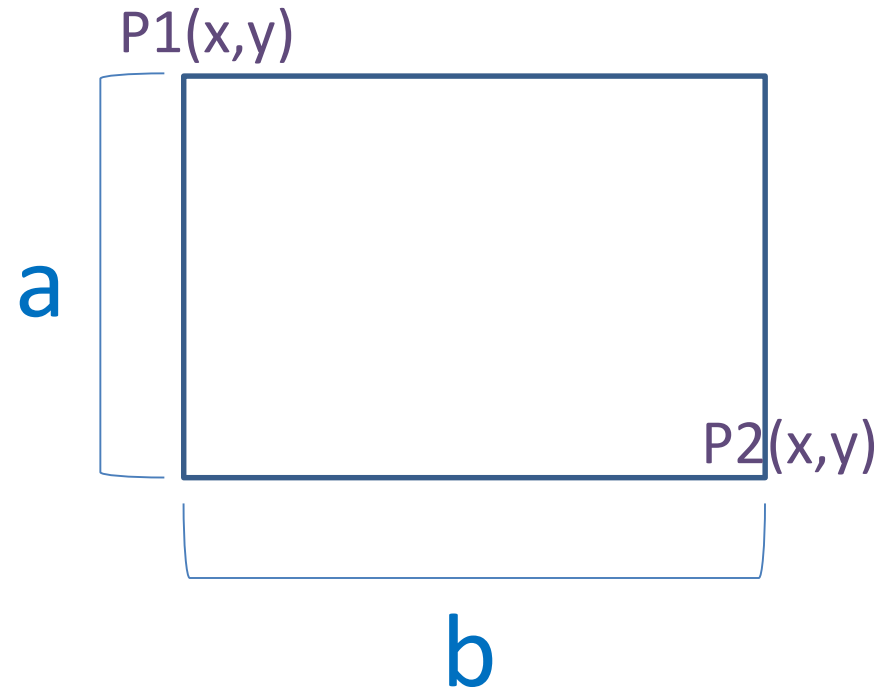
# Polimorfismo- Concepto

Hallar el área de un círculo y un rectángulo

$$\text{área} = \pi r^2$$



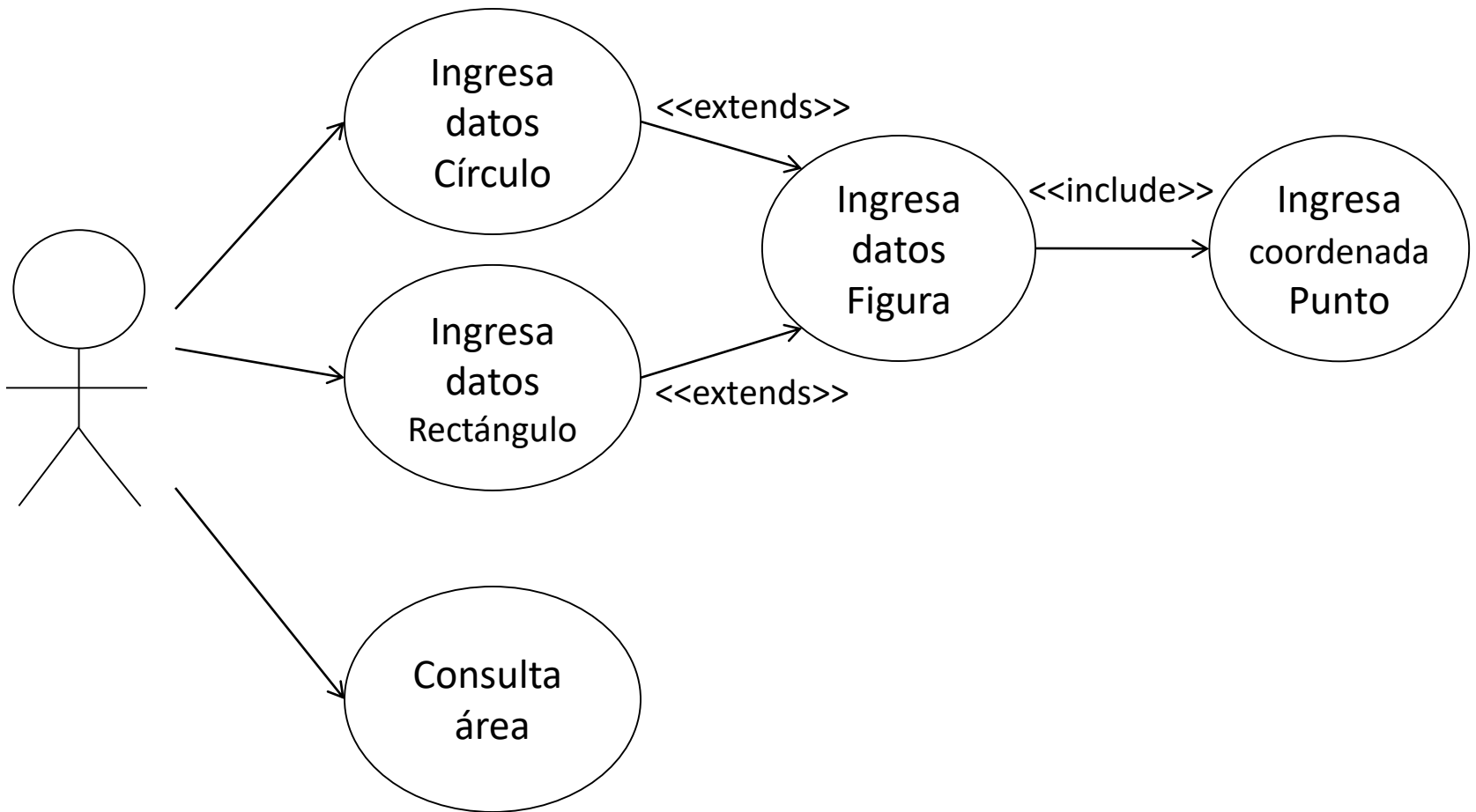
$$\text{área} = b * a$$





# Polimorfismo - Concepto

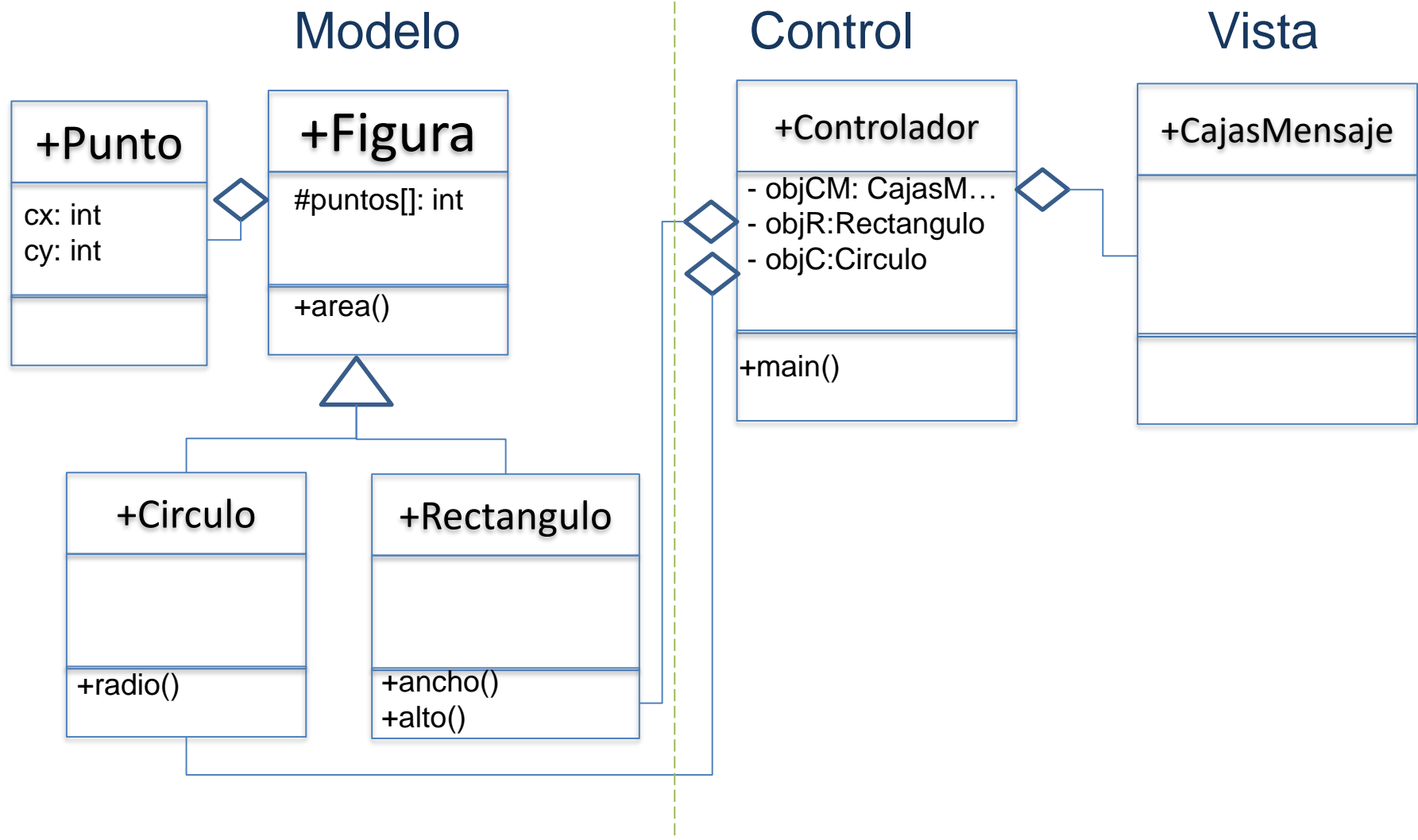
## Diagrama de Caso de Uso





# Polimorfismo - Concepto

## Diagrama de Clases

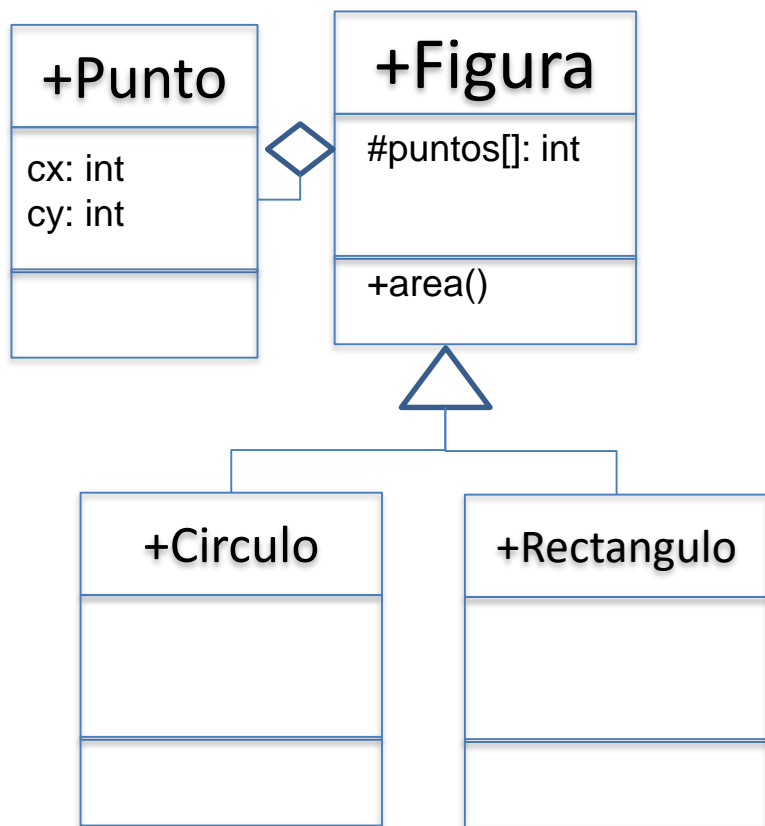




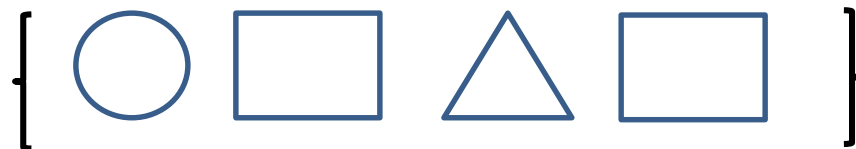
# Polimorfismo: Concepto

## Polimorfismo

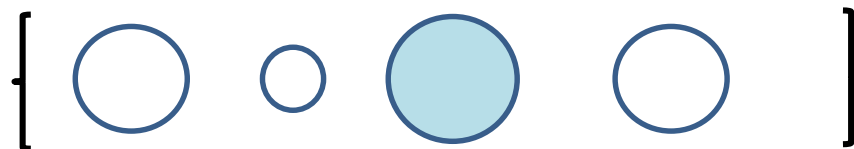
- Forma en que las clases derivadas implementan o sobrescriben un comportamiento (método) de una clase padre (super).
- Transformar un objeto en otro.



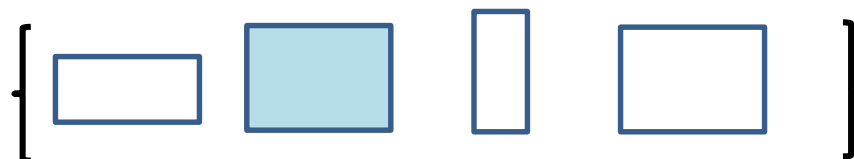
## Figuras



## Circulos



## Rectángulos





# Polimorfismo - Colecciones

Permite agrupar conjuntos de objetos con comportamientos o atributos comunes.

- A Todas las Figuras se les calcula el área .
- Una lista conjunta de elementos de un super-tipo (Figura) evita crear una lista por cada clase derivada, para funcionalidades comunes por ejemplo listar los puntos de cada Figura, calcular totales, etc.

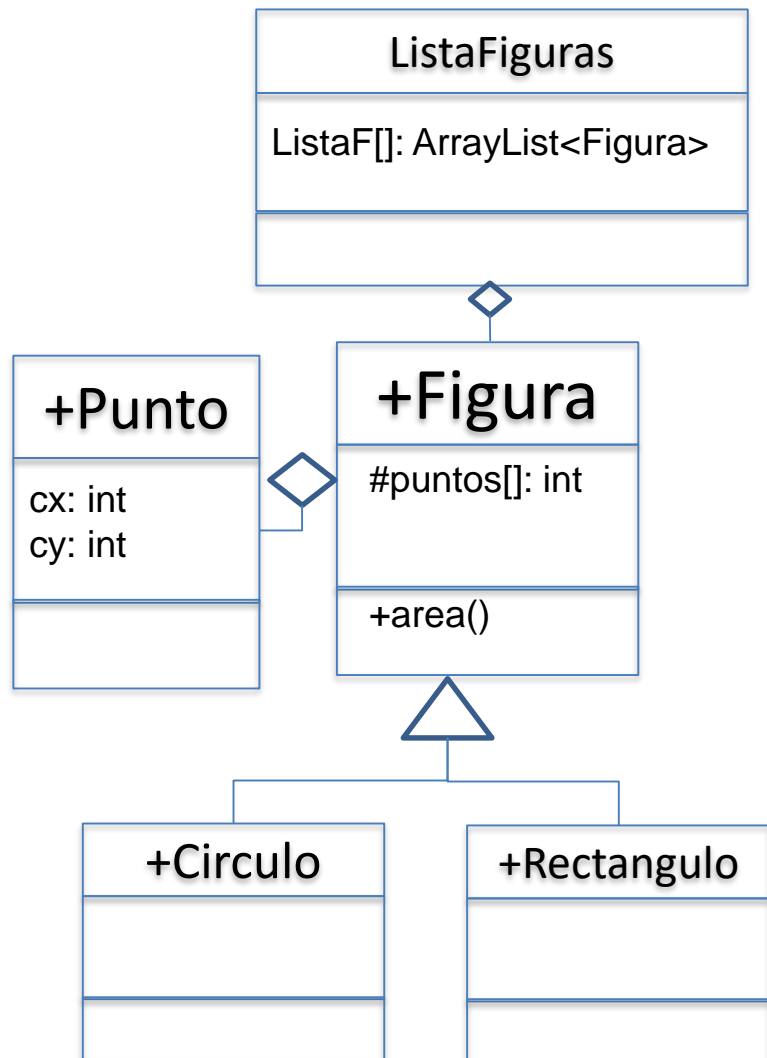
## Lista Figuras



```
public class ListaFiguras    {  
    ArrayList <Figura> listaF;  
  
    public ListaFiguras()    {  
        this.listaF = new <Figura> ArrayList();  
        listaF.add(new Circulo());  
        listaF.add(new Rectangulo());  
        listaF.add(new Rectangulo());  
        listaF.add(new Circulo());  
    }  
}
```



# Polimorfismo: Colecciones



## Colección Figuras



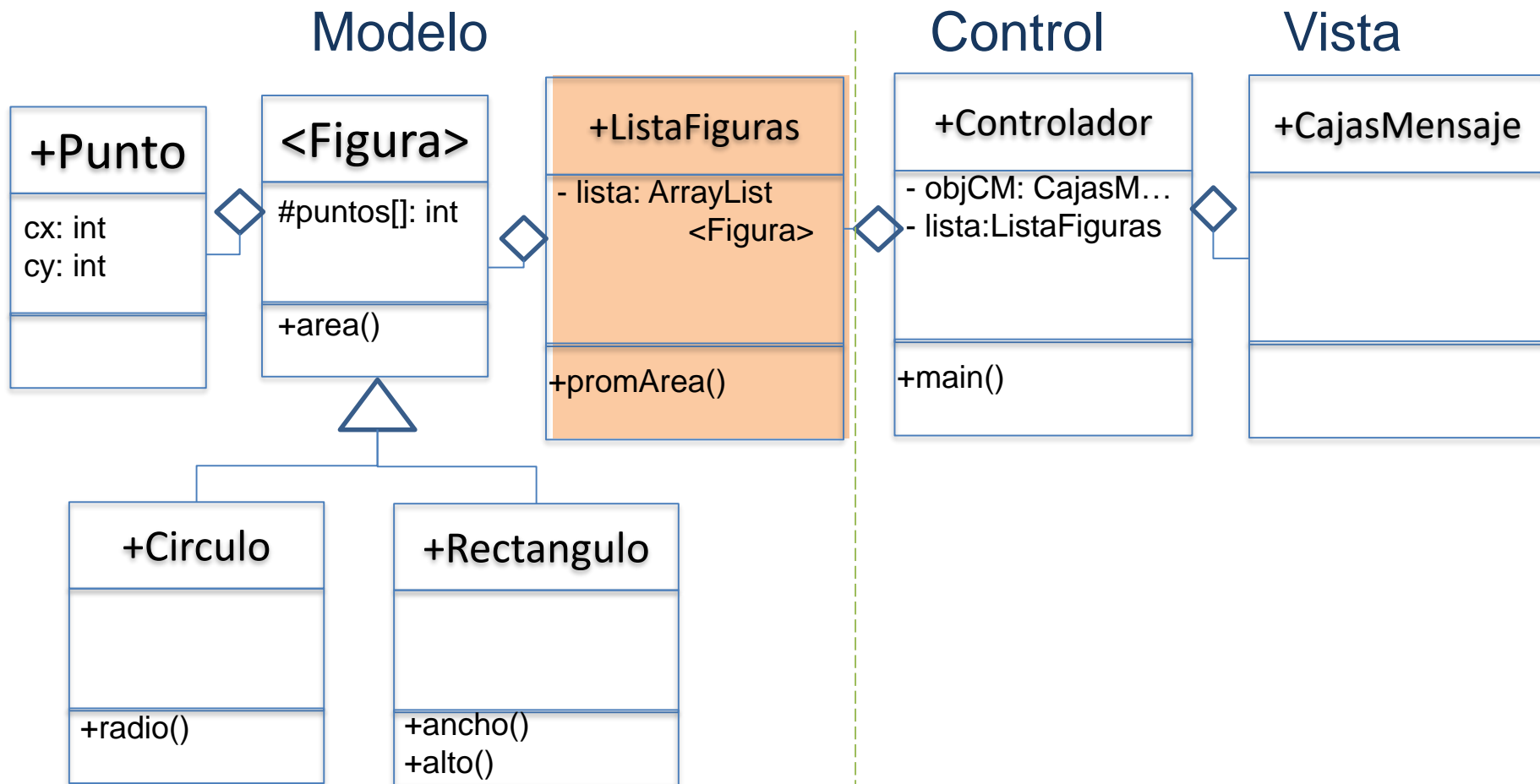
```
public class ListaFiguras {
    ArrayList <Figura> listaF;

    public ListaFiguras() {
        this.listaF = new <Figura> ArrayList();
        listaF.add(new Circulo());
        listaF.add(new Rectangulo());
        listaF.add(new Rectangulo());
        listaF.add(new Circulo());
    }
}
```



# Polimorfismo

## Diagrama de Clases







# Clase Figura

```
public abstract class Figura {  
    protected ArrayList <Punto> puntos;  
  
    public Figura(ArrayList<Punto> puntos) {  
        this.puntos = puntos;  
    }  
    public Figura() {  
        this.puntos = new <Punto> ArrayList ();  
    }  
  
    public ArrayList <Punto> getPuntos() {  
        return puntos;  
    }  
  
    public void setPuntos(ArrayList <Punto> puntos) {  
        this.puntos = puntos;  
    }  
  
    @Override  
    public String toString() {  
        String pts="";  
        for(int i=0; i< puntos.size();i++)  
            pts+=puntos.get(i).toString()+"\n";  
        return "puntos=" + pts;  
    }  
    public abstract double area();  
}
```



# Clase Círculo

```
public class Circulo extends Figura {  
    public Circulo(ArrayList<Punto> puntos) {  
        super(puntos);  
    }  
    public Circulo() {  
        super();  
    }  
    public double radio() {  
        return Math.abs(  
            this.getPuntos().get(1).getCx() - this.getPuntos().get(0).getCx()  
        )/2;  
    }  
    @Override  
    public double area() {  
        return Math.PI* Math.pow(radio(), 2);  
    }  
}
```



# Clase ListaFiguras

```
public class ListaFiguras {  
    ArrayList <Figura> Lista;  
    public ListaFiguras(ArrayList<Figura> Lista) {  
        this.Lista = Lista;  
    }  
    public ListaFiguras() {  
        this.Lista = new ArrayList <Figura>();  
    }  
    public double sumaA() {  
        double figs=0;  
        for(int i=0; i< Lista.size();i++)  
            figs+=Lista.get(i).area();  
        return figs;  
    }  
    @Override  
    public String toString() {  
        String fig,figs;  
        figs="\n";  
        for(int i=0; i< Lista.size();i++){  
            if(Lista.get(i) instanceof Circulo)  
                fig="Circulo";  
            else  
                fig="Rectángulo";  
            figs+="Figura " + (i+1)+" Tipo: "+fig+"\n"+ Lista.get(i).toString()+"\n";  
        }  
        return "Lista de Figuras: " + figs;  
    }  
    public ArrayList<Figura> getLista() { ...3 lines }  
    public void setLista(ArrayList<Figura> Lista) { ...3 lines }  
}
```

Sumatoria de Áreas

Si es una instancia de tipo  
rectángulo



# Clase ControladorListaF

```
public class ControladorListaF {  
    ListaFiguras lista;  
    CajasMensaje objCM;  
    public ControladorListaF() {  
        this.lista = new ListaFiguras();  
        this.objCM = new CajasMensaje();  
    }  
    public void iniciar() {  
        do {  
            objCM.setTitulo("Ejemplo Polimorfismo listas");  
            //primera forma para agregar un punto al arreglo  
            int opc=objCM.leerEntero("FIGURAS \n 1. Circulo \n 2.Rectángulo "+  
                "\n Ingrese número de figura a registrar...");  
            Figura objF = null;  
            switch(opc) {  
                case 1:  
                    objF= new Circulo(); //polimorfismo Figura se transforma en círculo  
                    for (int i = 0; i < 2; i++) {  
                        objF.getPuntos().add(new Punto(  
                            objCM.leerEntero("Coordenada X punto "+(i+1)),  
                            objCM.leerEntero("Coordenada Y punto "+(i+1))));  
                    }  
                    objCM.mostrar("Datos Circulo "+ objF.toString()+  
                        "\n área: "+objF.area());  
                    break;  
                case 2:  
                    objF= new Rectangulo(); //polimorfismo Figura se transforma en Rectángulo  
                    //código registro rectángulo  
                    break;  
            }  
            lista.getList().add(objF); //Polimorfismo  
        } while (objCM.confirmar("Desea registrar otra Figura?"));  
        objCM.mostrar("Listado de figuras"+lista.toString()+  
            "\n Sumatoria de áreas "+lista.sumaA());  
    }  
}
```



# Polimorfismo - Ejercicio

Registrar vehículos(Motos, Autos), calcular impuesto y obtener el recaudo total de vehículos registrados.

Criterio	tarifa
Cilindraje >125	1,5

Criterio	tarifa
Modelo< 2000	1,5
Otros	2,5



## Vehiculo

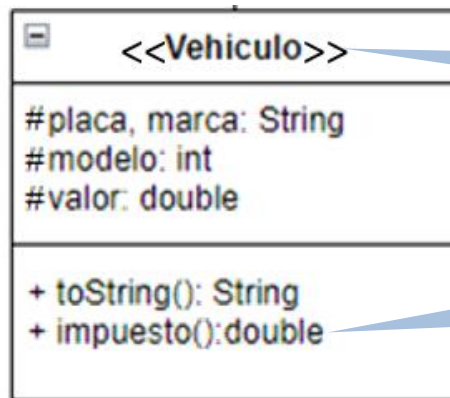
```
#placa,marca: String  
# modelo,cilind:int  
# valor:double
```

```
+impuesto():double
```

El concepto es el mismo, cada clase lo implementa, según sus necesidades. (Cambia de forma.)



# Polimorfismo: Colecciones



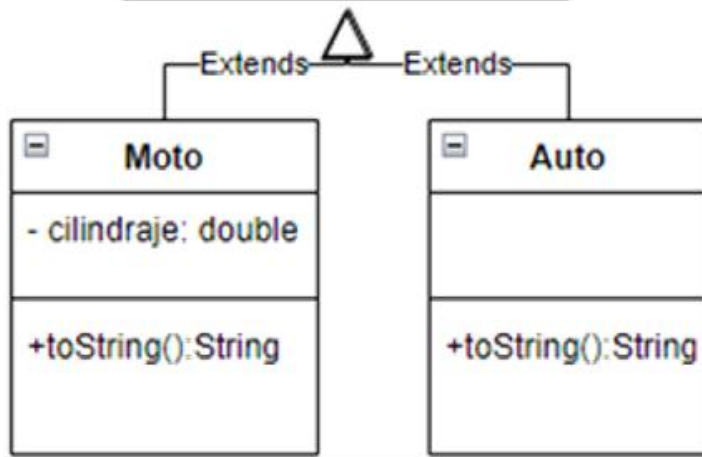
Clase Abstracta

?  
Vehículo

Método Abstracto



```
@Override
public double impuesto() {
    if(cilindraje>=125)
        return this.valor*0.05;
    else
        return 0;
}
```

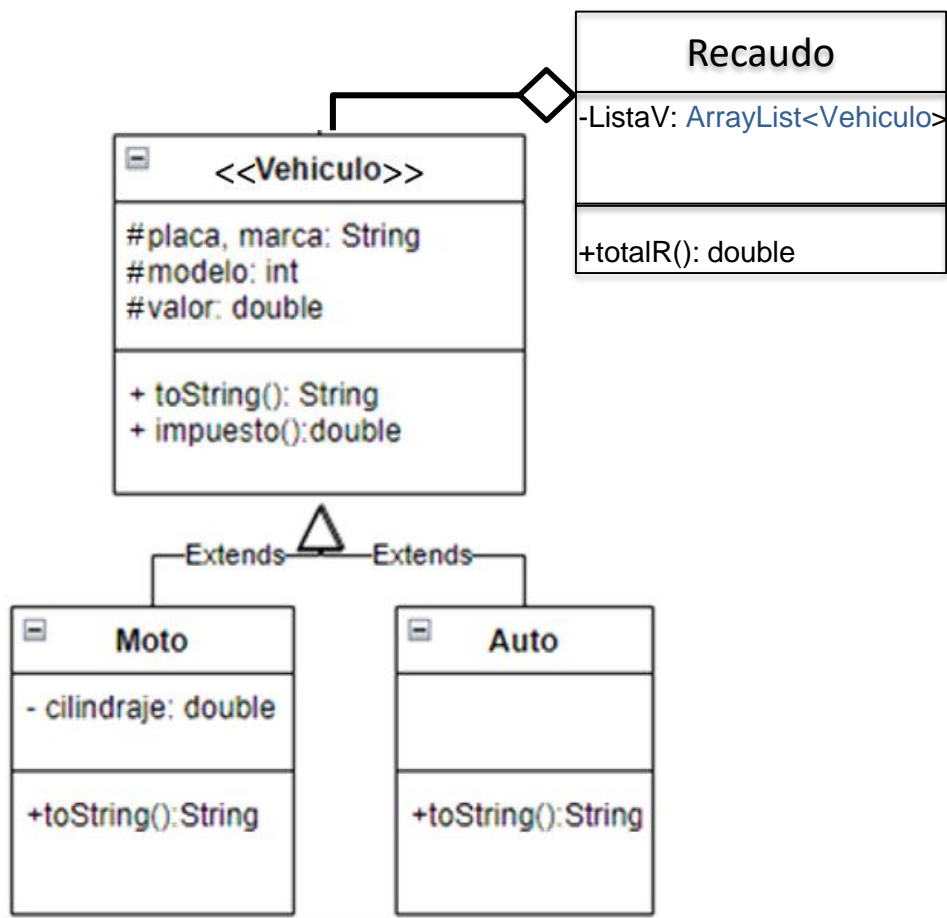


```
@Override
public double impuesto() {
    if(modelo>1990)
        return this.valor*0.05;
    else
        return this.valor*0.10;
}
```

Forma 1 Polimorfismo  
Sobrescribir el método  
abstracto



# Polimorfismo: Colecciones



## Array Vehículo



## Motos



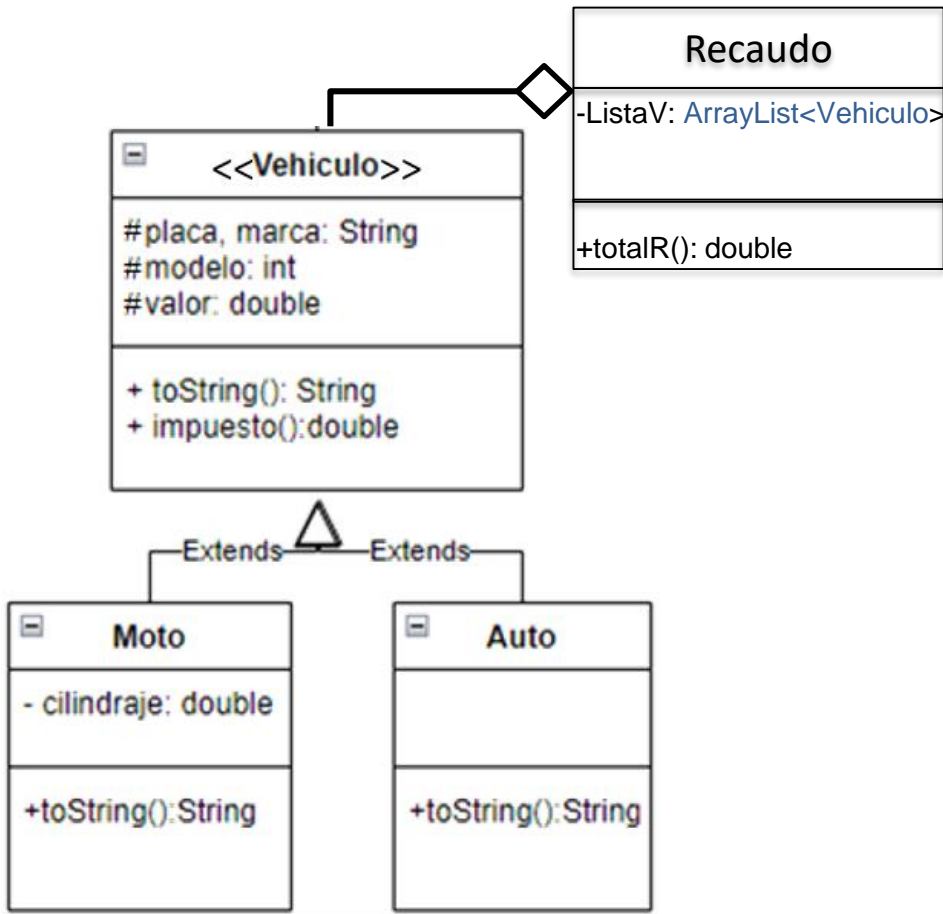
## Autos







# Polimorfismo: Colecciones



## Array Vehículos



```
public class Recaudo {
    private ArrayList<Vehiculo> listaV;

    public Recaudo() {
        this.listaV = new ArrayList<Vehiculo>();
        this.listaV.add(new Auto());
        this.listaV.add(new Moto());
    }
}
```

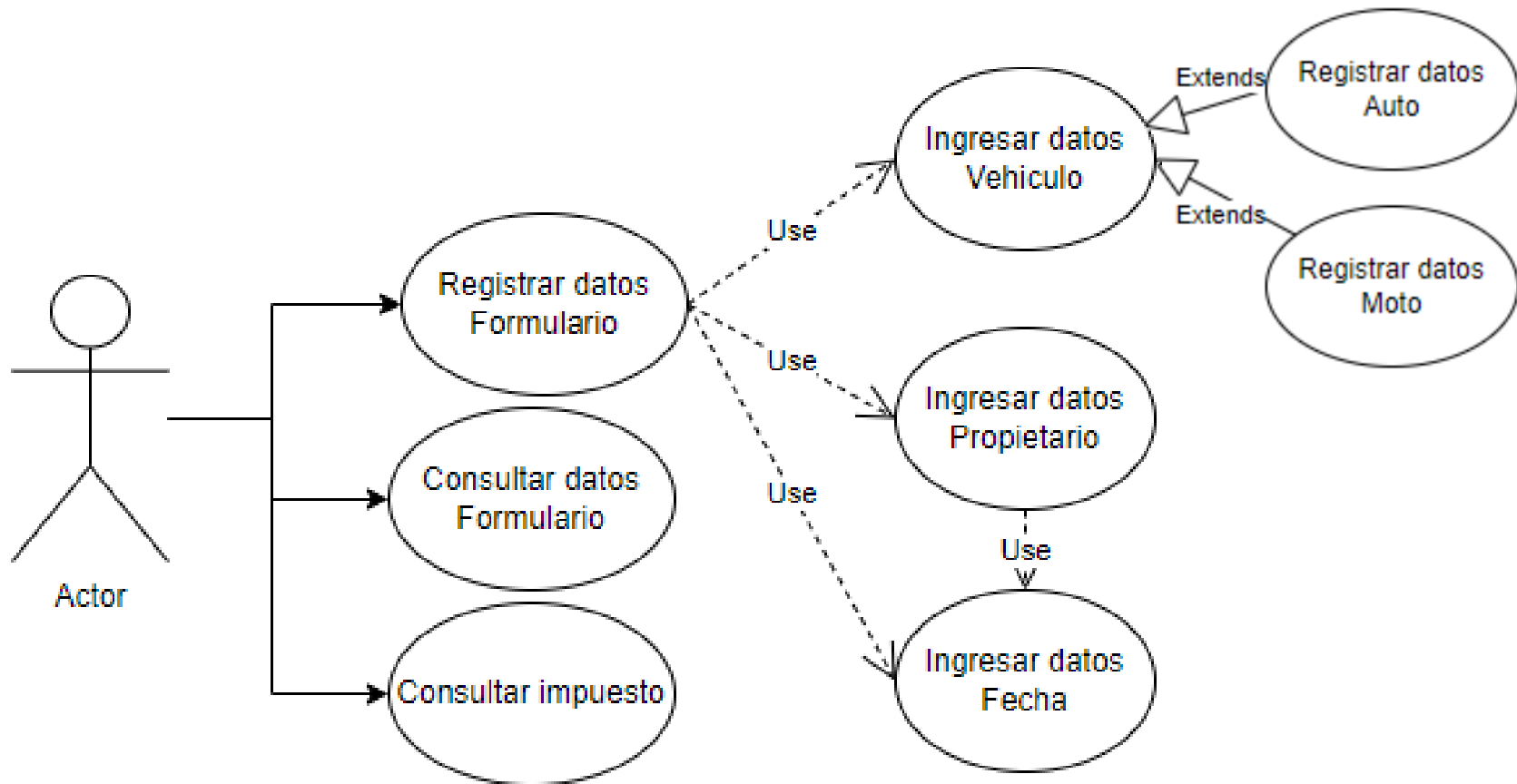
**SEGUNDA FORMA DE POLIMORFISMO,**  
transformar las clases hijas en clase Super





# Polimorfismo

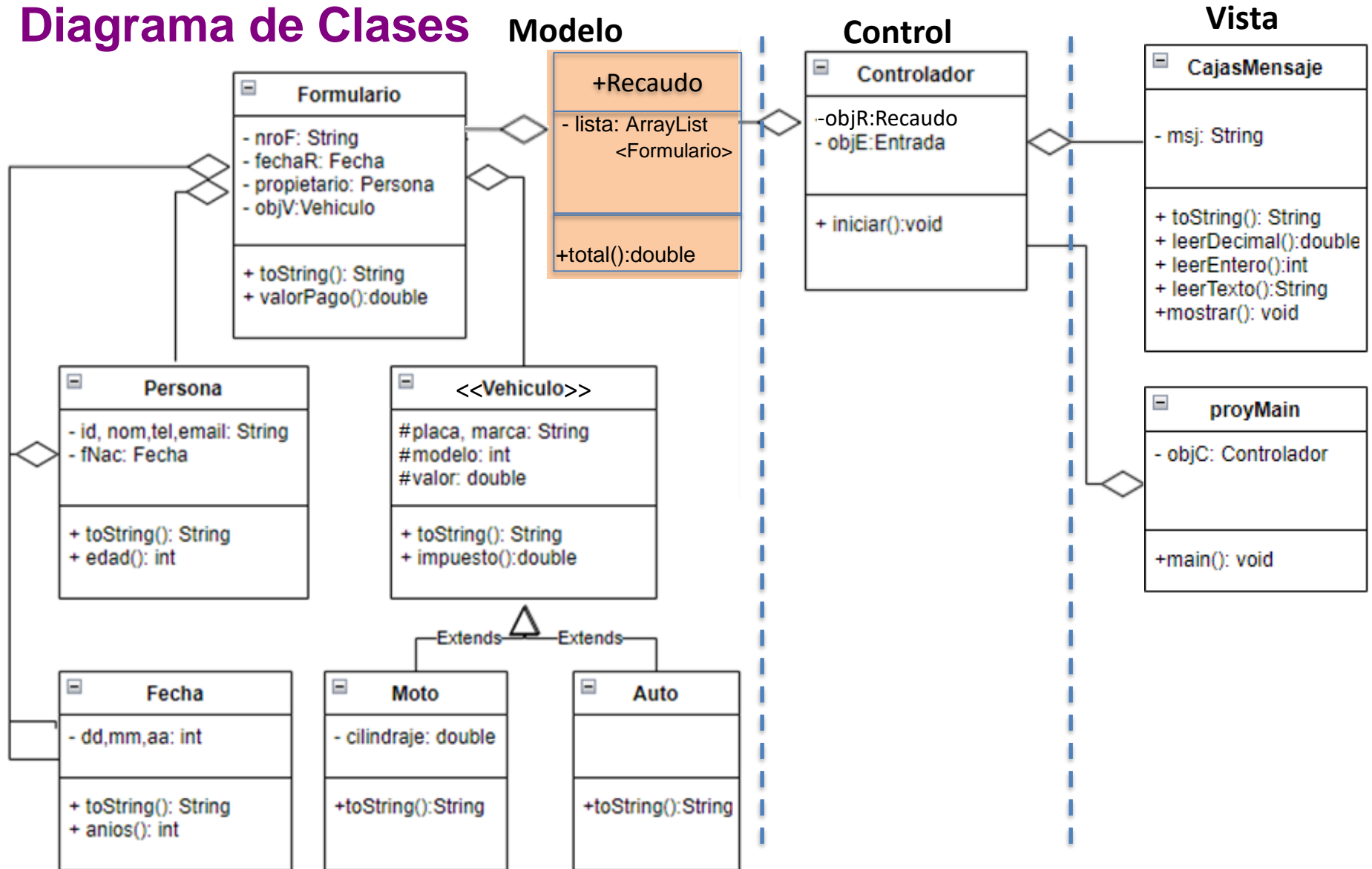
## Diagrama de Caso de Uso





# Polimorfismo

## Diagrama de Clases





# Clase CajasMensaje - Vista

```
public class CajasMensaje {  
    private String titulo,msj;  
    public CajasMensaje(String titulo, String msj) {...4 lines }  
    public CajasMensaje() {  
        this.titulo = "";  
        this.msj = "";  
    }  
    public String getTitulo() {...3 lines }  
    public void setTitulo(String titulo) {...3 lines }  
    public String getMsj() {...3 lines }  
    public void setMsj(String msj) {...3 lines }  
    public int leerEntero(String msj) {  
        int valor;  
        valor=Integer.parseInt(  
            JOptionPane.showInputDialog(null,  
                msj,titulo,1));  
        return valor;  
    }  
    public double leerDecimal(String msj) {...7 lines }  
    public String leerTexto(String msj) {...4 lines }  
    public void mostrar(String msj) {  
        JOptionPane.showMessageDialog(null,  
            msj, titulo, 1);  
    }  
    public boolean confirmar(String msj) {...8 lines }  
}
```



# Clase Persona (Modelo)

```
public class Persona {  
    private String id, nom, tel, email;  
    private Fecha fNac;  
  
    public Persona(String id, String nom, String tel, String email, Fecha fNac)  
    public Persona() {  
        this.id = "";  
        this.nom = "";  
        this.tel = "";  
        this.email = "";  
        this.fNac = new Fecha(); //Agregación  
    }  
    @Override  
    public String toString() {  
        return "Propietario: " +  
            "\n Identificación: " + id +  
            "\n Nombre: " + nom +  
            "\n Teléfono: " + tel +  
            "\n Email: " + email +  
            "\n fecha Nacimiento: " + fNac.toString();  
    }  
    public int edad() {...2 lines }  
    public String getId() {...3 lines }  
    public void setId(String id) {...3 lines }  
    public String getNom() {...3 lines }  
    public void setNom(String nom) {...3 lines }  
    public String getTel() {...3 lines }  
    public void setTel(String tel) {...3 lines }  
    public String getEmail() {...3 lines }  
    public void setEmail(String email) {...3 lines }  
    public Fecha getfNac() {...3 lines }  
    public void setfNac(Fecha fNac) {...3 lines }  
}
```



# Clase Vehiculo (Modelo)

```
package modelo;

public abstract class Vehiculo {
    protected String placa,marca;
    protected int modelo;
    protected double valor;
    public Vehiculo(String placa, String marca, int modelo, double valor) {
        this.placa = placa;    this.marca = marca;
        this.modelo = modelo;  this.valor = valor;
    }
    public Vehiculo() {
        this.placa = "";    this.marca = "";
        this.modelo = 0;    this.valor = 0;
    }
    public String getPlaca() {return placa;    }
    public void setPlaca(String placa) { this.placa = placa;}
    public String getMarca() {return marca;    }
    public void setMarca(String marca) { this.marca = marca;}
    public int getModelo() { return modelo;    }
    public void setModelo(int modelo) { this.modelo = modelo;}
    public double getValor() {return valor;    }
    public void setValor(double valor) { this.valor = valor; }
    @Override
    public String toString() {
        return  "placa: " + placa +
                "\n marca=" + marca +
                "\n modelo=" + modelo +
                "\n valor=" + valor ;
    }
    public abstract double impuesto();
}
```



# Clase Auto (Modelo)

```
public class Auto extends Vehiculo{  
    public Auto(String placa, String marca, int modelo, double valor) {  
        super(placa,marca,modelo, valor);  
    }  
    public Auto() {  
        super();  
    }  
    @Override  
    public double impuesto() {  
        if(modelo<2000)  
            return valor*0.05;  
        else  
            return valor*0.1;  
    }  
}
```



# Clase Formulario (Modelo)

```
public class Formulario {  
    private String nroF;  
    private Fecha fRegistro;  
    private Persona propietario;  
    private Vehiculo objV;  
    public Formulario() {  
        int cod= (int) (Math.random()*999 + 100);  
        this.nroF = "FRM-"+cod;  
        this.fRegistro = new Fecha();  
        this.propietario = new Persona();  
        this.objV = null;  
    }  
    @Override  
    public String toString() {  
        return "Formulario Impuesto\n" +  
            " nroF:" + nroF +  
            "\n fRegistro: " + fRegistro.toString() +  
            "\n Propietario: " + propietario.toString() +  
            "\n Vehículo: \n" + objV.toString();  
    }  
    public double valorPago() {...3 lines }  
    public String getNroF() {...3 lines }  
    public void setNroF(String nroF) {...3 lines }  
    public Fecha getfRegistro() {...3 lines }  
    public void setfRegistro(Fecha fRegistro) {...3 lines }  
    public Persona getPropietario() {...3 lines }  
    public void setPropietario(Persona propietario) {...3 lines }  
    public Vehiculo getObjV() {...3 lines }  
    public void setObjV(Vehiculo objV) {...3 lines }  
}
```



# Clase Recaudo - Modelo

```
public class Recaudo {  
    private ArrayList<Formulario> listaF;  
    public Recaudo(ArrayList<Formulario> listaF) {...3 lines }  
    public Recaudo() {  
        this.listaF = new ArrayList<Formulario>() ;  
    }  
    public double total() {  
        double datos=0;  
        for (int i = 0; i < listaF.size(); i++) {  
            datos+=listaF.get(i).getObjV().impuesto();  
        }  
        return datos;  
    }  
    @Override  
    public String toString() {  
        String datos="";  
        for (int i = 0; i < listaF.size(); i++) {  
            datos+=listaF.get(i).toString()+"\n";  
        }  
        return "Recaudo :\n" + datos;  
    }  
    public int cantAutos() {  
        int cant=0;  
        for (int i = 0; i < listaF.size(); i++) {  
            if(listaF.get(i).getObjV() instanceof Auto)  
                cant++;  
        }  
        return cant;  
    }  
    public ArrayList<Formulario> getListaF() {...3 lines }  
    public void setListaF(ArrayList<Formulario> listaF) {...3 lines }  
}
```





# Clase Controlador - Control

```
public class ControladorPol {  
    CajasMensaje objE; //objeto de la vista  
    Recaudo objR; //objeto del modelo  
    public ControladorPol(CajasMensaje objE, Recaudo objR) {...4 lines }  
    public ControladorPol() {  
        this.objE = new CajasMensaje();  
        this.objR = new Recaudo();  
    }  
    public void iniciar(){  
        objE.setTitulo("Impuestos Vehículo");  
        do{  
            Formulario objF= new Formulario();  
            objE.mostrar("Datos del propietario");  
            objF.getPropietario().setId(objE.leerTexto("Identificación: "));  
            objF.getPropietario().setNom(objE.leerTexto("Nombre: "));  
            objF.getPropietario().setTel(objE.leerTexto("Teléfono: "));  
            objF.getPropietario().setEmail(objE.leerTexto("Email: "));  
            String [] fecha=objE.leerTexto("Fecha Nacimiento (dd/mm/aa): ").split("/");  
            objF.getPropietario().setfNac(new Fecha(Integer.parseInt(fecha[0]),  
                                                    Integer.parseInt(fecha[1]),  
                                                    Integer.parseInt(fecha[2])));  
            int opc= objE.leerEntero("Indique el tipo de vehiculo:\n "+  
                                    "1.Auto \n2.Moto");  
            switch(opc){  
                case 1: Auto objA= new Auto();  
                    objE.setTitulo("Datos del Auto");  
                    objA.setPlaca(objE.leerTexto("Placa: "));  
                    objA.setMarca(objE.leerTexto("Marca: "));  
                    objA.setModelo(objE.leerEntero("Modelo: "));  
                    objA.setValor(objE.leerDecimal("Valor: "));  
                    objF.setObjV(objA); //Polimorfismo  
                    break;  
                case 2: //implementación Con Moto;  
                default: objE.mostrar("Error: Vehículo no existe...");  
            }  
            objE.mostrar("Datos registrados\n"+ objF.toString()+  
                        "\n Pago Impuesto: "+objF.valorPago());  
            objR.getListaf().add(objF);  
        } while(objE.confirmar("Desea registrar otro vehiculo?"));  
        objE.mostrar("Datos registrados\n"+ objR.toString()+  
                    "\n Total Recaudado: "+objR.total());  
    }  
}
```



# Programa Principal

```
public class ProyImpuesto {  
    public static void main(String[] args) {  
        Controlador objC= new Controlador();  
        objC.iniciar();  
    }  
}
```

```
}
```

DATOS



Datos

listado de vehiculos=prop=id=101000222, nom=Pedro Pérez, tel=3102222222, placa=ABC123, marca=suzuki, modelo=2010, valor=7000000.0

prop=id=79888888, nom=Juan Vásquez, tel=3155555555, placa=ZXY246, marca=Yamaha, modelo=2000, cilind=125, valor=3000000.0

Total Recaudo 105000.0

Aceptar



# Bibliografía

---

- Documentación Java Oracle paquete Swing. Disponible en:  
<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>.
- Deitel y Deitel. Programación Java. Editorial Mc Graw Hill.
- Pinzón, Sonia Alexandra. Material de Clase Moodle y Drive.
- Pinzón, Sonia Alexandra. Rodríguez Guerrero, Rocío. Vanegas, Carlos Alberto. Java y el patrón Modelo- Vista – Controlador (MVC). Editorial Universidad Distrital F.J.D.C. 2021