

## **Actividad 4**

### **Programación Orientada a Objetos**

#### **Grupo 4**

#### **Estudiantes**

**Juan Miguel Cadavid Jiménez**  
**Maria Camila Agudelo Espinosa**

#### **Docente**

**Walter Arboleda**

**Medellín**  
**2023**

## Ejercicio 1: Herencia

### Clase Cuenta

```
package banco;

public class Cuenta {

    protected float saldo;
    protected int num_consignaciones = 0;
    protected int num_retiros = 0;
    protected float tasa_anual;
    protected float comision = 0;

    public Cuenta(float saldo, float tasa_anual) {
        this.saldo = saldo;
        this.tasa_anual = tasa_anual;
    }

    public void consignar(float ingreso) {
        this.saldo += ingreso;
        this.num_consignaciones += 1;
    }

    public void retirar(float retiro) {
        if (retiro <= this.saldo) {
            this.saldo -= retiro;
            this.num_retiros += 1;
        }
    }

    private void calcular_interes() {
        float valor_interes = this.saldo*((this.tasa_anual/12)/100);
        this.saldo += valor_interes;
    }

    public void extracto_mensual() {
        this.saldo -= this.comision;
        this.calcular_interes();
    }

    public void imprimir() {
        System.out.println("Valores de la cuenta\n");
        System.out.println("- Saldo: "+this.saldo);
        System.out.println("- Número de consignaciones: "+this.num_consignaciones);
        System.out.println("- Número de retiros: "+this.num_retiros);
    }
}
```

```

        System.out.println("- Tasa anual: "+this.tasa_anual);
        System.out.println("- Comisión: "+this.comision);
    }
}

```

## Clase CuentaAhorros

```

package banco;

public class CuentaAhorros extends Cuenta {

    private boolean activa;

    public CuentaAhorros(float saldo, float tasa_anual) {
        super(saldo, tasa_anual);
        this.activa = (saldo >= 10000);
    }

    public void consignar(float ingreso) {
        if (this.activa) {
            super.consignar(ingreso);
        }
    }

    public void retirar(float retiro) {
        if (this.activa) {
            super.retirar(retiro);
        }
    }

    public void extracto_mensual() {
        if (this.num_retiros > 4) {
            this.saldo -= (1000*(num_retiros-4));
        }
        super.extracto_mensual();
        this.activa = (this.saldo >= 10000);
    }

    public void imprimir() {
        System.out.println("Valores de la cuenta\n");
        System.out.println("- Saldo: "+this.saldo);
        System.out.println("- Comisión: "+this.comision);
        System.out.println("- Número de transacciones:
"+(this.num_retiros+this.num_consignaciones));
    }
}

```

```
}
```

## Clase CuentaCorriente

```
package banco;
```

```
public class CuentaCorriente extends Cuenta {
```

```
    private float sobregiro = 0;
```

```
    public void retirar(float retiro) {  
        if (retiro > this.saldo) {  
            this.sobregiro = (retiro - this.saldo);  
            this.saldo = 0;  
            this.num_retiros += 1;  
        }  
        else {  
            super.retirar(retiro);  
        }  
    }  
}
```

```
    public void consignar(float ingreso) {  
        if (this.sobregiro >= ingreso) {  
            this.sobregiro -= ingreso;  
            this.num_consignaciones += 1;  
        }  
        else {  
            super.consignar(ingreso-sobregiro);  
            this.sobregiro = 0;  
        }  
    }  
}
```

```
    public void extracto_mensual() {  
        super.extracto_mensual();  
    }  
}
```

```
    public void imprimir() {  
        System.out.println("Valores de la cuenta\n");  
        System.out.println("- Saldo: "+this.saldo);  
        System.out.println("- Comisión: "+this.comision);  
        System.out.println("- Número de transacciones:  
"+(this.num_consignaciones+this.num_retiros));  
        System.out.println("- Sobregiro: "+this.sobregiro);  
    }  
}
```

## Clase Main

```
import java.util.Scanner;

import banco.CuentaAhorros;

public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        CuentaAhorros myCuenta = new CuentaAhorros(10000, 12);

        while (true) {
            System.out.println("-----");
            System.out.println("Menú:");
            System.out.println("1- Consignar");
            System.out.println("2- Retirar");
            System.out.println("3- Calcular extracto");
            System.out.println("4- Ver valores de la cuenta");
            System.out.println("5- Salir");
            System.out.println("");
            System.out.println("Seleccione una opción: ");
            System.out.println("-----");

            int option = in.nextInt();

            if (option == 1) {
                System.out.println("Ingrese el monto a consignar: ");
                float test = in.nextFloat();
                System.out.println("Consignado: "+test);
                myCuenta.consignar(test);
            }
            else if (option == 2) {
                System.out.println("Ingrese el monto a retirar: ");
                myCuenta.retirar(in.nextFloat());
            }
            else if (option == 3) {
                myCuenta.extracto_mensual();
                System.out.println("Los extractos mensuales fueron calculados...\n");
            }
            else if (option == 4) {
                myCuenta.imprimir();
            }
            else {
```

```

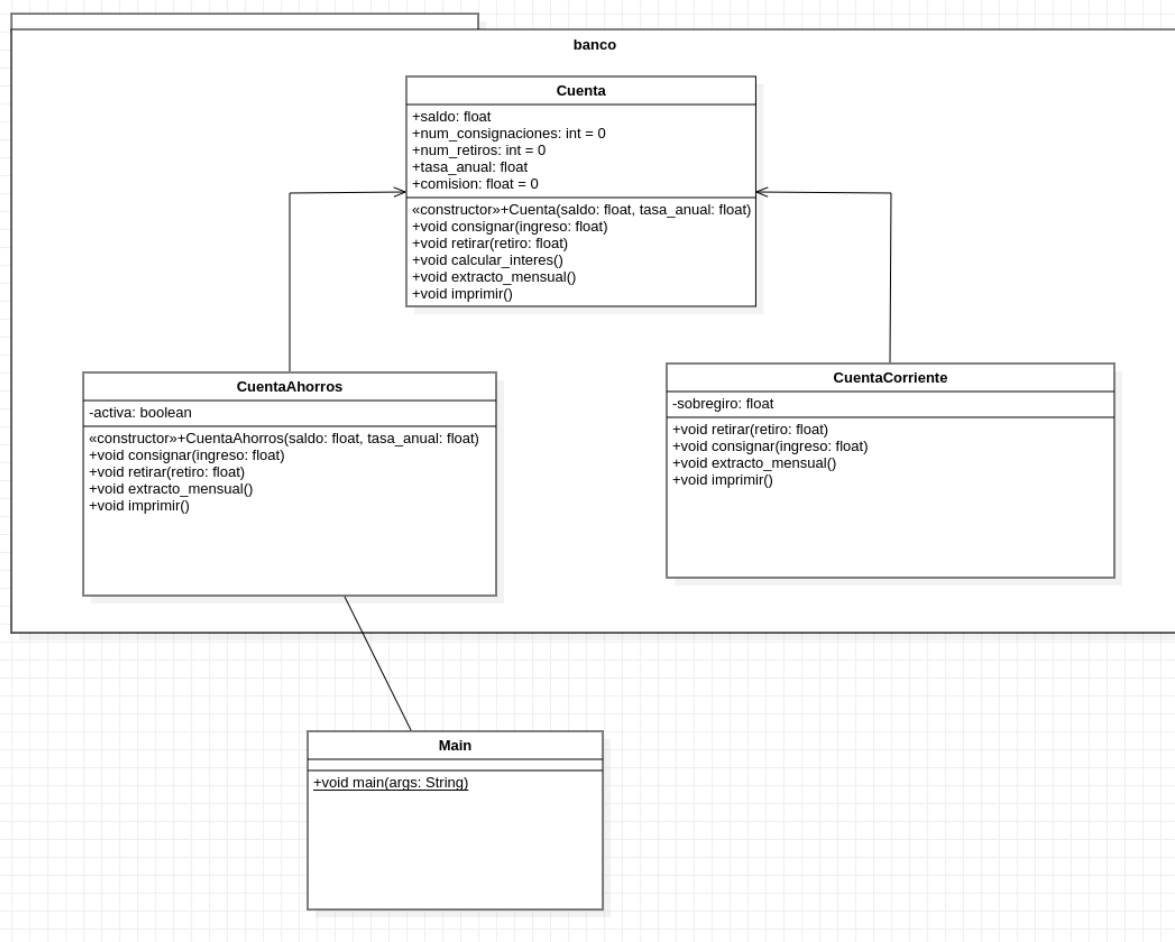
        break;
    }
}

}

}

```

## Diagrama de clases (UML)



## Código en Github:

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4/herencia](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4/herencia)

## Ejercicio 2: Paquetes

### Clase Local

```
package inmobiliario.local;

import inmobiliario.Inmueble;

public abstract class Local extends Inmueble {

    public String localizacion;

    public Local() {}

    public Local(int tipo, int area, String direccion, String localizacion) {
        super(tipo, area, direccion);
        this.localizacion = localizacion;
    }
}
```

### Clase Comercial

```
package inmobiliario.local;

import java.math.BigDecimal;

public class Comercial extends Local {

    public String centro_comercial;

    public Comercial(int tipo, int area, String direccion, String localizacion, String
centro_comercial) {
        super(tipo, area, direccion, localizacion);
        this.centro_comercial = centro_comercial;
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(3000000));
    }
}
```

## Clase Oficina

```
package inmobiliario.local;

import java.math.BigDecimal;

public class Oficina extends Local {

    public boolean gobierno;

    public Oficina() {}

    public Oficina(int tipo, int area, String direccion, String localizacion, boolean gobierno) {
        super(tipo, area, direccion, localizacion);
        this.gobierno = gobierno;
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(3500000));
    }

}
```

## Clase Inmueble

```
package inmobiliario;
import java.math.BigDecimal;

public abstract class Inmueble {

    protected int tipo;
    protected int area;
    protected String direccion;

    public Inmueble() {}

    public Inmueble(int tipo, int area, String direccion) {
        this.tipo = tipo;
        this.area = area;
        this.direccion = direccion;
    }

    public abstract BigDecimal valor_compra();

}
```



## Clase Vivienda

```
package inmobiliario.vivienda;

import inmobiliario.Inmueble;

public abstract class Vivienda extends Inmueble {

    protected int habitaciones;
    protected int baños;

    public Vivienda() {}

    public Vivienda(int tipo, int area, String direccion, int habitaciones, int baños) {
        super(tipo, area, direccion);
        this.habitaciones = habitaciones;
        this.baños = baños;
    }
}
```

## Clase Apartaestudio

```
package inmobiliario.vivienda.apartamento;

import inmobiliario.vivienda.Vivienda;
import java.math.BigDecimal;

public class Apartaestudio extends Vivienda {

    public Apartaestudio() {
        this.habitaciones = 1;
    }

    public Apartaestudio(int tipo, int area, String direccion, int baños) {
        super(tipo, area, direccion, 1, baños);
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(1500000));
    }
}
```

## Clase Familiar

```
package inmobiliario.vivienda.apartamento;

import inmobiliario.vivienda.Vivienda;
import java.math.BigDecimal;

public class Familiar extends Vivienda {

    public float valor_administracion;

    public Familiar() {}

    public Familiar(int tipo, int area, String direccion, int habitaciones, int baños, float
valor_administracion) {
        super(tipo, area, direccion, habitaciones, baños);
        this.valor_administracion = valor_administracion;
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(2000000));
    }

}
```

## Clase Casa

```
package inmobiliario.vivienda.casa;

import inmobiliario.vivienda.Vivienda;

public abstract class Casa extends Vivienda {

    public int pisos;

    public Casa() {}

    public Casa(int tipo, int area, String direccion, int habitaciones, int baños, int pisos) {
        super(tipo, area, direccion, habitaciones, baños);
        this.pisos = pisos;
    }

}
```

## Clase Rural

```
package inmobiliario.vivienda.casa;
import java.math.BigDecimal;

public class Rural extends Casa {

    public float distancia_municipio;
    public float altitud;

    public Rural() {}

    public Rural(int tipo, int area, String direccion, int habitaciones, int baños, int pisos, float
    distancia_municipio, float altitud) {
        super(tipo, area, direccion, habitaciones, baños, pisos);
        this.distancia_municipio = distancia_municipio;
        this.altitud = altitud;
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(1500000));
    }
}
```

## Clase Conjunto

```
package inmobiliario.vivienda.casa.urbana;

import inmobiliario.vivienda.casa.Casa;
import java.math.BigDecimal;

public class Conjunto extends Casa {

    public float valor_administracion;
    public boolean incluye_areas_comunes;

    public Conjunto() {}

    public Conjunto(int tipo, int area, String direccion, int habitaciones, int baños, int pisos,
    float valor_administracion, boolean incluye_areas_comunes) {
        super(tipo, area, direccion, habitaciones, baños, pisos);
        this.valor_administracion = valor_administracion;
        this.incluye_areas_comunes = incluye_areas_comunes;
    }

    public BigDecimal valor_compra() {
```

```

        return new BigDecimal(this.area).multiply(new BigDecimal(2500000));
    }
}

```

## Clase Independiente

```

package inmobiliario.vivienda.casa.urbana;

import inmobiliario.vivienda.casa.Casa;
import java.math.BigDecimal;

public class Independiente extends Casa {

    public Independiente() {}

    public Independiente(int tipo, int area, String direccion, int habitaciones, int baños, int
pisos) {
        super(tipo, area, direccion, habitaciones, baños, pisos);
    }

    public BigDecimal valor_compra() {
        return new BigDecimal(this.area).multiply(new BigDecimal(3000000));
    }

}

```

## Clase Main

```

import inmobiliario.vivienda.casa.Rural;
import inmobiliario.vivienda.casa.urbana.Conjunto;
import inmobiliario.vivienda.casa.urbana.Independiente;
import inmobiliario.vivienda.apartamento.Familiar;
import inmobiliario.vivienda.apartamento.Apartaestudio;
import inmobiliario.local.Comercial;
import inmobiliario.local.Oficina;

public class Main {

    public static void main(String[] args) {

        Rural casa_rural = new Rural(0, 300, "Carrera 70 #82 40", 5, 2, 2, 1052, 1520);
        Conjunto casa_conjunto = new Conjunto(1, 100, "Calle 23 #32 16", 3, 1, 2, 230000,
true);
        Independiente casa_independiente = new Independiente(2, 180, "Cll 80 #32 44", 5, 2,
1);
    }
}

```

```

Familiar apartamento_familiar = new Familiar(3, 80, "Cra 30 #12 24", 3, 1, 180000);
Apartaestudio apartaestudio = new Apartaestudio(4, 60, "Cra 22 #11 6", 1);
Comercial local_comercial = new Comercial(5, 120, "Cll 50 #60 27", "Afuera", "CC
Parque Fabricato");
Oficina oficina = new Oficina(6, 180, "Cll 70 #72 44", "Adentro", false);

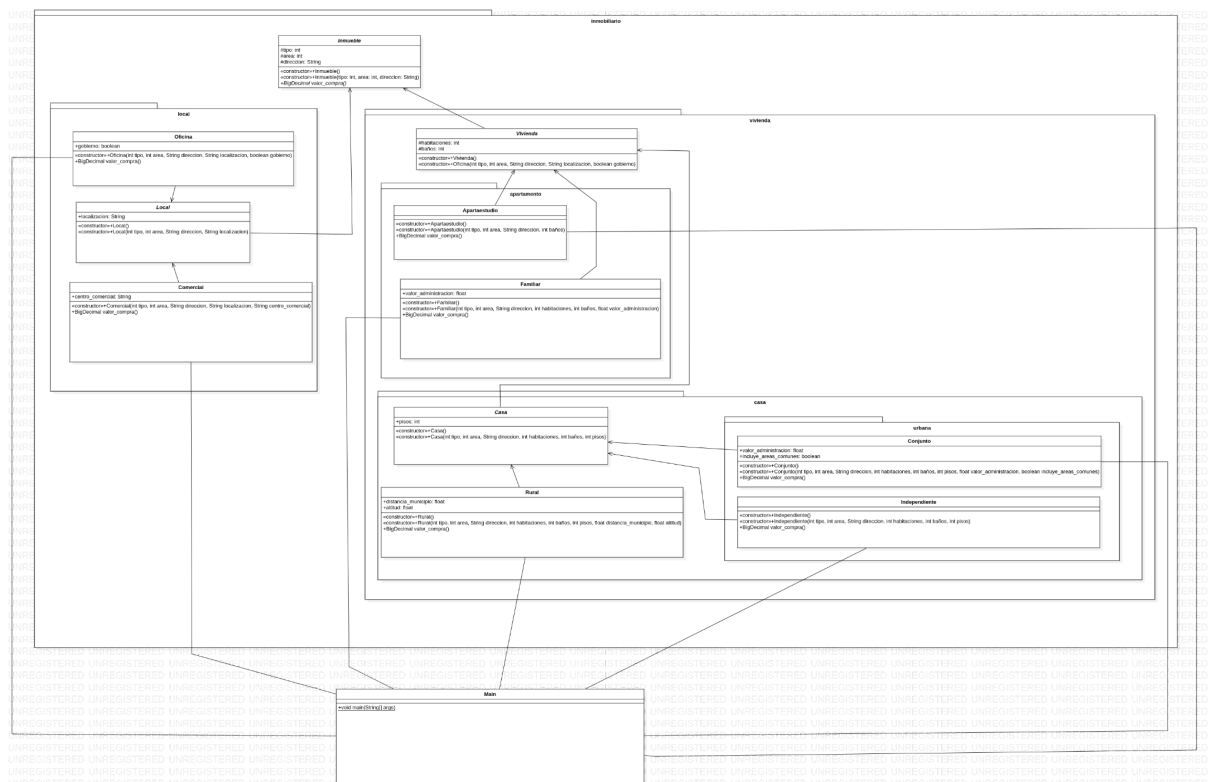
System.out.println("\nValores:");
System.out.println("Casa Rural: "+casa_rural.valor_compra());
System.out.println("Casa Urbana Conjunto Residencial:
"+casa_conjunto.valor_compra());
System.out.println("Casa Urbana Independiente:
"+casa_independiente.valor_compra());
System.out.println("Apartamento Familiar: "+apartamento_familiar.valor_compra());
System.out.println("Apartaestudio: "+apartaestudio.valor_compra());
System.out.println("Local comercial: "+local_comercial.valor_compra());
System.out.println("Oficina: "+oficina.valor_compra());

}

}

```

## Diagrama de clases (UML)



## Código en Github:

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4/paquetes](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4/paquetes)

## Ejercicio 3: Clases abstractas

### Clase Animal

```
package animales;

public abstract class Animal {

    protected String sonido;
    protected String alimento;
    protected String habitat;
    protected String nombre_cientifico;

    public abstract String getNombreCientifico();
    public abstract String getSonido();
    public abstract String getAlimento();
    public abstract String getHabitat();

}
```

### Clase Canido

```
package animales;

public abstract class Canido extends Animal {

    public String getNombreCientifico() {
        return this.nombre_cientifico;
    }

    public String getSonido() {
        return this.sonido;
    }

    public String getAlimento() {
```

```
        return this.alimento;
    }

    public String getHabitat() {
        return this.habitat;
    }
}
```

## Clase Felino

```
package animales;

public abstract class Felino extends Animal {

    public String getNombreCientifico() {
        return this.nombre_cientifico;
    }

    public String getSonido() {
        return this.sonido;
    }

    public String getAlimento() {
        return this.alimento;
    }

    public String getHabitat() {
        return this.habitat;
    }
}
```

## Clase Gato

```
package animales;

public class Gato extends Felino {

    public Gato() {
        this.sonido = "maullido";
        this.alimento = "ratones";
        this.habitat = "domestico";
        this.nombre_cientifico = "Felis silvestris catus";
    }
}
```

```
}
```

## Clase Leon

```
package animales;
```

```
public class Leon extends Felino {
```

```
    public Leon() {  
        this.sonido = "rugido";  
        this.alimento = "carnívoro";  
        this.habitat = "pradera";  
        this.nombre_cientifico = "Panthera leo";  
    }
```

```
}
```

## Clase Lobo

```
package animales;
```

```
public class Lobo extends Canido {
```

```
    public Lobo() {  
        this.sonido = "aullido";  
        this.alimento = "carnívoro";  
        this.habitat = "bosque";  
        this.nombre_cientifico = "Canislupus";  
    }
```

```
}
```

## Clase Perro

```
package animales;
```

```
public class Perro extends Canido {
```

```
    public Perro() {  
        this.sonido = "ladrido";  
        this.alimento = "carnívoro";  
        this.habitat = "doméstico";  
        this.nombre_cientifico = "Canislupus familiaris";  
    }
```

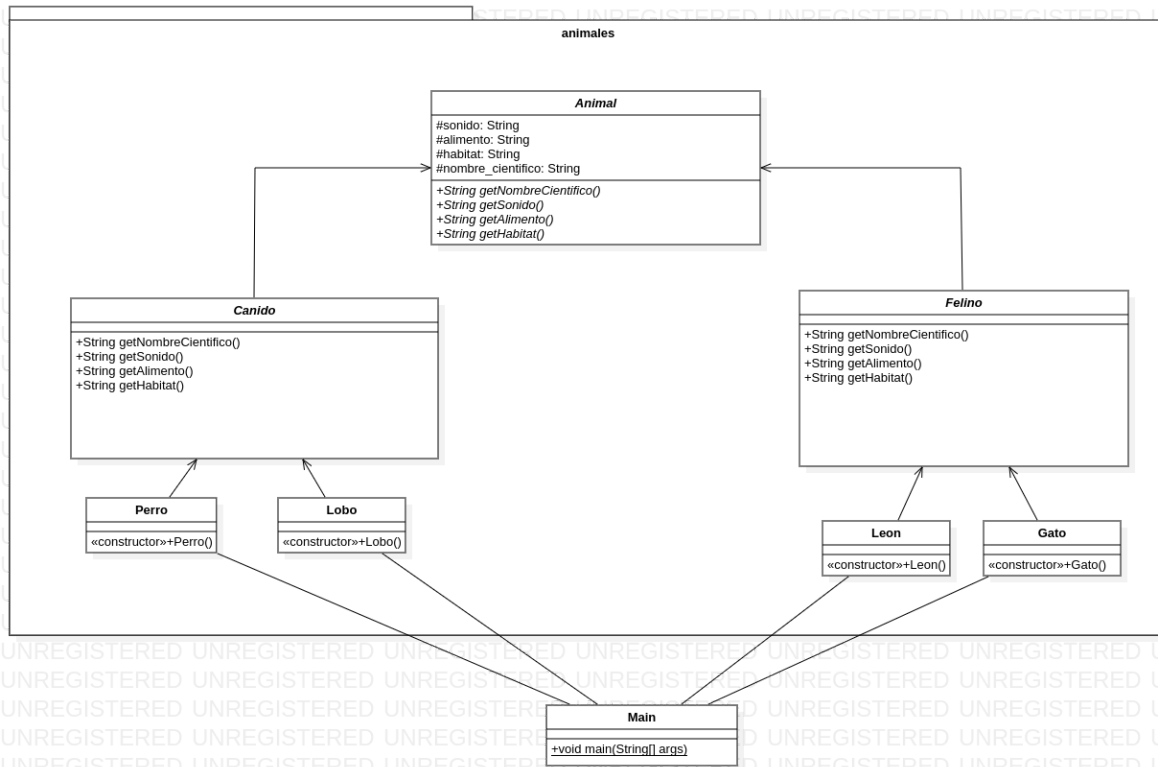


```
}  
  
}
```

## Clase Main

```
import animales.*;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Animal[] myAnimals = {new Perro(), new Lobo(), new Gato(), new Leon()};  
        for (int i=0; i<myAnimals.length; i++) {  
            System.out.println("Animal "+(i+1));  
            System.out.println("Nombre científico: "+myAnimals[i].getNombreCientifico());  
            System.out.println("Hábitat: "+myAnimals[i].getHabitat());  
            System.out.println("Alimento: "+myAnimals[i].getAlimento());  
            System.out.println("Sonido: "+myAnimals[i].getSonido()+"\n");  
        }  
  
    }  
  
}
```

## Diagrama de clases (UML)



## Código en Github:

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4/clases\\_abstractas](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4/clases_abstractas)

## Ejercicio 4: Métodos abstractos

### Clase Ciclista

```
package clases;
```

```
public abstract class Ciclista {  
  
    private int identificador;  
    private String nombre;  
    private float tiempo_acumulado = 0;  
  
    public Ciclista() {};  
  
    public Ciclista(int identificador, String nombre, float tiempo_acumulado) {  
        this.identificador = identificador;  
        this.nombre = nombre;  
        this.tiempo_acumulado = tiempo_acumulado;  
    }  
  
    public abstract String imprimirTipo();  
  
    public int getIdentificador() {  
        return this.identificador;  
    }  
  
    public String getNombre() {  
        return this.nombre;  
    }  
  
    public float getTiempoAcumulado() {  
        return this.tiempo_acumulado;  
    }  
  
    public void setIdentificador(int identificador) {  
        this.identificador = identificador;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public void setTiempoAcumulado(float tiempo_acumulado) {  
        this.tiempo_acumulado = tiempo_acumulado;  
    }  
}
```

```

    public void imprimir() {
        System.out.println("Identificador: " + this.identificador);
        System.out.println("Nombre: " + this.nombre);
        System.out.println("Tiempo acumulado: " + this.tiempo_acumulado);
    }
}

```

## Clase Contrarrelojista

```

package clases;

public class Contrarrelojista extends Ciclista {

    private double velocidad_maxima;

    public Contrarrelojista() {}

    public Contrarrelojista(int identificador, String nombre, float tiempo_acumulado, double
velocidad_maxima) {
        super(identificador, nombre, tiempo_acumulado);
        this.velocidad_maxima = velocidad_maxima;
    }

    public double getVelocidadMaxima() {
        return this.velocidad_maxima;
    }

    public void setVelocidadMaxima(double velocidad_maxima) {
        this.velocidad_maxima = velocidad_maxima;
    }

    public String imprimirTipo() {
        return "Es un contrarrelojista";
    }

    public void imprimir() {
        super.imprimir();
        System.out.println("Velocidad máxima: " + this.velocidad_maxima);
    }
}

```

## Clase Equipo

```
package clases;
```

```
public class Equipo {

    private String nombre;
    private Ciclista[] ciclistas;
    private static double tiempo_total = 0;

    public Equipo() {}

    public Equipo(String nombre, Ciclista[] ciclistas) {
        this.nombre = nombre;
        this.ciclistas = ciclistas;
    }

    public String getNombre() {
        return this.nombre;
    }

    public Ciclista[] getCiclistas() {
        return this.ciclistas;
    }

    public double getTiempo() {
        this.computeTime();
        return this.tiempo_total;
    }

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

    public void setCiclistas(Ciclista[] ciclistas) {
        this.ciclistas = ciclistas;
    }

    private static Ciclista[] push(Ciclista[] array, Ciclista push) {
        Ciclista[] longer = new Ciclista[array.length + 1];
        for (int i = 0; i < array.length; i++)
            longer[i] = array[i];
        longer[array.length] = push;
        return longer;
    }

    public void agregarCiclista(Ciclista ciclista) {
```

```

        this.ciclistas = this.push(this.ciclistas, ciclista);
    }

    public void computeTime() {
        double tiempo = 0;
        for (int i=0; i<this.ciclistas.length; i++) {
            tiempo = tiempo + this.ciclistas[i].getTiempoAcumulado();
        }
        this.tiempo_total = tiempo;
    }

    public void mostrarNombres() {
        System.out.println("Nombres:");
        for (int i=0; i<this.ciclistas.length; i++) {
            System.out.println(ciclistas[i].getNombre());
        }
    }

    public void mostrarDatos(int id) {
        for (int i=0; i<this.ciclistas.length; i++) {
            if (ciclistas[i].getIdentificador() == id) {
                System.out.println(ciclistas[i].imprimirTipo());
                ciclistas[i].imprimir();
            }
        }
    }
}

```

## Clase Escalador

```
package clases;
```

```

public class Escalador extends Ciclista {

    private float aceleracion_promedio_subida;
    private float grado_soportado;

    public Escalador() {}

    public Escalador(int identificador, String nombre, float tiempo_acumulado, float
    aceleracion_promedio_subida, float grado_soportado) {
        super(identificador, nombre, tiempo_acumulado);
        this.aceleracion_promedio_subida = aceleracion_promedio_subida;
        this.grado_soportado = grado_soportado;
    }
}

```

```

    public float getAceleracionPromedioSubida() {
        return this.aceleracion_promedio_subida;
    }

    public float getGradoSoportado() {
        return this.grado_soportado;
    }

    public void setAceleracionPromedioSubida(float aceleracion_promedio_subida) {
        this.aceleracion_promedio_subida = aceleracion_promedio_subida;
    }

    public void setGradoSoportado(float grado_soportado) {
        this.grado_soportado = grado_soportado;
    }

    public String imprimirTipo() {
        return "Es un escalador";
    }

    public void imprimir() {
        super.imprimir();
        System.out.println("Aceleración promedio en subida: " +
this.aceleracion_promedio_subida);
        System.out.println("Grado soportado: " + this.grado_soportado);
    }
}

```

## Clase Velocista

```
package clases;
```

```

public class Velocista extends Ciclista {

    private double potencia_promedio;
    private double velocidad_promedio;

    public Velocista() {};

    public Velocista(int identificador, String nombre, float tiempo_acumulado, double
potencia_promedio, double velocidad_promedio) {
        super(identificador, nombre, tiempo_acumulado);
        this.potencia_promedio = potencia_promedio;
        this.velocidad_promedio = velocidad_promedio;
    }
}

```

```

    }

    public double getPotenciaPromedio() {
        return this.potencia_promedio;
    }

    public double getVelocidadPromedio() {
        return this.velocidad_promedio;
    }

    public void setPotenciaPromedio(double potencia_promedio) {
        this.potencia_promedio = potencia_promedio;
    }

    public void setVelocidadPromedio(double velocidad_promedio) {
        this.velocidad_promedio = velocidad_promedio;
    }

    public String imprimirTipo() {
        return "Es un velocista";
    }

    public void imprimir() {
        super.imprimir();
        System.out.println("Potencia promedio: " + this.potencia_promedio);
        System.out.println("Velocidad promedio: " + this.velocidad_promedio);
    }
}

```

## Clase Main

```

import classes.*;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Velocista velocista = new Velocista(1, "Juan", 10.5f, 1.5, 5.0);
        Escalador escalador = new Escalador(2, "Andres", 12.2f, 2.5f, 1.32f);
        Contrarrelojista contrarrelojista = new Contrarrelojista(3, "María", 23.2f, 3.4);

        Ciclista[] ciclistas = {velocista, escalador, contrarrelojista};
    }
}

```



```
while (true) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Ingrese un identificador para mostrar los datos del ciclista: ");
    equipo.mostrarDatos(sc.nextInt());
}
```

```

classDiagram
    class Equipo {
        +nombre: String
        +ciclistas: Array<Ciclista>
        +tiempo_total: double = 0
        +constructor=Equipo()
        +constructor=Equipo(String nombre, Ciclista[] ciclistas)
        +String getNombre()
        +Ciclista getCiclista()
        +void agregarCiclista(Ciclista ciclista)
        +void computeTime()
        +void mostrarNumeros()
        +void mostrarDatos(int id)
    }
    class Ciclista {
        +identificador: int
        +nombre: String
        +tiempo_acumulado: float = 0
        +constructor=Ciclista()
        +constructor=Ciclista(int identificador, String nombre, float tiempo_acumulado)
        +String imprimirTipo()
        +int getIdentificador()
        +String getNombre()
        +float getTiempoAcumulado()
        +void setIdentificador(int identificador)
        +void setNombre(String nombre)
        +void setTiempoAcumulado(float tiempo_acumulado)
        +void imprimir()
    }
    class Velocista {
        +potencia_promedio: double
        +velocidad_promedio: double
        +constructor=Velocista()
        +constructor=Velocista(int identificador, String nombre, float tiempo_acumulado, double potencia_promedio, double velocidad_promedio)
        +double getPotenciaPromedio()
        +double getVelocidadPromedio()
        +void setPotenciaPromedio()
        +void setVelocidadPromedio()
        +String imprimirTipo()
        +void imprimir()
    }
    class Escalador {
        +aceleracion_promedio_subida: float
        +grado_soportado: float
        +constructor=Escalador()
        +constructor=Escalador(int identificador, String nombre, float tiempo_acumulado, float aceleracion_promedio_subida, float grado_soportado)
        +float getAceleracionPromedioSubida()
        +float getGradoSoportado()
        +void setAceleracionPromedioSubida(float aceleracion_promedio_subida)
        +void setGradoSoportado(float grado_soportado)
        +String imprimirTipo()
        +void imprimir()
    }
    class Contrarrelogista {
        +velocidad_maxima: double
        +atributos
        +constructor=Contrarrelogista()
        +constructor=Contrarrelogista(int identificador, String nombre, float tiempo_acumulado, double velocidad_maxima)
        +double getVelocidadMaxima()
        +void setVelocidadMaxima(double velocidad_maxima)
        +String imprimirTipo()
        +void imprimir()
    }
    class Main {
        +void main(String[] args)
    }
    Equipo --> Ciclista
    Equipo --> Velocista
    Equipo --> Escalador
    Equipo --> Contrarrelogista
    Main --> Equipo
    Main --> Ciclista
    Main --> Velocista
    Main --> Escalador
    Main --> Contrarrelogista

```

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4/metodos\\_abstractos](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4/metodos_abstractos)

## Ejercicio 5: Interfaces gráficas

### Clase Persona

```
package clases;

public class Persona {

    public String nombre, apellidos, direccion, telefono;

    public Persona(String nombre, String apellidos, String direccion, String telefono) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.direccion = direccion;
        this.telefono = telefono;
    }

    public String toString() {
        return this.nombre+" "+this.apellidos;
    }

}
```

### Clase AgregarPersona (Interfaz)

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this
 template
 */
package interfaz;

import clases.Persona;

/**
 *
 * @author jmcada
 */
public class AgregarPersona extends javax.swing.JFrame {

    /**
     * Creates new form AgregarPersona
     */

    public javax.swing.DefaultListModel<Persona> personas;
```

```

public AgregarPersona() {
    initComponents();
}

public void removeValues() {
    this.txt_nombre.setText("");
    this.txt_apellidos.setText("");
    this.txt_direccion.setText("");
    this.txt_telefono.setText("");
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    txt_telefono = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    txt_nombre = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    txt_apellidos = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    txt_direccion = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("AGREGAR PERSONA");

    jLabel7.setText("Teléfono:");

    jLabel3.setText("Nombre:");

    jLabel4.setText("Apellidos:");

    jLabel6.setText("Dirección:");

    jButton1.setText("Agregar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Cancelar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton2)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(10, 10, 10)
                        .addComponent(jButton1)))
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel7)
                .addGap(10, 10, 10)
                .addComponent(txt_telefono)
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel6)
                .addGap(10, 10, 10)
                .addComponent(txt_direccion)
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel4)
                .addGap(10, 10, 10)
                .addComponent(txt_apellidos)
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel3)
                .addGap(10, 10, 10)
                .addComponent(txt_nombre)
                .addContainerGap(10, Short.MAX_VALUE))
    );

```

```

        .addComponent(jLabel1)
        .addGap(147, 147, 147)))
        .addContainerGap(32, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(26, 26, 26)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(txt_nombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(txt_apellidos, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel6)
                .addComponent(txt_direccion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel7)
                .addComponent(txt_telefono, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 31,
Short.MAX_VALUE)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
                .addGap(22, 22, 22))
    );

    pack();
} // </editor-fold> //GEN-END: initComponents

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton2ActionPerformed
    dispose();

```

```

} //GEN-LAST:event_jButton2ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton1ActionPerformed
    this.personas.addElement(new Persona(this.txt_nombre.getText(),
this.txt_apellidos.getText(), this.txt_direccion.getText(), this.txt_telefono.getText()));
    dispose();
} //GEN-LAST:event_jButton1ActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(AgregarPersona.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(AgregarPersona.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(AgregarPersona.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AgregarPersona.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
    }
}
//</editor-fold>

    /* Create and display the form */

```

```

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new AgregarPersona().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JTextField txt_apellidos;
    private javax.swing.JTextField txt_direccion;
    private javax.swing.JTextField txt_nombre;
    private javax.swing.JTextField txt_telefono;
    // End of variables declaration//GEN-END:variables
}

```

## Clase MainWindow (Interfaz)

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 * this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this
 * template
 */
package interfaz;

import clases.Persona;

/**
 *
 * @author jmcada
 */
public class MainWindow extends javax.swing.JFrame {

    /**
     * Creates new form MainWindow
     */

    public javax.swing.DefaultListModel<Persona> personas;
    public AgregarPersona add_window;
}

```

```

public javax.swing.JList<Persona> lista_personas;

public MainWindow() {
    initComponents();

    add_window = new AgregarPersona();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    lista_personas = new javax.swing.JList<>();
    btn_agregar = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    txt_nombre = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    txt_apellidos = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    txt_direccion = new javax.swing.JTextField();
    jLabel7 = new javax.swing.JLabel();
    txt_telefono = new javax.swing.JTextField();
    btn_eliminar = new javax.swing.JButton();
    btn_borrar_todos = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("INFORMACIÓN DE LA PERSONA SELECCIONADA:");

    lista_personas.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
        public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
            lista_personasValueChanged(evt);
        }
    });
    jScrollPane1.setViewportView(lista_personas);

    btn_agregar.setText("Agregar");
    btn_agregar.addActionListener(new java.awt.event.ActionListener() {

```



```

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

    jLabel2.setText("PERSONAS");

    jLabel3.setText("Nombre:");

    jLabel4.setText("Apellidos:");

    jLabel6.setText("Dirección:");

    jLabel7.setText("Teléfono:");

    btn_eliminar.setText("Eliminar seleccionado");
    btn_eliminar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_eliminarActionPerformed(evt);
        }
    });

    btn_borrar_todos.setText("Borrar todos");
    btn_borrar_todos.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_borrar_todosActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(42, 42, 42)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel7)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(txt_telefono))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel6)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(txt_direccion, javax.swing.GroupLayout.DEFAULT_SIZE,
379, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)

```

```

        .addGap(18, 18, 18)
        .addComponent(txt_apellidos))
    .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel3)
        .addGap(18, 18, 18)
        .addComponent(txt_nombre))
    .addComponent(jLabel1)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btn_agregar)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btn_eliminar))
    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
454, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btn_borrar_todos,
javax.swing.GroupLayout.Alignment.TRAILING))
    .addContainerGap(64, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(btn_agregar)
                .addComponent(jLabel2)
                .addComponent(btn_eliminar)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btn_borrar_todos)
                .addGap(28, 28, 28)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(txt_nombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)

```

```

        .addComponent(txt_apellidos, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel6)
            .addComponent(txt_direccion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel7)
            .addComponent(txt_telefono, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(57, Short.MAX_VALUE))
    );

```

```

    pack();
} // </editor-fold> // GEN-END: initComponents

```

```

private void btn_agregarActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btn_agregarActionPerformed
    this.add_window.removeValues();
    this.add_window.setVisible(true);
} // GEN-LAST:event_btn_agregarActionPerformed

```

```

private void lista_personasValueChanged(javax.swing.event.ListSelectionEvent evt)
{ // GEN-FIRST:event_lista_personasValueChanged
    Persona persona = this.lista_personas.getSelectedValue();
    if (persona != null) {
        this.txt_nombre.setText(persona.nombre);
        this.txt_apellidos.setText(persona.telefono);
        this.txt_direccion.setText(persona.direccion);
        this.txt_telefono.setText(persona.telefono);
    }
    else {
        this.txt_nombre.setText("");
        this.txt_apellidos.setText("");
        this.txt_direccion.setText("");
        this.txt_telefono.setText("");
    }
} // GEN-LAST:event_lista_personasValueChanged

```

```

private void btn_eliminarActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btn_eliminarActionPerformed
    this.personas.remove(this.lista_personas.getSelectedIndex());
} // GEN-LAST:event_btn_eliminarActionPerformed

```

```

private void btn_borrar_todosActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btn_borrar_todosActionPerformed
        this.personas.clear();
    //GEN-LAST:event_btn_borrar_todosActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
        feel.
         * For details see
         * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.
            SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.
            SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.
            SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.
            SEVERE, null, ex);
        }
    }
    //</editor-fold>

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

```

```

    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btn_agregar;
private javax.swing.JButton btn_borrar_todos;
private javax.swing.JButton btn_eliminar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField txt_apellidos;
private javax.swing.JTextField txt_direccion;
private javax.swing.JTextField txt_nombre;
private javax.swing.JTextField txt_telefono;
// End of variables declaration//GEN-END:variables
}

```

## Clase Main

```

import interfaz.MainWindow;
import clases.Persona;

import java.util.ArrayList;
import javax.swing.DefaultListModel;

public class Main {

    public static void main(String[] args) {

        MainWindow window = new MainWindow();
        window.setVisible(true);

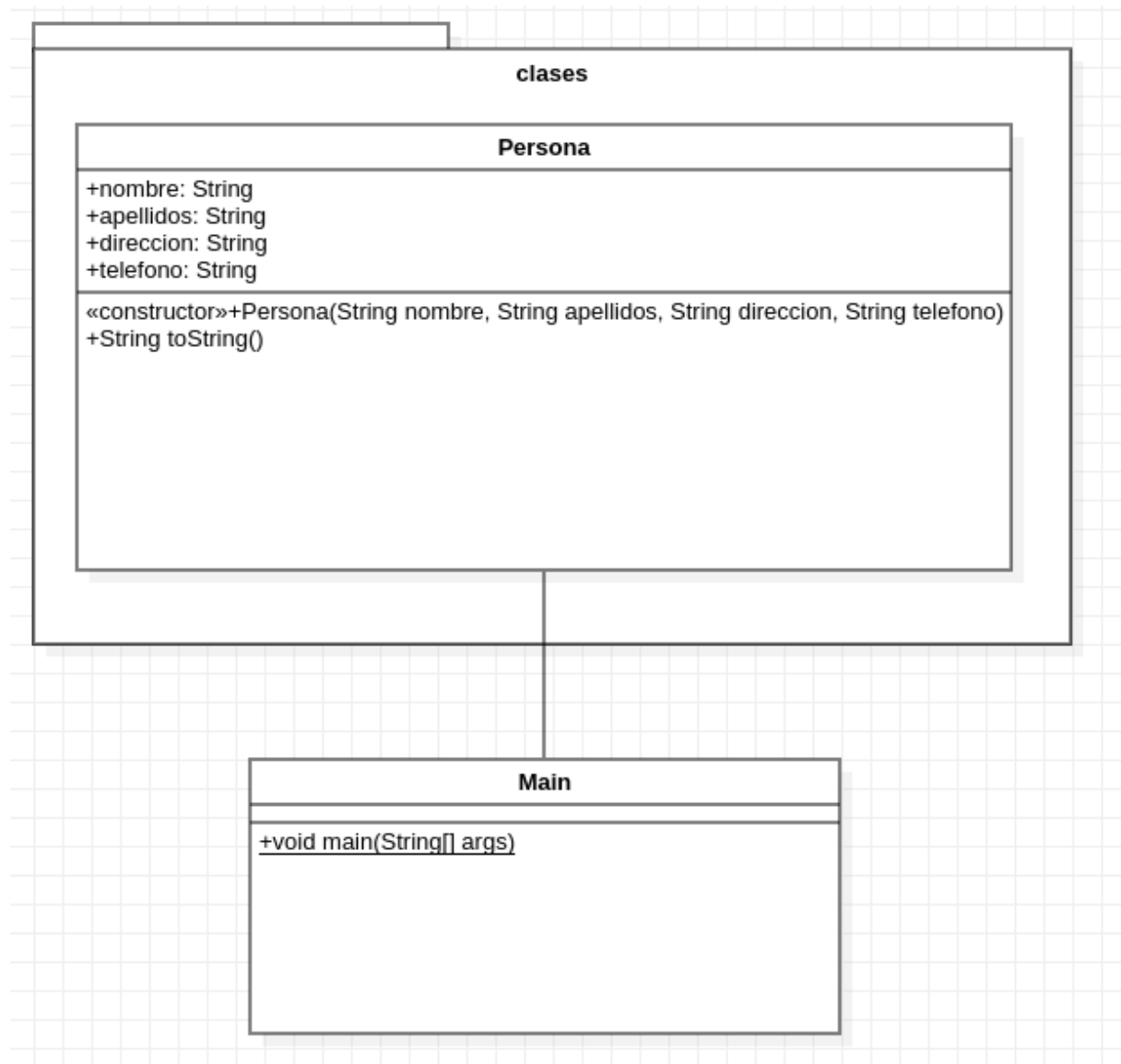
        DefaultListModel<Persona> personas = new DefaultListModel<Persona>();
        window.lista_personas.setModel(personas);

        window.personas = personas;
        window.add_window.personas = personas;

    }
}

```

## Diagrama de clases (UML)



## Código en Github:

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4/ejercicio5](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4/ejercicio5)

## Código en Github con todos los ejercicios:

[https://github.com/jmcada2503/ejercicios\\_poo/tree/master/actividad4](https://github.com/jmcada2503/ejercicios_poo/tree/master/actividad4)