

## **Actividad 4**

### **Programación Orientada A Objetos**

#### **Estudiantes:**

Cristian Emilio Velásquez Lora

Leandro Cardona Rios

Manuel Restrepo Sierra

#### **Grupo:**

6

#### **Semestre:**

2024-2

**Universidad Nacional De Colombia**

**Medellin**

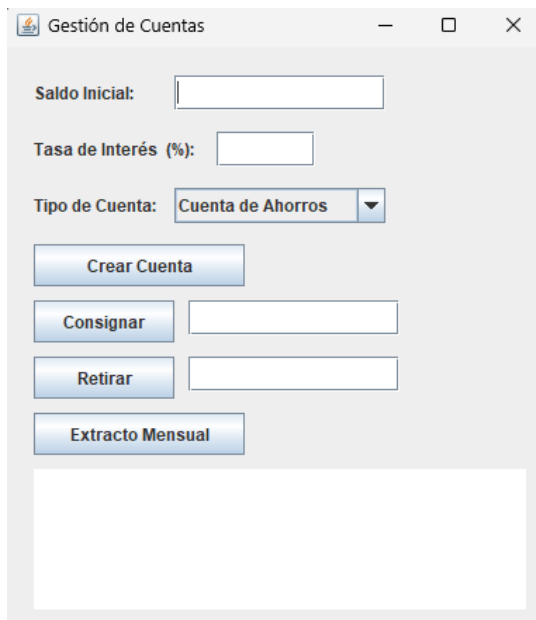
**2025**

Repositorio GitHub Completo: <https://bit.ly/3wx0y9B>

Texto: Ejercicios de Programación Orientada a Objetos con Java y UML – Leonardo Bermón Angarita – Universidad Nacional de Colombia

URL archivo “Activity Codes” en GitHub: <https://bit.ly/4ghWAmP>

## Ejercicio 4.1 Herencia



```
package com.mycompany.CuentaBancaria;

import java.util.Scanner;

public class PruebaCuenta {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Cuenta de ahorros");
        System.out.print("Ingrese saldo inicial: $");
        float saldoInicialAhorros = input.nextFloat();
        System.out.print("Ingrese tasa de interés: ");
        float tasaAhorros = input.nextFloat();

        CuentaAhorros cuental = new CuentaAhorros(saldoInicialAhorros, tasaAhorros);

        System.out.print("Ingresar cantidad a consignar: $");
        float cantidadDepositar = input.nextFloat();
        cuental.consignar(cantidadDepositar);

        System.out.print("Ingresar cantidad a retirar: $");
        float cantidadRetirar = input.nextFloat();
        cuental.retirar(cantidadRetirar);

        cuental.extractoMensual();
        cuental.imprimir();

        input.close();
    }
}
```

```

package com.mycompany.CuentaBancaria;

public class Cuenta {
    protected float saldo;
    protected int numeroConsignaciones = 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 consignar(float cantidad) {
        saldo += cantidad;
        numeroConsignaciones++;
    }

    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)/100;
        float interesMensual = saldo * tasaMensual;
        saldo += interesMensual;
    }

    public void extractoMensual() {
        saldo -= comisionMensual;
        calcularInteres();
    }
}

```

```

package com.mycompany.CuentaBancaria;

public class CuentaAhorros extends Cuenta {
    private boolean activa;

    public CuentaAhorros(float saldo, float tasa) {
        super(saldo, tasa);
        activa = saldo >= 10000;
    }

    @Override
    public void retirar(float cantidad) {
        if (activa) {
            super.retirar(cantidad);
        }
    }

    @Override
    public void consignar(float cantidad) {
        if (activa) {
            super.consignar(cantidad);
        }
    }

    @Override
    public void extractoMensual() {
        if (numeroRetiros > 4) {
            comisionMensual += (numeroRetiros - 4) * 1000;
        }
        super.extractoMensual();
        activa = saldo >= 10000;
    }

    public void imprimir() {
        System.out.println("Saldo = $ " + saldo);
        System.out.println("Comisión mensual = $ " + comisionMensual);
        System.out.println("Número de transacciones = " + (numeroConsignaciones + numeroRetiros));
    }
}

```

```

package com.mycompany.CuentaBancaria;

public class CuentaCorriente extends Cuenta {
    float sobregiro;

    public CuentaCorriente(float saldo, float tasa) {
        super(saldo, tasa);
        sobregiro = 0;
    }

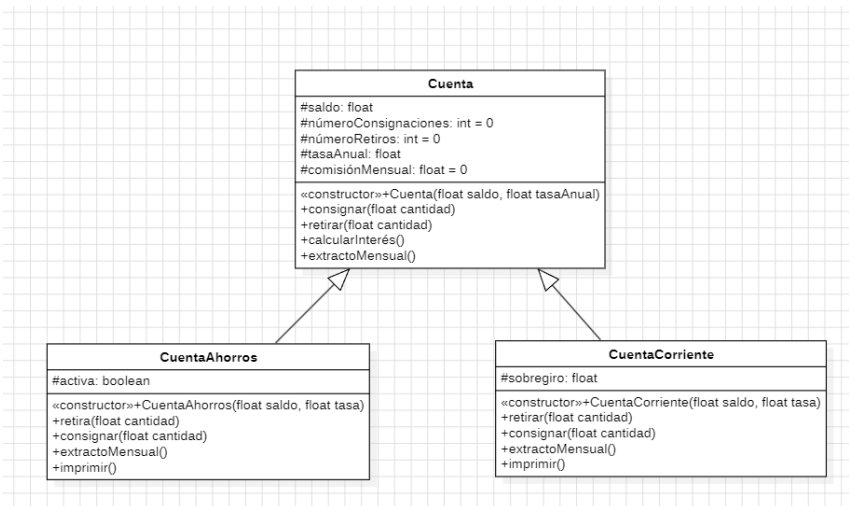
    @Override
    public void retirar(float cantidad) {
        float resultado = saldo - cantidad;
        if (resultado < 0) {
            sobregiro -= resultado;
            saldo = 0;
        } else {
            super.retirar(cantidad);
        }
    }

    @Override
    public void consignar(float cantidad) {
        if (sobregiro > 0) {
            if (sobregiro > cantidad) {
                sobregiro -= cantidad;
            } else {
                saldo += (cantidad - sobregiro);
                sobregiro = 0;
            }
        } else {
            super.consignar(cantidad);
        }
    }

    @Override
    public void extractoMensual() {
        super.extractoMensual();
    }

    public void imprimir() {
        System.out.println("Saldo = $ " + saldo);
        System.out.println("Comisión mensual = $ " + comisionMensual);
        System.out.println("Número de transacciones = " + (numeroConsignaciones + numeroRetiros));
        System.out.println("Valor de sobregiro = $ " + sobregiro);
    }
}

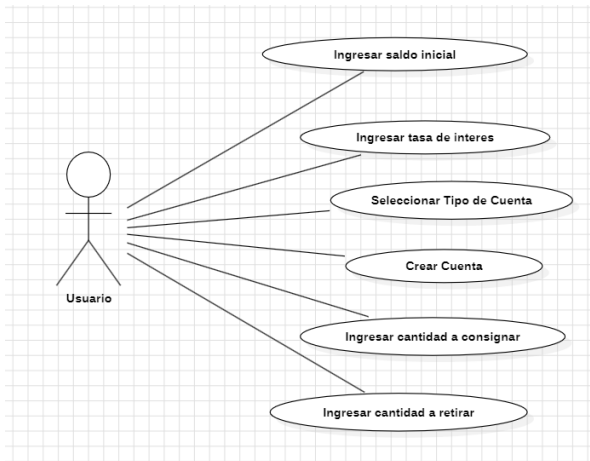
```



```

cuenta1: CuentaAhorro
#numeroConsignaciones = 0
#numeroRetiros = 0
#comisiónMensual = 0

```



## Ejercicio 4.2 Paquetes y método de acceso

Gestión de Inmuebles

Casa Rural

Identificador inmobiliario:

Área (m2):

Dirección:

Número de habitaciones:

Número de baños:

Número de pisos:

Distancia a la cabecera municipal (km):

Altitud sobre el nivel del mar (m):

Calcular Precio Venta Precio de venta: \$0

```

package com.mycompany.inmuebles;

public class PruebaInmuebles {
    public static void main(String[] args) {
        // Create an instance of ApartamentoFamiliar
        ApartamentoFamiliar aptol = new ApartamentoFamiliar(103067, 120, "Avenida Santander 45-45", 3, 2, 200000);
        System.out.println("Datos apartamento familiar:");
        aptol.calcularPrecioVenta(aptol.valorArea);
        aptol.imprimir();

        // Create an instance of Apartaestudio
        Apartaestudio aptestudiol = new Apartaestudio(12354, 50, "Avenida Caracas 30-15");
        aptestudiol.calcularPrecioVenta(aptestudiol.valorArea);
        aptestudiol.imprimir();
    }
}
  
```

```
package com.mycompany.Inmuebles;
```

```
public class Apartaestudio extends Apartamento {  
    protected static double valorArea = 1500000;
```

```
    public Apartaestudio(int identificadorInmobiliario, int area, String direccion) {  
        super(identificadorInmobiliario, area, direccion, 1, 1); // 1 bedroom, 1 bathroom  
    }
```

```
    void imprimir() {  
        super.imprimir();  
    }
```

```
}
```

```
package com.mycompany.Inmuebles;
```

```
public class Apartamento extends InmuebleVivienda {
```

```
    public Apartamento(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños) {  
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños);  
    }
```

```
    void imprimir() {  
        super.imprimir();  
    }
```

```
}
```

```
package com.mycompany.Inmuebles;
```

```
public class ApartamentoFamiliar extends Apartamento {  
    protected static double valorArea = 2000000;  
    protected int valorAdministracion;
```

```
    public ApartamentoFamiliar(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños) {  
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños);  
        this.valorAdministracion = valorAdministracion;  
    }
```

```
    void imprimir() {  
        super.imprimir();  
        System.out.println("Valor de la administración = $" + valorAdministracion);  
    }
```

```
}
```

```
package com.mycompany.Inmuebles;
```

```
public class Casa extends InmuebleVivienda {  
    protected int numeroPisos;
```

```
    public Casa(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños, int numeroPisos) {  
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños);  
        this.numeroPisos = numeroPisos;  
    }
```

```
    void imprimir() {  
        super.imprimir();  
        System.out.println("Número de pisos = " + numeroPisos);  
    }
```

```
}
```

```

package com.mycompany.Inmuebles;

public class CasaConjuntoCerrado extends Casa {
    protected static double valorArea = 2500000;
    protected int valorAdministracion;
    protected boolean tienePiscina;
    protected boolean tieneCamposDeportivos;

    public CasaConjuntoCerrado(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numero
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños, numeroPisos);
        this.valorAdministracion = valorAdministracion;
        this.tienePiscina = tienePiscina;
        this.tieneCamposDeportivos = tieneCamposDeportivos;
    }

    void imprimir() {
        super.imprimir();
        System.out.println("Valor de la administración = $" + valorAdministracion);
        System.out.println("Tiene piscina? = " + tienePiscina);
        System.out.println("Tiene campos deportivos? = " + tieneCamposDeportivos);
    }
}

```

```

package com.mycompany.Inmuebles;

public class CasaIndependiente extends CasaUrbana {
    protected static double valorArea = 3000000;

    public CasaIndependiente(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños, numeroPisos);
    }

    void imprimir() {
        super.imprimir();
    }
}

```

```

package com.mycompany.Inmuebles;

public class CasaRural extends Casa {
    protected static double valorArea = 1500000;
    protected int distanciaCabera;
    protected int altitud;

    public CasaRural(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños, int
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños, numeroPisos);
        this.distanciaCabera = distanciaCabera;
        this.altitud = altitud;
    }

    void imprimir() {
        super.imprimir();
        System.out.println("Distancia a la cabecera municipal = " + distanciaCabera + " km.");
        System.out.println("Altitud sobre el nivel del mar = " + altitud + " metros.");
    }
}

```

```

package com.mycompany.Inmuebles;

public class CasaUrbana extends Casa {
    public CasaUrbana(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños, in
        super(identificadorInmobiliario, area, direccion, numeroHabitaciones, numeroBaños, numeroPisos);
    }

    void imprimir() {
        super.imprimir();
    }
}

```

```

package com.mycompany.Inmuebles;

public class Inmueble {
    protected int identificadorInmobiliario;
    protected int area;
    protected String direccion;
    protected double precioVenta;

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

    double calcularPrecioVenta(double valorArea) {
        precioVenta = area * valorArea;
        return precioVenta;
    }

    void imprimir() {
        System.out.println("Identificador inmobiliario = " + identificadorInmobiliario);
        System.out.println("Área = " + area);
        System.out.println("Dirección = " + direccion);
        System.out.println("Precio de venta = $" + precioVenta);
    }
}

```

```

package com.mycompany.Inmuebles;

public class InmuebleVivienda extends Inmueble {
    protected int numeroHabitaciones;
    protected int numeroBaños;

    public InmuebleVivienda(int identificadorInmobiliario, int area, String direccion, int numeroHabitaciones, int numeroBaños) {
        super(identificadorInmobiliario, area, direccion);
        this.numeroHabitaciones = numeroHabitaciones;
        this.numeroBaños = numeroBaños;
    }

    void imprimir() {
        super.imprimir();
        System.out.println("Número de habitaciones = " + numeroHabitaciones);
        System.out.println("Número de baños = " + numeroBaños);
    }
}

```

```

package com.mycompany.Inmuebles;

public class Local extends Inmueble {
    enum tipo {INTERNO, CALLE};
    protected tipo tipoLocal;

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

    void imprimir() {
        super.imprimir();
        System.out.println("Tipo de local = " + tipoLocal);
    }
}

```



```

package com.mycompany.Inmuebles;

public class LocalComercial extends Local {
    protected static double valorArea = 3000000;
    protected String centroComercial;

    public LocalComercial(int identificadorInmobiliario, int area, String direccion, tipo tipoLocal, String centroComercial) {
        super(identificadorInmobiliario, area, direccion, tipoLocal);
        this.centroComercial = centroComercial;
    }

    void imprimir() {
        super.imprimir();
        System.out.println("Centro comercial = " + centroComercial);
    }
}

```

```

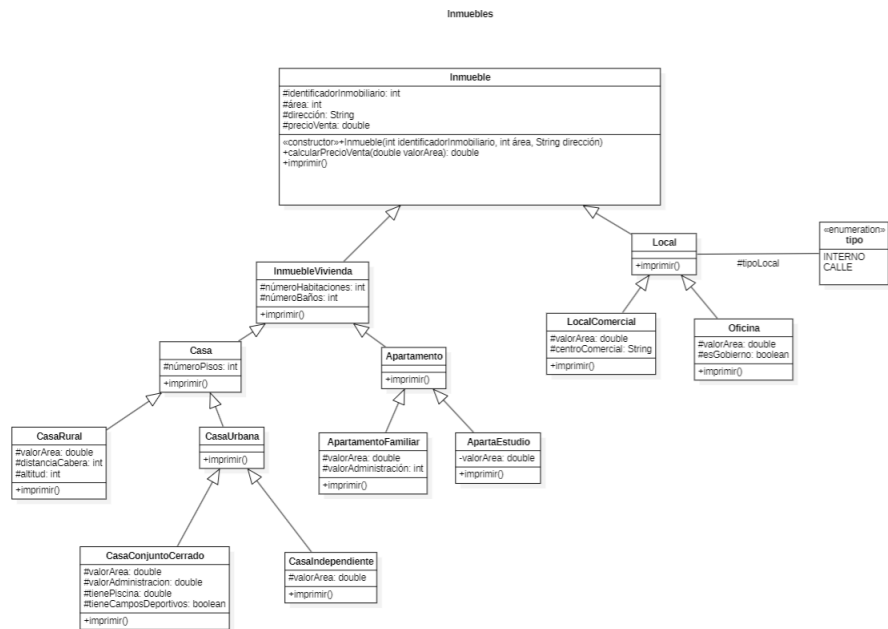
package com.mycompany.Inmuebles;

public class Oficina extends Local {
    protected static double valorArea = 3500000;
    protected boolean esGobierno;

    public Oficina(int identificadorInmobiliario, int area, String direccion, tipo tipoLocal, boolean esGobierno) {
        super(identificadorInmobiliario, area, direccion, tipoLocal);
        this.esGobierno = esGobierno;
    }

    void imprimir() {
        super.imprimir();
        System.out.println("Es oficina gubernamental = " + esGobierno);
    }
}

```

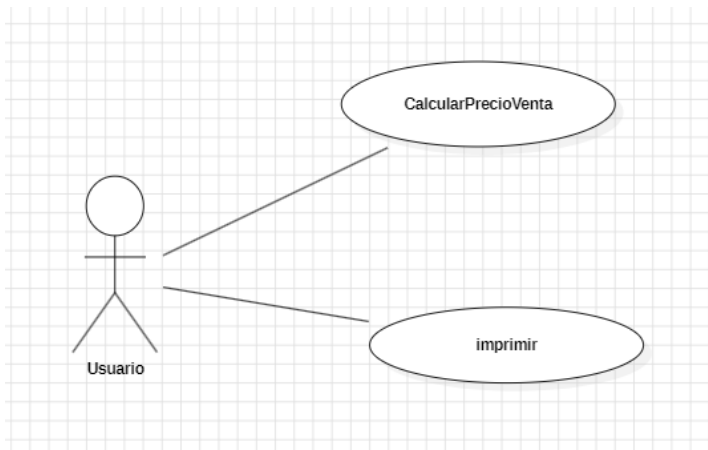


**apto1: ApartamentoFamiliar**

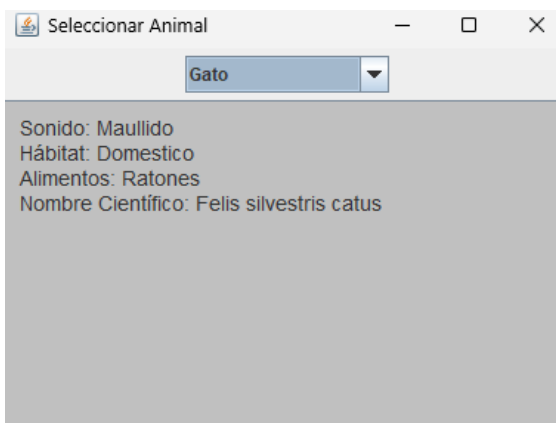
#identificadorInmobiliario = 103067  
 #area = 120  
 #direccion = "Avenida Santander 45-45"  
 #numeroHabitaciones = 3  
 #numeroBaños = 2  
 #valorAdministración = 200000

**aptestudio: Apartaestudio**

#identificadorInmobiliario = 12354  
 #area = 50  
 #direccion = "Avenida Caracas 30-15"  
 #numeroHabitaciones = 1  
 #numeroBaños = 2



## Ejercicio 4.7 Clases abstractas



```
package com.mycompany.animales;

public class PruebaAnimales {

    public static void main(String[] args) {
        Animal[] animales = new Animal[4];
        animales[0] = new Gato();
        animales[1] = new Perro();
        animales[2] = new Lobo();
        animales[3] = new Leon();

        for(int i = 0; i < animales.length; i++) {
            System.out.println(animales[i].getnombreCientifico());
            System.out.println("Sonido: " + animales[i].getSonido());
            System.out.println("Alimentos: " + animales[i].getAlimentos());
            System.out.println("Habitat: " + animales[i].getHabitat());
            System.out.println();
        }
    }
}
```

```
package com.mycompany.animales;

public abstract class Animal {
    protected String sonido;
    protected String Habitat;
    protected String Alimentos;
    protected String nombreCientifico;

    public abstract String getSonido();
    public abstract String getHabitat();
    public abstract String getAlimentos();
    public abstract String getnombreCientifico();
}
```

```
package com.mycompany.animales;

public abstract class Canido extends Animal {
    // Puede contener métodos específicos para los canidos, si es necesario
}
```

```
package com.mycompany.animales;

public abstract class Felino extends Animal {
    // Puede contener métodos específicos para los felinos, si es necesario
}
```

```
package com.mycompany.animales;

public class Gato extends Felino {
    public String getSonido() {
        return "Maullido";
    }
    public String getAlimentos() {
        return "Ratones";
    }
    public String getHabitat() {
        return "Domestico";
    }
    public String getnombreCientifico() {
        return "Felis silvestris catus";
    }
}
```

```
package com.mycompany.animales;
```

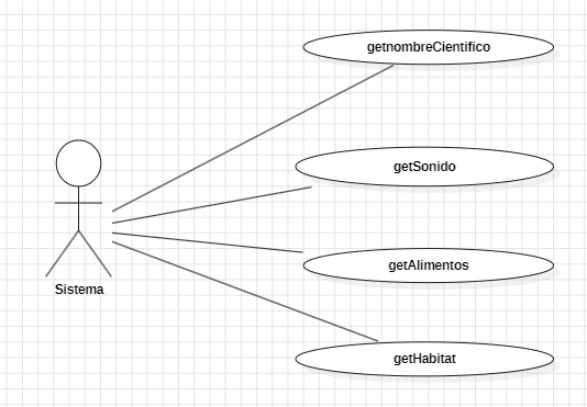
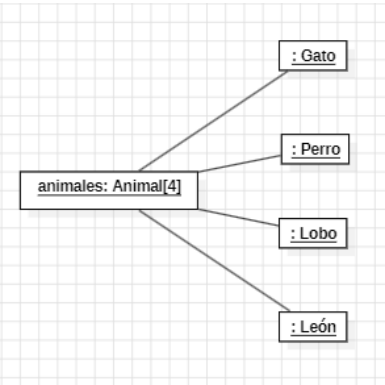
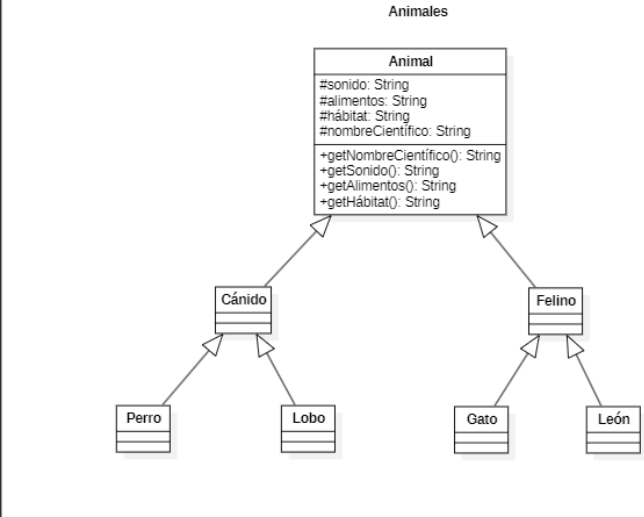
```
public class Leon extends Felino {  
    public String getSonido() {  
        return "Rugido";  
    }  
    public String getAlimentos() {  
        return "Carnivoro";  
    }  
    public String getHabitat() {  
        return "Praderas";  
    }  
    public String getnombreCientifico() {  
        return "Panthera Leo";  
    }  
}
```

```
package com.mycompany.animales;
```

```
public class Lobo extends Canido {  
    public String getSonido() {  
        return "Aullido";  
    }  
    public String getAlimentos() {  
        return "Carnivoro";  
    }  
    public String getHabitat() {  
        return "Bosque";  
    }  
    public String getnombreCientifico() {  
        return "Canis Lupus";  
    }  
}
```

```
package com.mycompany.animales;
```

```
public class Perro extends Canido {  
    public String getSonido() {  
        return "Ladrido";  
    }  
    public String getAlimentos() {  
        return "Carnivoro";  
    }  
    public String getHabitat() {  
        return "Domestico";  
    }  
    public String getnombreCientifico() {  
        return "Canis lupus familiaris";  
    }  
}
```



### Ejercicio 4.8 Métodos abstractos



```
package com.mycompany.carreraciclistica;

public class PruebaCarreraCiclistica {

    public static void main(String[] args) {
        // Equipo 1
        Equipo equipol = new Equipo("Ineos Grenadiers", "Reino Unido");
        Velocista velocistal = new Velocista(123979, "Geraint Thomas", 320, 25);
        Escalador escalador1 = new Escalador(123980, "Egan Bernal", 25, 10);
        Contrarrelojista contrarrelojista1 = new Contrarrelojista(123981, "Jonathan Castroviejo", 120);

        equipol.añadirCiclista(velocistal);
        equipol.añadirCiclista(escalador1);
        equipol.añadirCiclista(contrarrelojista1);

        velocistal.setTiempoAcumulado(365);
        escalador1.setTiempoAcumulado(385);
        contrarrelojista1.setTiempoAcumulado(370);

        equipol.calcularTotalTiempo();
        equipol.imprimir();
        equipol.listarEquipo();
    }
}
```

```
package com.mycompany.carreraciclistica;
```

```
public abstract class Ciclista {  
    private int identificador;  
    private String nombre;  
    private int tiempoAcumulado = 0;  
  
    public Ciclista(int identificador, String nombre) {  
        this.identificador = identificador;  
        this.nombre = nombre;  
    }  
  
    abstract String imprimirTipo();  
  
    protected int getIdentificador() {  
        return identificador;  
    }  
  
    protected void setIdentificador(int identificador) {  
        this.identificador = identificador;  
    }  
  
    protected String getNombre() {  
        return nombre;  
    }  
  
    protected void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    protected int getPosicionGeneral(int posicionGeneral) {  
        return posicionGeneral;  
    }  
  
    protected void setPosicionGeneral(int posicionGeneral) {  
        // No se usa, ya que este campo se calcula en base al tiempo  
    }  
  
    protected int getTiempoAcumulado() {  
        return tiempoAcumulado;  
    }  
  
    protected void setTiempoAcumulado(int tiempoAcumulado) {  
        this.tiempoAcumulado = tiempoAcumulado;  
    }  
  
    protected void imprimir() {  
        System.out.println("Identificador = " + identificador);  
        System.out.println("Nombre = " + nombre);  
        System.out.println("Tiempo Acumulado = " + tiempoAcumulado);  
    }  
}
```

```
package com.mycompany.carreraciclistica;
```

```
public class Contrarrelojista extends Ciclista {  
    private double velocidadMaxima;
```

```
    public Contrarrelojista(int identificador, String nombre, double velocidadMaxima) {  
        super(identificador, nombre);  
        this.velocidadMaxima = velocidadMaxima;  
    }
```

```
    protected double getVelocidadMaxima() {  
        return velocidadMaxima;  
    }
```

```
    protected void setVelocidadMaxima(double velocidadMaxima) {  
        this.velocidadMaxima = velocidadMaxima;  
    }
```

```
    protected void imprimir() {  
        super.imprimir();  
        System.out.println("Velocidad máxima = " + velocidadMaxima);  
    }
```

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



```
package com.mycompany.carreraciclistica;
import java.util.*;

public class Equipo {
    private String nombre;
    private static double totalTiempo;
    private String pais;
    Vector listaCiclistas;

    public Equipo(String nombre, String pais) {
        this.nombre = nombre;
        this.pais = pais;
        totalTiempo = 0;
        listaCiclistas = new Vector();
    }

    public String getNombre() {
        return nombre;
    }

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

    private String getPais() {
        return pais;
    }

    private void setPais(String pais) {
        this.pais = pais;
    }

    void añadirCiclista(Ciclista ciclista) {
        listaCiclistas.add(ciclista);
    }

    void listarEquipo() {
        for (int i = 0; i < listaCiclistas.size(); i++) {
            Ciclista c = (Ciclista) listaCiclistas.elementAt(i);
            System.out.println(c.getNombre());
        }
    }
}
```

```
void buscarCiclista() {
    Scanner sc = new Scanner(System.in);
    String nombreCiclista = sc.next();
    for (int i = 0; i < listaCiclistas.size(); i++) {
        Ciclista c = (Ciclista) listaCiclistas.elementAt(i);
        if (c.getNombre().equals(nombreCiclista)) {
            System.out.println(c.getNombre());
        }
    }
}

void calcularTotalTiempo() {
    for (int i = 0; i < listaCiclistas.size(); i++) {
        Ciclista c = (Ciclista) listaCiclistas.elementAt(i);
        totalTiempo = totalTiempo + c.getTiempoAcumulado();
    }
}

void imprimir() {
    System.out.println("Nombre del equipo = " + nombre);
    System.out.println("Pais = " + pais);
    System.out.println("Total tiempo del equipo = " + totalTiempo);
}
}
```

```
package com.mycompany.carreraciclistica;
```

```
public class Escalador extends Ciclista {  
    private double aceleracionPromedio;  
    private double gradoRampa;  
  
    public Escalador(int identificador, String nombre, double aceleracionPromedio, double gradoRampa) {  
        super(identificador, nombre);  
        this.aceleracionPromedio = aceleracionPromedio;  
        this.gradoRampa = gradoRampa;  
    }  
  
    protected double getAceleracionPromedio() {  
        return aceleracionPromedio;  
    }  
  
    protected void setAceleracionPromedio(double aceleracionPromedio) {  
        this.aceleracionPromedio = aceleracionPromedio;  
    }  
  
    protected double getGradoRampa() {  
        return gradoRampa;  
    }  
  
    protected void setGradoRampa(double gradoRampa) {  
        this.gradoRampa = gradoRampa;  
    }  
  
    protected void imprimir() {  
        super.imprimir();  
        System.out.println("Aceleracion promedio = " + aceleracionPromedio);  
        System.out.println("Grado de rampa = " + gradoRampa);  
    }  
  
    protected String imprimirTipo() {  
        return "Es un escalador";  
    }  
}
```

```

package com.mycompany.carreraciclistica;

public class Velocista extends Ciclista {
    private double potenciaPromedio;
    private double velocidadPromedio;

    public Velocista(int identificador, String nombre, double potenciaPromedio, double velocidadPromedio) {
        super(identificador, nombre);
        this.potenciaPromedio = potenciaPromedio;
        this.velocidadPromedio = velocidadPromedio;
    }

    protected double getPotenciaPromedio() {
        return potenciaPromedio;
    }

    protected void setPotenciaPromedio(double potenciaPromedio) {
        this.potenciaPromedio = potenciaPromedio;
    }

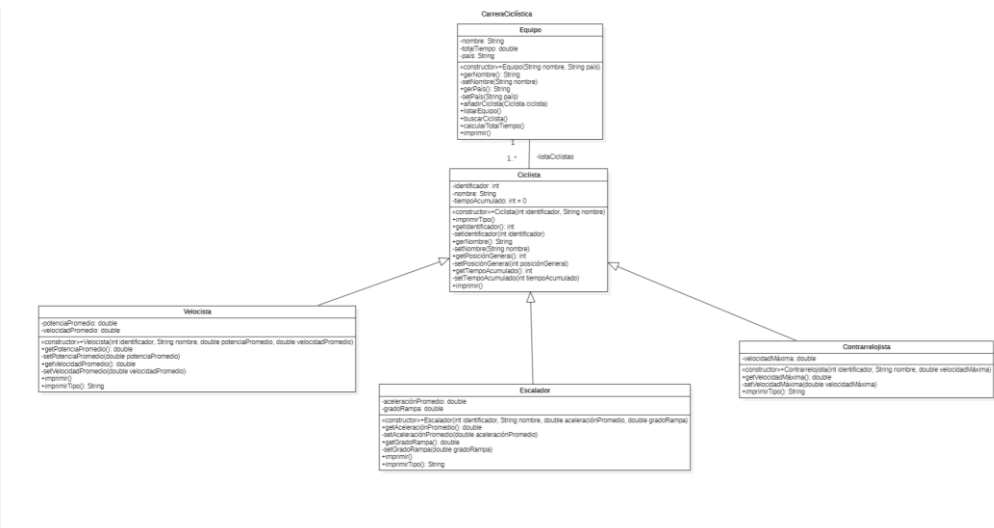
    protected double getVelocidadPromedio() {
        return velocidadPromedio;
    }

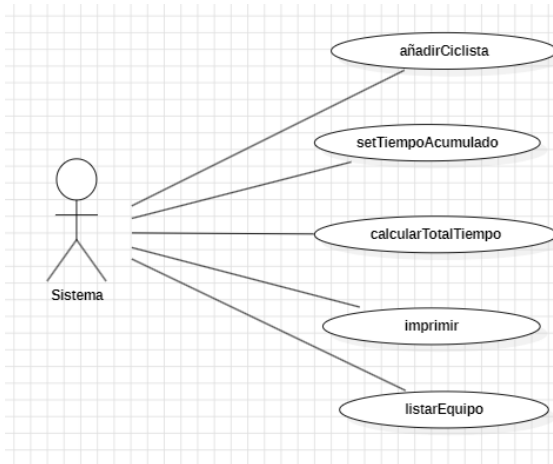
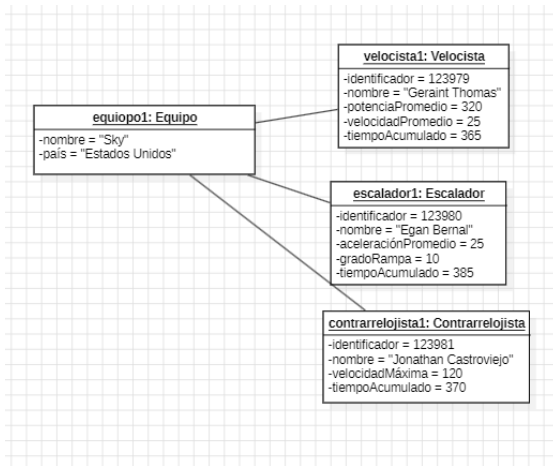
    protected void setVelocidadPromedio(double velocidadPromedio) {
        this.velocidadPromedio = velocidadPromedio;
    }

    protected void imprimir() {
        super.imprimir();
        System.out.println("Potencia promedio = " + potenciaPromedio);
        System.out.println("Velocidad promedio = " + velocidadPromedio);
    }

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

```





## Ejercicio 8.1

Personas

Nombre:

Apellidos:

Teléfono:

Dirección:

Añadir

Eliminar

Borrar Lista

```
package com.mycompany.datospersona;
```

```
public class Principal {  
    public static void main(String[] args) {  
        VentanaPrincipal miVentanaPrincipal;  
        miVentanaPrincipal = new VentanaPrincipal();  
        miVentanaPrincipal.setVisible(true);  
    }  
}
```

```
package com.mycompany.datospersona;
```

```
import java.util.*;
```

```
public class ListaPersonas {  
    Vector listaPersonas;  
  
    public ListaPersonas() {  
        listaPersonas = new Vector();  
    }  
  
    public void añadirPersona(Persona p) {  
        listaPersonas.add(p);  
    }  
  
    public void eliminarPersona(int i) {  
        listaPersonas.removeElementAt(i);  
    }  
  
    public void borrarLista() {  
        listaPersonas.removeAllElements();  
    }  
}
```

```
package com.mycompany.datospersona;
```

```
public class Persona {  
    String nombre;  
    String apellidos;  
    String teléfono;  
    String dirección;  
  
    public Persona(String nombre, String apellidos, String teléfono, String dirección) {  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
        this.teléfono = teléfono;  
        this.dirección = dirección;  
    }  
}
```

