














# Trabajo Grupal 2

 Luisa Fernanda Mazo Perez	 <a href="mailto:lmazo@unal.edu.co">lmazo@unal.edu.co</a>	 1001233955
 Juan Esteban Yela	 <a href="mailto:jjelab@unal.edu.co">jjelab@unal.edu.co</a>	 1193559688
 Daniel Castillo Giraldo	 <a href="mailto:dcastillogi@unal.edu.co">dcastillogi@unal.edu.co</a>	 1002592605

 <https://bit.ly/TrabajoGrupalGithub>

 [https://drive.google.com/drive/folders/1DQT\\_6NXSCvHEskmMvr7vBgryijUgvRhW?usp=sharing](https://drive.google.com/drive/folders/1DQT_6NXSCvHEskmMvr7vBgryijUgvRhW?usp=sharing)

## Tabla de Contenidos

Tabla de Contenidos

Ejercicio 1

Código

Cuenta

CuentaAhorros

CuentaCorriente

Ejercicio1

Diagrama de Clases

Ejercicio 2

Código

Apartaestudio

Apartamentos

Casas

Comerciales

ConjuntoCerrado

Familiar

Independiente

Inmuebles

Locales

Main

Oficinas

Rurales

Urbanas

Vivivenda

Diagrama de Clases

Ejercicio 3

Código

Animal

Canidos

Felinos

Gatos

Leones

Lobos

[Perros](#)  
[Prueba](#)  
[Diagrama de Clases](#)  
[Ejercicio 4](#)  
[Código](#)  
[Ciclista](#)  
[ContraReloj](#)  
[Ejercicio4](#)  
[Equipo](#)  
[Escalador](#)  
[Velocista](#)  
[Diagrama de Clases](#)  
[Ejercicio 5](#)  
[Código](#)  
[Main](#)  
[Persona](#)  
[UI](#)  
[Diagrama de Clases](#)  
[Interfaz Gráfica](#)

---

## Ejercicio 1



[https://github.com/danielcgiraldo/POO\\_Grupal/tree/main/Trabajo\\_2/Ejercicio1/src/main/java/Ejercicio1](https://github.com/danielcgiraldo/POO_Grupal/tree/main/Trabajo_2/Ejercicio1/src/main/java/Ejercicio1)



<https://drive.google.com/file/d/1XYMJ2QD15GgsL1IqQHkdHubCjzWQpImi/view?usp=sharing>

## Código

### Cuenta

```
package Ejercicio1;

public class Cuenta {

    //Ejercicio herencia
    //Clase cuenta
    protected float saldo;
    protected int numConsignacion = 0;
    protected int numRetiros = 0;
    protected float tasaAnual; //porcentaje
    protected float comisionMensual;

    //constructor
    public Cuenta(float saldoC, float tasaAnualC){
        saldo = saldoC;
        tasaAnual = tasaAnualC;
    }

    //metodos de la clase cuenta
    public void Consignar(float cantidad) {
        saldo = saldo + cantidad;
        numConsignacion = numConsignacion + 1;
    }

    public void Retirar(float cantidad) {
```

```

        if (cantidad <= saldo) {
            saldo = saldo - cantidad;
            numRetiros = numRetiros + 1;
        } else {
            System.out.println("No es posible retirar el dinero");
        }
    }

    public void CalcInteresMensual() {
        float tasamensual = tasaAnual / 12;

        float interesmensual = saldo * tasamensual;
        saldo += interesmensual;
    }

    public void ExtractoMensual() {
        saldo -= comisionMensual;
        CalcInteresMensual();
    }
}

```

## CuentaAhorros

```

package Ejercicio1;

public class CuentaAhorros extends Cuenta{

    private boolean activa;

    public CuentaAhorros(float saldoC, float tasaC) {
        super(saldoC, tasaC);
        if (saldoC < 10000) {
            activa = false;
        } else {
            activa = true;
        }
    }

    //metodos de clase cuentaAhorros
    public void Consignar(float cantidad) {
        if (activa == true) {
            super.Consignar(cantidad);
        } else {
            System.out.println("Cuenta inactiva");
        }
    }

    public void Retirar(float cantidad) {
        if (activa) {
            super.Retirar(cantidad);
        } else {
            System.out.println("Cuenta inactiva");
        }
    }

    public void extractoMensual() {

        if (saldo < 10000) {
            activa = false;
            System.out.println("Cuenta inactiva");
        } else {
            if (numRetiros > 4) {
                comisionMensual += (numRetiros - 4) * 1000;
            }
            super.ExtractoMensual();
        }
    }
}

```

```

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

```

## CuentaCorriente

```

package Ejercicio1;

public class CuentaCorriente extends Cuenta {

    float sobregiro;

    //constructor cuentacorrente

    public CuentaCorriente(float saldoC, float tasaAnualC) {
        super(saldoC, tasaAnualC);
        sobregiro = 0;
    }

    //metodos de clase cuentaCorriente
    public void Retirar(float cantidad) {
        float resultado = saldo - cantidad;

        if (resultado < 0) {
            sobregiro = sobregiro - resultado;
            saldo = 0;
        } else {
            super.Retirar(cantidad);
        }
    }

    public void Consignar(float cantidad) {
        float residuo = sobregiro - cantidad;

        if (sobregiro > 0) {
            if (residuo > 0) {
                sobregiro = 0;
                saldo = residuo;
            } else {
                sobregiro = -residuo;
                saldo = 0;
            }
        } else {
            super.Consignar(cantidad);
        }
    }

    public void extractoMensual() {

        super.ExtractoMensual();
    }

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

```

## Ejercicio1

```

package Ejercicio1;

import java.util.Scanner;

public class Ejercicio1 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Ingresa tipo de cuenta");
        System.out.println("ingresa 1 : cuenta de ahorros");
        System.out.println("ingresa 2 : cuenta corriente");
        int tipoCuenta = input.nextInt();

        if (tipoCuenta == 1) {
            System.out.println("Cuenta de ahorros");
            System.out.println("Ingresa saldo inicial= $");
            float saldoInicialAhorros = input.nextFloat();
            System.out.print("Ingresa tasa de interés= ");
            float tasaAhorros = input.nextFloat();

            CuentaAhorros cuenta1 = new CuentaAhorros(saldoInicialAhorros, tasaAhorros);

            System.out.print("Ingresar cantidad a consignar: $");
            float cantidadDepositar = input.nextFloat();
            cuenta1.Consignar(cantidadDepositar);

            System.out.print("Ingresar cantidad a retirar: $");
            float cantidadRetirar = input.nextFloat();
            cuenta1.Retirar(cantidadRetirar);
            cuenta1.extractoMensual();
            cuenta1.imprimir();

        }else{
            System.out.println("Cuenta corriente");
            System.out.println("Ingresa saldo inicial= $");
            float saldoInicialAhorros = input.nextFloat();
            System.out.print("Ingresa tasa de interés= ");
            float tasaAhorros = input.nextFloat();

            CuentaCorriente cuenta1 = new CuentaCorriente(saldoInicialAhorros, tasaAhorros);

            System.out.print("Ingresar cantidad a consignar: $");
            float cantidadDepositar = input.nextFloat();
            cuenta1.Consignar(cantidadDepositar);

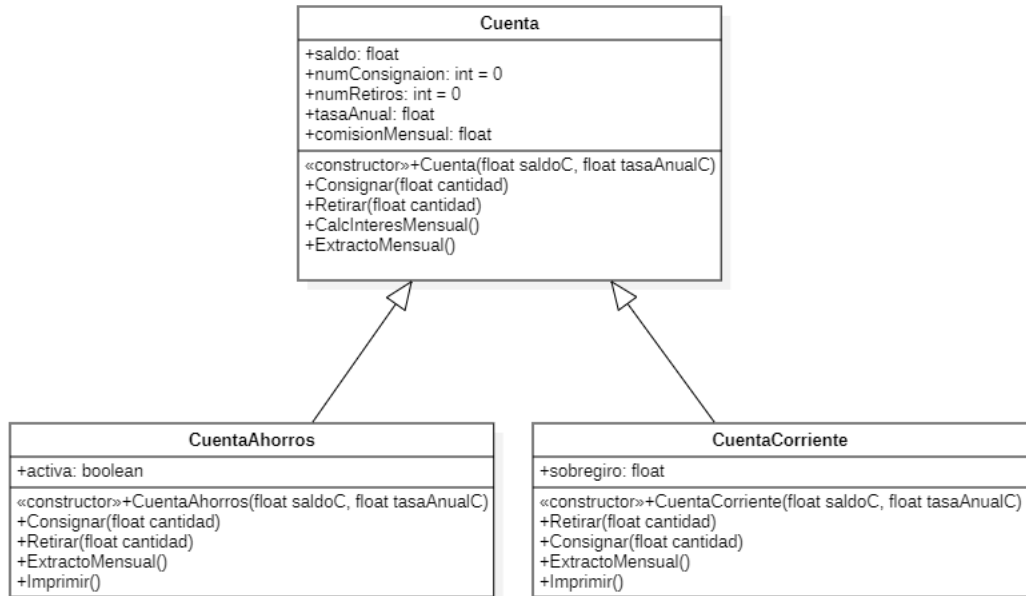
            System.out.print("Ingresar cantidad a retirar: $");
            float cantidadRetirar = input.nextFloat();
            cuenta1.Retirar(cantidadRetirar);
            cuenta1.extractoMensual();
            cuenta1.imprimir();

        }
    }
}

```

## Diagrama de Clases

Diagrama de clases  
Ejercicio 1



## Ejercicio 2



[https://github.com/danielcgiraldo/P00\\_Grupal/tree/main/Trabajo\\_2/Ejercicio2/src/main/java/Inmuebles](https://github.com/danielcgiraldo/P00_Grupal/tree/main/Trabajo_2/Ejercicio2/src/main/java/Inmuebles)



<https://drive.google.com/file/d/1U75Qca87KzTIga-xCCSp1XkQYcPmqxpT/view?usp=sharing>

## Código

### Apartaestudio

```

package Inmuebles;

public class Apartaestudio extends Apartamentos {
    int NumHabitaciones = 1;
    int ValorCompra = 1500000;

    public void imprimir() {
        System.out.println("==== Apartaestudio =====");
        super.imprimir();

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}
  
```

### Apartamentos

```

package Inmuebles;

public class Apartamentos extends Vivienda {
    public void imprimir() {
  
```

```

        super.imprimir();
    }
}

```

## Casas

```

package Inmuebles;

public class Casas extends Vivienda {
    int NumPisos;

    public void imprimir() {
        super.imprimir();
        System.out.println("Número de Pisos: " + this.NumPisos);
    }
}

```

## Comerciales

```

package Inmuebles;

public class Comerciales extends Locales {
    String CentroComercial;
    int ValorCompra = 3000000;

    public void imprimir() {
        System.out.println("==== Locales Comerciales ====");
        super.imprimir();

        System.out.println("Centro Comercial: " + this.CentroComercial);

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## ConjuntoCerrado

```

package Inmuebles;

public class ConjuntoCerrado extends Urbanas {
    int Administracion;
    boolean Piscina;
    boolean CampoDeportivo;
    int ValorCompra = 2500000;

    public void imprimir() {
        System.out.println("==== Casa Conjunto Cerrado ====");
        super.imprimir();

        System.out.println("Valor Administración: " + this.Administracion);
        if(this.Piscina) System.out.println("Piscina: Sí");
        else System.out.println("Piscina: No");
        if(this.CampoDeportivo) System.out.println("Campo Deportivo: Sí");
        else System.out.println("Campo Deportivo: No");

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## Familiar

```

package Inmuebles;

public class Familiar extends Apartamentos {
    int Administracion;
    int ValorCompra = 2000000;
}

```

```

    public void imprimir() {
        System.out.println("==== Apartamento Familiar ====");
        super.imprimir();

        System.out.println("Valor Administración: " + this.Administracion);

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## Independiente

```

package Inmuebles;

public class Independiente extends Urbanas {
    int ValorCompra = 3000000;

    public void imprimir() {
        System.out.println("==== Casa Independiente ====");
        System.out.println("Identificador Inmobiliario: " + this.IdentificadorInmobiliario);
        System.out.println("Área: " + this.Area);
        System.out.println("Dirección: " + this.IdentificadorInmobiliario);
        System.out.println("Número de Habitaciones: " + this.NumHabitaciones);
        System.out.println("Número de Baños: " + this.NumBaños);
        System.out.println("Número de Pisos: " + this.NumPisos);

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## Inmuebles

```

package Inmuebles;

public class Inmuebles {
    int IdentificadorInmobiliario, Area;
    String Direccion;

    protected void imprimir() {
        System.out.println("Identificador Inmobiliario: " + this.IdentificadorInmobiliario);
        System.out.println("Área: " + this.Area);
        System.out.println("Dirección: " + this.IdentificadorInmobiliario);
    }
}

```

## Locales

```

package Inmuebles;

public class Locales extends Inmuebles {
    String Localizacion;
    protected void imprimir() {
        super.imprimir();
    }
}

```

## Main

```

package Inmuebles;

public class Main {
    public static void main(String[] args) {
        // Ejemplo de una casa rural con todos los atributos de todas sus clases heredadas definidos.
        Rurales casarural = new Rurales();
        casarural.IdentificadorInmobiliario = 20;
        casarural.Area = 100;
        casarural.Direccion = "Calle 55 #80-67";
    }
}

```



```

        casarural.NumHabitaciones = 3;
        casarural.NumBanos = 2;
        casarural.NumPisos = 2;
        casarural.DistanciaCabecera = 200;
        casarural.Altitud = 1200;
        casarural.imprimir();
    }
}

```

## Oficinas

```

package Inmuebles;

public class Oficinas extends Locales {
    boolean Gobierno;
    int ValorCompra = 3500000;

    public void imprimir() {
        System.out.println("==== Oficinas ===");
        super.imprimir();

        if(this.Gobierno) System.out.println("Es Oficina Gubernamental: Si");
        else System.out.println("Es Oficina Gubernamental: No");

        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## Rurales

```

package Inmuebles;

public class Rurales extends Casas {
    int DistanciaCabecera;
    int Altitud;
    int ValorCompra = 1500000;

    public void imprimir() {
        System.out.println("==== Casa Rural ===");
        super.imprimir();
        System.out.println("Distancia de la Cabecera Municipal: " + this.DistanciaCabecera);
        System.out.println("Altitud: " + this.Altitud);
        System.out.println("Precio de Venta: " + (this.Area * this.ValorCompra));
    }
}

```

## Urbanas

```

package Inmuebles;

public class Urbanas extends Casas {
    public void imprimir() {
        super.imprimir();
    }
}

```

## Vivienda

```

package Inmuebles;

public class Vivienda extends Inmuebles {
    int NumHabitaciones;
    int NumBanos;

    public void imprimir() {
        super.imprimir();
        System.out.println("Número de Habitaciones: " + this.NumHabitaciones);
    }
}

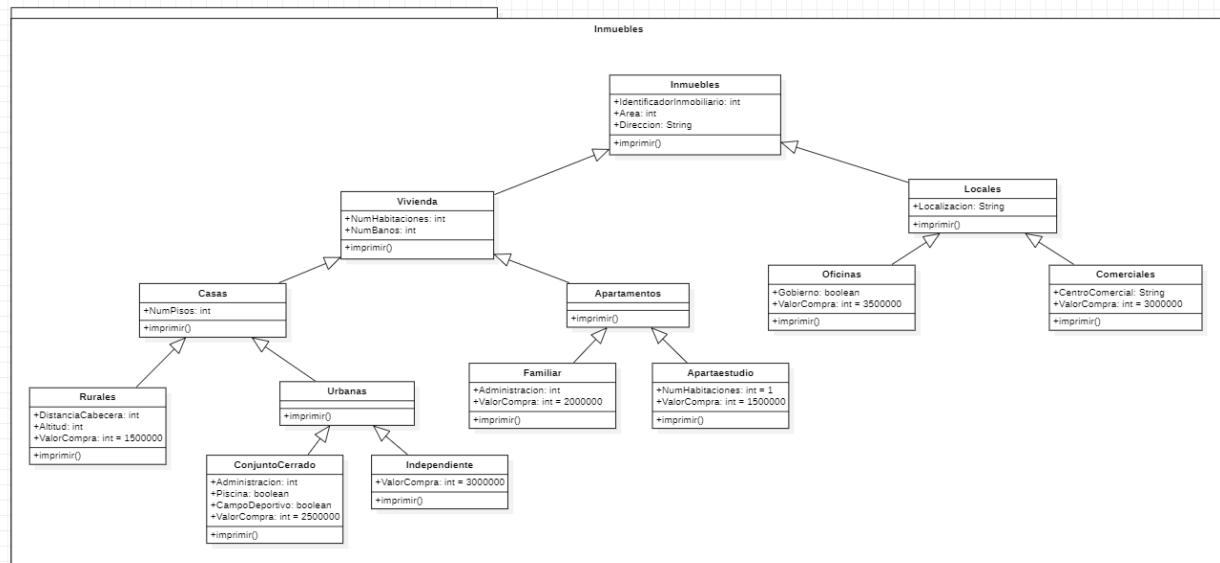
```

```


        System.out.println("Número de Baños: " + this.NumBaños);
    }
}


```

## Diagrama de Clases



## Ejercicio 3

 [https://github.com/danielcgiraldo/P00\\_Grupal/tree/main/Trabajo\\_2/Ejercicio3/src/main/java/Animal](https://github.com/danielcgiraldo/P00_Grupal/tree/main/Trabajo_2/Ejercicio3/src/main/java/Animal)

 <https://drive.google.com/file/d/1NTRwmHjiUAQSzbF6gjqgT8xP190Cdms/view?usp=sharing>

## Código

### Animal

```

package Animal;

public abstract class Animal {

    String sonidos, alimentos, habitat, nombreCientifico;

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

}

```

### Canidos

```
package Animal;

public abstract class Canidos extends Animal {

}
```

## Felinos

```
package Animal;

public abstract class Felinos extends Animal {

}
```

## Gatos

```
package Animal;

public class Gatos extends Felinos {

    public String getSonido() {
        return "Mauullido";
    }

    public String getAlimentos() {
        return "Ratones";
    }

    public String getHabitat() {
        return "Doméstico";
    }

    public String getNombreCientifico() {
        return "Felis silvestris catus";
    }
}
```

## Leones

```
package Animal;

public class Leones extends Felinos {

    public String getSonido() {
        return "Rugido";
    }

    public String getAlimentos() {
        return "Carnívoro";
    }

    public String getHabitat() {
        return "Pradera";
    }

    public String getNombreCientifico() {
        return "Panthera leo";
    }
}
```

## Lobos

```
package Animal;

public class Lobos extends Canidos {

    public String getSonido() {
```

```

        return "Aullido";
    }

    public String getAlimentos() {
        return "Carnívoro";
    }

    public String getHabitat() {
        return "Bosque";
    }

    public String getNombreCientifico() {
        return "Canis lupus";
    }
}

```

## Perros

```

package Animal;

public class Perros extends Canidos {

    public String getSonido() {
        return "Ladrado";
    }

    public String getAlimentos() {
        return "Carnívoro";
    }

    public String getHabitat() {
        return "Doméstico";
    }

    public String getNombreCientifico() {
        return "Canis lupus familiaris";
    }
}

```

## Prueba

```

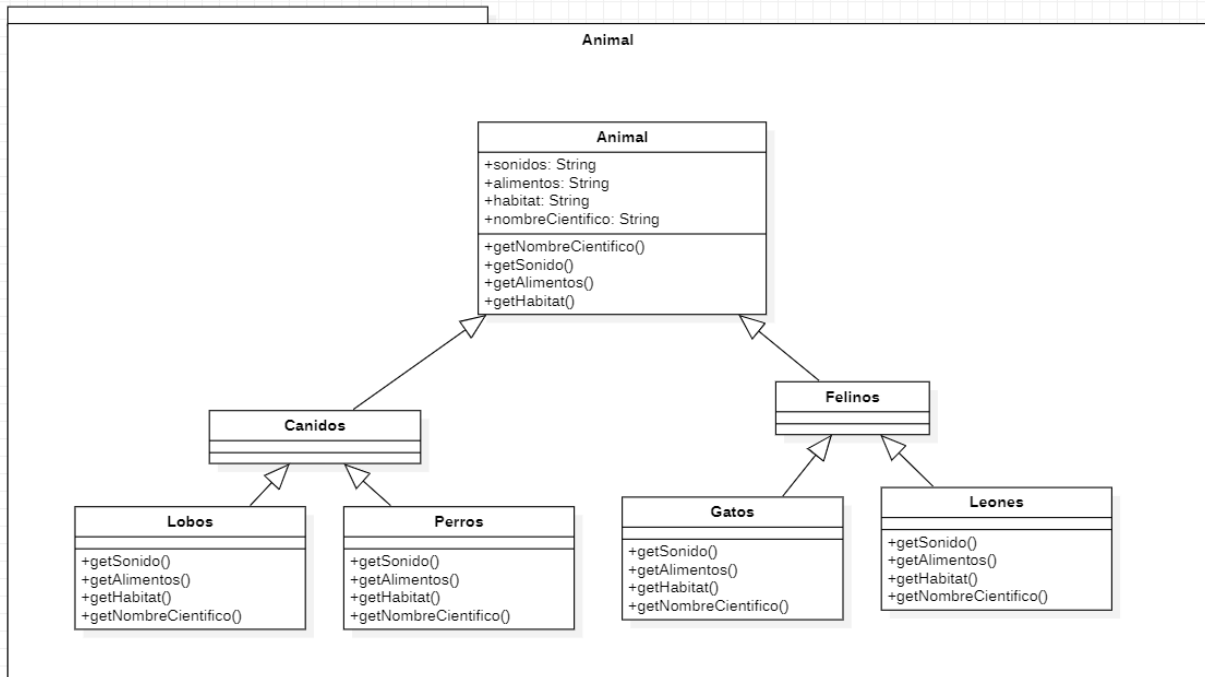
package Animal;

public class Prueba {
    public static void main(String[] args) {
        Animal[] animales = new Animal[4];
        animales[0] = new Gatos();
        animales[1] = new Leones();
        animales[2] = new Lobos();
        animales[3] = new Perros();

        for (int i = 0; i < 4; i++) {
            System.out.println("=====" + ((animales[i].getClass().getName()).split("\\.")[1] + " =====");
            System.out.println("Nombre Científico: " + animales[i].getNombreCientifico());
            System.out.println("Sonido: " + animales[i].getSonido());
            System.out.println("Alimentos: " + animales[i].getAlimentos());
            System.out.println("Hábitat: " + animales[i].getHabitat());
        }
    }
}

```

## Diagrama de Clases



## Ejercicio 4



[https://github.com/danielcgiraldo/P00\\_Grupal/tree/main/Trabajo\\_2/Ejercicio4/src/main/java/Ejercicio4](https://github.com/danielcgiraldo/P00_Grupal/tree/main/Trabajo_2/Ejercicio4/src/main/java/Ejercicio4)



<https://drive.google.com/file/d/1K4CpdanknxUy9nKjFYAVMtXgLjl2aW8R/view?usp=sharing>

## Código

### Ciclista

```

package Ejercicio04;

public abstract class Ciclista {
    //atributos
    private int id;
    private String nombre;
    private int tiempoAcumulado;

    //metodo abstracto
    abstract String ImprimirTipo(String texto);

    //constructor
    public Ciclista(int idC, String nombreC){
        id=idC;
        nombre=nombreC;
        tiempoAcumulado= 0;
    }

    //metodo get
    protected int getId() {
        return id;
    }
  
```

```

    }

    protected String getNombre() {
        return nombre;
    }

    protected int getTiempoCarrera() {
        return tiempoAcumulado;
    }

    //metodo set
    protected void setId(int idC){
        id = idC;
    }

    protected void setNombre(String nombreC){
        nombre = nombreC;
    }

    protected void setTiempoCarrera(int tiempoAcumuladoC){
        tiempoAcumulado = tiempoAcumuladoC;
    }

    //metodo imprimir
    protected void Imprimir(){

        System.out.println("Identificador: " + id);
        System.out.println("Nombre: " + nombre);
        System.out.println("Tiempo acumulado: " + tiempoAcumulado);

    }
}

```

## ContraReloj

```

package Ejercicio4;

public class ContraReloj extends Ciclista {

    private int velMaxima;

    //constructor
    public ContraReloj(int idC, String nombreC,int velmaxC){
        super(idC,nombreC);
        velMaxima=velmaxC;
    }

    //metodo get
    protected int getVelMax() {
        return velMaxima;
    }

    //metodo set
    protected void setVelMax(int velMaxC) {
        velMaxima = velMaxC;
    }

    protected void imprimir() {
        super.Imprimir();
        System.out.println("velocidad maxima: " + velMaxima);
    }

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

    @Override
    String ImprimirTipo(String texto) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/C
    }
}

```

## Ejercicio4

```
package Ejercicio4;

import java.util.Scanner;

public class Ejercicio4 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        Equipo equipo1 = new Equipo("Sky", "Estados Unidos");

        /* System.out.println("Ingresa integrantes de tu equipo!");
        System.out.println("Ingresa 1 : Velocista");
        System.out.println("Ingresa 2 : Escalador");
        System.out.println("Ingresa 3 : Contrarelojista");
        int tipoC = input.nextInt();
        */

        Velocista velocista1 = new Velocista(123979, "Geraint Thomas", 320, 25);
        Escalador escalador1 = new Escalador(123980, "Egan Bernal", 25, 10);
        ContraRelej contrarrelojista1 = new ContraRelej(123981, "Jonathan Castroviejo", 120);
        equipo1.Añadir(velocista1);
        equipo1.Añadir(escalador1);
        equipo1.Añadir(contrarrelojista1);
        velocista1.setTiempoCarrera(365);
        escalador1.setTiempoCarrera(385);
        //ContraRelej.setTiempoCarrera(370);
        equipo1.CalcularTotalTiempo();
        equipo1.Imprimir();
        equipo1.ListaNombres();

    }
}
```

## Equipo

```
package Ejercicio4;

import java.util.Vector;

public class Equipo {
    //atributos
    private String nombreEquipo;
    private static int totalCarrera;
    private String pais;

    Vector integrantes;

    //constructor
    public Equipo(String nombreE, String paisE){
        nombreEquipo=nombreE;
        pais=paisE;

        integrantes = new Vector();
    }

    //metodos get
    protected String getNombre() {
        return nombreEquipo;
    }

    protected int getTotalCarrera() {
        return totalCarrera;
    }

    protected String getPais() {
        return pais;
    }

    //metodos set
```

```

protected void setNombre(String nombreE){
    nombreEquipo = nombreE;

}
protected void setTotalCarrera(int total){
    totalCarrera = total;

}
protected void setPais(String paisE){
    pais=paisE;

}

//metodos
protected void Imprimir() {
    System.out.println("Nombre del equipo: " + nombreEquipo);
    System.out.println("Total tiempo de carrera: " + totalCarrera);
    System.out.println("Pais del quipo: " + pais);

}

protected void Añadir(Ciclista C) {
    integrantes.add(C);
}

protected void CalcularTotalTiempo(){
    for (int i=0;i<integrantes.size();i++){
        Ciclista C =(Ciclista) integrantes.elementAt(i);
        totalCarrera=totalCarrera + C.getTiempoCarrera();
    }

}

protected void ListaNombres(){
    for (int i=0;i<integrantes.size();i++){
        Ciclista C =(Ciclista) integrantes.elementAt(i);
        System.out.println(C.getNombre());
    }

}

protected void identificar(int idC){

}

}

```

## Escalador

```

package Ejercicio4;

public class Escalador extends Ciclista{
    private float aceleracionProm;
    private float rampa;

    //constructor
    public Escalador(int idC, String nombreC,float aceleracionC,float rampaC){
        super(idC,nombreC);
        aceleracionProm = aceleracionC;
        rampa = rampaC;
    }

    //metodo get
    protected float getAceleracion() {
        return aceleracionProm;
    }

    protected float getRampa() {
        return rampa;
    }

    //metodo set
    protected void setAceleracion(float aceleracionC){
        aceleracionProm = aceleracionC;
    }
}

```



```

protected void setRampa(float rampaC){
    rampa = rampaC;
}

protected void Imprimir(){
    super.Imprimir();
    System.out.println("Aceleracion promedio subida: " + aceleracionProm );
    System.out.println("Rampa: " + rampa);
}

protected String imprimirTipo(){
    return "Es Escalador";
}

@Override
String ImprimirTipo(String texto) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/C
}
}

```

## Velocista

```

package Ejercicio4;

public class Velocista extends Ciclista {

    private double potenciaProm;
    private double velProm;

    //constructor
    public Velocista(int idC, String nombreC, double potenciaC, double velPromC){
        super(idC, nombreC);
        potenciaProm = potenciaC;
        velProm = velPromC;
    }

    //metodo get
    protected double getPotencia() {
        return potenciaProm;
    }

    protected double getVelocidad() {
        return velProm;
    }

    //metodo set
    protected void setPotencia(double potenciaC) {
        potenciaProm = potenciaC;
    }

    protected void setVelocidad(double velocidadC) {
        velProm = velocidadC;
    }

    //metodos
    protected void Imprimir() {
        super.Imprimir();
        System.out.println("Potencia promedio : " + potenciaProm);
        System.out.println("Velocidad promedio: " + velProm);
    }

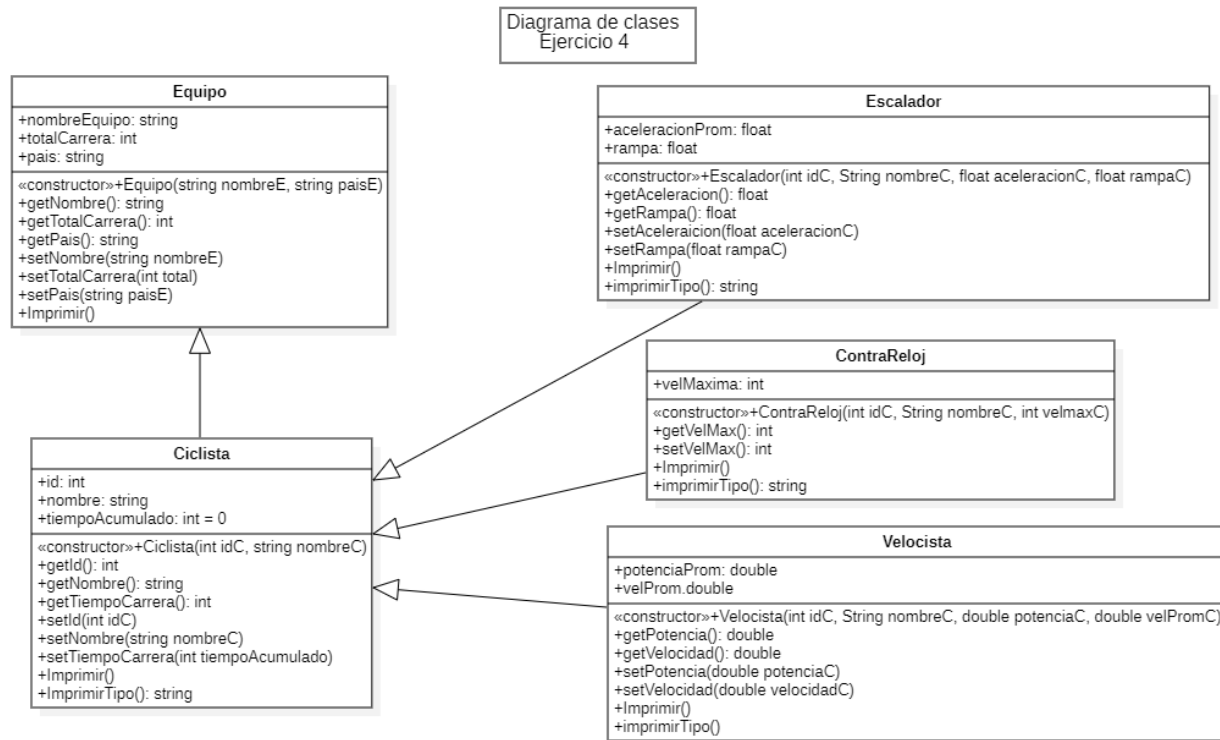
    protected String imprimirTipo(){
        return "Es velocista";
    }

}

@Override
String ImprimirTipo(String texto) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/C
}
}

```

## Diagrama de Clases



## Ejercicio 5



[https://github.com/danielcgirald/P00\\_Grupal/tree/main/Trabajo\\_2/Ejercicio5/src/main/java/Persona](https://github.com/danielcgirald/P00_Grupal/tree/main/Trabajo_2/Ejercicio5/src/main/java/Persona)



<https://drive.google.com/file/d/1kou7o0MrN8A8XxdjmUnt8qAKzajQl8iL/view?usp=sharing>

## Código

### Main

```
package Persona;

public class Main {

    public static void main(String[] args) {
        UI ventana = new UI();
        ventana.setVisible(true);
    }

}
```

### Persona

```
package Persona;

public class Persona {
    String name, surname, phone, address;
}
```

## UI

Nota: Solo se colocó el código generado por nosotros, el código generado por NetBeans para la interface gráfica fue omitido, sin embargo, en el GitHub se encuentra disponible.

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // ===== Create Persona ===== //
    Persona persona = new Persona();
    persona.name = txtName.getText();
    persona.surname = txtSurname.getText();
    persona.phone = txtPhone.getText();
    persona.address = txtAddress.getText();
    // ===== Add Row Table ===== //
    DefaultTableModel model = (DefaultTableModel) Table.getModel();
    model.addRow(new Object[]{persona.name, persona.surname, persona.phone, persona.address});
    txtName.setText("");
    txtSurname.setText("");
    txtPhone.setText("");
    txtAddress.setText("");
}

private void btnCleanActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) Table.getModel();
    for (int i = 0; i < Table.getRowCount(); i++) {
        model.removeRow(i);
        i -= 1;
    }
}

private void btnDeleteRowActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) Table.getModel();
    for (int i = 0; i < Table.getSelectedRows().length; i++) {
        model.removeRow(Table.getSelectedRow());
    }
}
```

## Diagrama de Clases

20