

## EJERCICIO 1:

CLASE :CUENTA.

```
package Herencia;
```

```
public class Cuenta {
```

```
    protected float saldo;
```

```
    protected int númeroConsignaciones = 0;
```

```
    protected int númeroRetiros = 0;
```

```
    protected float tasaAnual;
```

```
    protected float comisiónMensual = 0;
```

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

```
    public void consignar(float cantidad) {  
        saldo = saldo + cantidad;
```

```
        númeroConsignaciones = númeroConsignaciones + 1;
```

```
    }
```

```
    public void retirar(float cantidad) {  
        float nuevoSaldo = saldo - cantidad;  
        if (nuevoSaldo >= 0) {  
            saldo -= cantidad;  
            númeroRetiros = númeroRetiros + 1;  
        }else {  
            System.out.println("La cantida a retirar excede el saldo actual");
```

```
        }  
    }
```

```
    public void calcularInterés() {  
        float tasaMensual = tasaAnual / 12;
```

```

float interesMensual = saldo * tasaMensual;

saldo += interesMensual;

}

public void extractoMensual() {
    saldo -= comisiónMensual;
    calcularInterés();
}
}

```

CLASE : CUENTA AHORRO.

```
package Herencia;
```

```

public class CuentaAhorro extends Cuenta {
    private boolean activa;

    public CuentaAhorro( float saldo, float tasaAnual) {
        super(saldo, tasaAnual);
        if (saldo < 10000)
            activa = false;
        else
            activa = true;
    }

    @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 (númeroRetiros > 4) {
            comisiónMensual += (númeroRetiros - 4) * 1000;
        }
    }
}

```

```

    }
        super.extractoMensual();
        if ( saldo < 10000 )
            activa = false;
    }

    public void imprimir() {
        System.out.println("Saldo = $ " + saldo);
        System.out.println("Comisión mensual = $ " + comisiónMensual);
        System.out.println("Número de transacciones = " + (númeroConsignaciones +
númeroRetiros));
        System.out.println();
    }
}

```

CLASE: CUENTACORRIENTE.

```
package Herencia;
```

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

```
float sobregiro;
```

```

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

```

```

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

```

```

    @Override
    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);
}
}

```

@Override

```

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

```

```

    public void imprimir() {
        System.out.println("Saldo = $ " + saldo);
        System.out.println("Cargo mensual = $ " + comisiónMensual);
        System.out.println("Número de transacciones = " + (númeroConsignaciones +
númeroRetiros));
        System.out.println("Valor de sogregiro = $" + (númeroConsignaciones +
númeroRetiros));
        System.out.println();
    }
}
CLASE ;PRUEBA.

```

```

package Herencia;
import java.util.*;

```

```

public class PruebaCuenta {

    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.println("Cuenta de ahorros");
        System.out.println("Ingrese saldo inicial= $");
        float saldoInicialAhorros = input.nextFloat();
        System.out.print("Ingrese tasa de interés= ");
        float tasaAhorros = input.nextFloat();
        CuentaAhorro cuenta1 = new CuentaAhorro(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();
}
}
```

## Ejercicio 2

```
package Inmuebles;

public class Apartaestudio extends Apartamento {

    protected static double valorArea = 1500000;

    public Apartaestudio(int identificadorInmobiliario, int área, String
    dirección, int númeroHabitaciones, int númeroBaños) {

        super(identificadorInmobiliario, área, dirección, 1, 1);
    }

    @Override
    void imprimir() {
        super.imprimir();
        System.out.println();
    }
}
```

```
package 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 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, int valorAdministracion)  
    {
```

```
        super(identificadorInmobiliario, area, direccion,  
            numeroHabitaciones, numeroBaños);  
        this.valorAdministracion = valorAdministracion;  
    }
```

```
    @Override
```

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

```
package 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, int valorAdministracion)
{

super(identificadorInmobiliario, area, direccion,
numeroHabitaciones, numeroBaños);
this.valorAdministracion = valorAdministracion;
}

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

```
package Inmuebles;

public class CasaConjuntoCerrado extends CasaUrbana {

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 numeroBaños,
int numeroPisos, int valorAdministracion, boolean tienePiscina,
boolean tieneCamposDeportivos) {

super(identificadorInmobiliario, area, direccion,
numeroHabitaciones, numeroBaños, numeroPisos);
```

```
this.valorAdministracion = valorAdministracion;
this.tienePiscina = tienePiscina;
this.tieneCamposDeportivos = tieneCamposDeportivos;
}
```

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

```
package Inmuebles;
```

```
public class CasaIndependiente extends CasaUrbana {
```

```
    protected static double valorArea = 3000000;
```

```
    public CasaIndependiente(int identificadorInmobiliario, int area,
        String direccion, int numeroHabitaciones, int numeroBaños, int
        numeroPisos) {
```

```
        super(identificadorInmobiliario, area, direccion,
            numeroHabitaciones, numeroBaños, numeroPisos);
    }
```

```
@Override
void imprimir() {
    super.imprimir();
    System.out.println();
}
}
```

```
package Inmuebles;
```

```
public class CasaRural extends Casa {
```



```

protected static double valorArea = 1500000;

protected int distanciaCabera;

protected int altitud;
    private String numeroHabitaciones;

public CasaRural(int identificadorInmobiliario, int area, String
direccion, int numeroHabitaciones, int numeroBaños, int
numeroPisos, int distanciaCabera, int altitud) {

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

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

```

```

package Inmuebles;

public class CasaUrbana extends Casa {

public CasaUrbana(int identificadorInmobiliario, int area, String
direccion, int numeroHabitaciones, int numeroBaños, int
numeroPisos) {

super(identificadorInmobiliario, area, direccion,
numeroHabitaciones, numeroBaños, numeroPisos);
}
}

```

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

```
package 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("Area = " + area);
        System.out.println("Dirección = " + direccion);
        System.out.println("Precio de venta = $" + precioVenta);
    }
```

```
}
```

```
package 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;
}
```

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

```
package 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;
    }
```

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

```
package 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;
}

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

```
package 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;
}

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

```
System.out.println();  
}  
}
```

```
package Inmuebles;
```

```
public class Prueba {  
  
    public static void main(String args[]) {  
        ApartamentoFamiliar apto1 = new  
        ApartamentoFamiliar(103067,120,  
        "Avenida Santander 45-45",3,2,200000);  
        System.out.println("Datos apartamento");  
        apto1.calcularPrecioVenta(apto1.valorArea);  
        apto1.imprimir();  
        System.out.println("Datos apartamento");  
        Apartaestudio aptestudio1 = new  
        Apartaestudio(12354,50,"Avenida Caracas 30-15",1,1);  
        aptestudio1.calcularPrecioVenta(aptestudio1.valorArea);  
        aptestudio1.imprimir();  
    }  
}
```

### **EJERCICIO 3:**

CLASE: ANIMAL.

```
package Animales;
```

```
public abstract class Animal {  
    protected String sonido;
```

```
    protected String alimentos;
```

```
    protected String hábitat;
```

```
protected String nombreCientifico;

public abstract String getNombreCientifico();

public abstract String getSonido();

public abstract String getAlimentos();

public abstract String getHabitat();

}
```

CLASE: ABSTRACT CÁNIDO..

```
package Animales;

public abstract class Canido extends Animal {

}
```

CLASE: PERRO.

```
package Animales;

public class Perro extends Canido{
    public String getSonido() {
        return "Ladrado";
    }

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

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

```
}

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

CLASE: LOBO.

package Animales;

public class Lobo extends Canido {

```
    public String getSonido() {
        return "Aullido";
    }
```

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

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

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

CLASE; FELINO.

package Animales;

public abstract class Felino extends Animal{

```
}
```

CLASE:LEÓN

package Animales;

public class Leon extends Felino {

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

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

```
}  
public String getHabitat() {  
    return "Praderas";  
}
```

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

CLASE:GATO.

```
package 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";  
    }  
}
```



CLASE: PRUEBA.

```
package Animales;
```

```
public class Prueba {
```

```
    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();
```

```
    }  
}  
}
```

## Ejercicio 4

```
package 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() {  
    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) {  
    posicionGeneral = posicionGeneral;  
}
```

```
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 CarreraCiclística;
```

```
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("Aceleración promedio = " +
velocidadMaxima);
}
```

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

```
package 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 CarreraCiclística;
```

```
public class Escalador extends Ciclista {
```

```
    private double aceleraciónPromedio;
```

```
    private double gradoRampa;
```

```
    public Escalador(int identificador, String nombre, double  
        aceleraciónPromedio, double gradoRampa) {  
        super(identificador, nombre);  
        this.aceleraciónPromedio = aceleraciónPromedio;  
        this.gradoRampa = gradoRampa;
```

```
}
```

```
protected double getAceleraciónPromedio() {  
    return aceleraciónPromedio;  
}
```

```
protected void setAceleraciónPromedio(double  
    aceleraciónPromedio) {  
    this.aceleraciónPromedio = aceleraciónPromedio;  
}
```

```
protected double getGradoRampa() {  
    return gradoRampa;  
}
```

```
protected void setGradoRampa(double gradoRampa) {  
    this.gradoRampa = gradoRampa;  
}
```

```
protected void imprimir() {  
    super.imprimir();  
    System.out.println("Aceleración promedio = " + aceleraciónPromedio);  
    System.out.println("Grado de rampa = " + gradoRampa);  
}
```

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

```
package CarreraCiclistica;
```

```
public class Prueba {
```

```
    public static void main(String args[]) {  
        Equipo equipo1 = new Equipo("Sky","Estados Unidos");  
        Velocista velocista1 = new Velocista(123979, "Geraint Thomas",  
            320, 25);  
        Escalador escalador1 = new Escalador(123980, "Egan Bernal",  
            25, 10);
```

```
Contrarrelojista contrarrelojista1 = new Contrarrelojista(123981,
"Jonathan Castroviejo", 120);
equipo1.añadirCiclista(velocista1);
equipo1.añadirCiclista(escalador1);
equipo1.añadirCiclista(contrarrelojista1);
velocista1.setTiempoAcumulado(365);
escalador1.setTiempoAcumulado(385);
contrarrelojista1.setTiempoAcumulado(370);
equipo1.calcularTotalTiempo();
equipo1.imprimir();
equipo1.listarEquipo();
}
}
```

```
package CarreraCiclística;
```

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

```
public Velocista(int identificador, String nombre, double
potenciaPromedio, double velocidadPromedio) {
super(identificador, nombre);
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 5

```
package Personas;  
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 Personas;
```

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

```
package Personas;
```

```
public class Principal {  
  
    public static void main(String[] args) {  
        VentanaPrincipal miVentanaPrincipal;  
        miVentanaPrincipal= new VentanaPrincipal();  
        miVentanaPrincipal.setVisible(true);  
    }  
}  
  
package Personas;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.*;
```

```
public class VentanaPrincipal extends JFrame implements  
    ActionListener {  
    private ListaPersonas lista;  
    private Container contenedor;  
    private JLabel nombre, apellidos, teléfono, dirección;
```

```
    private JTextField campoNombre, campoApellidos, campoTeléfono,
```

```
campoDirección;  
private JButton añadir, eliminar, borrarLista;  
private JList listaNombres;  
private DefaultListModel modelo;  
private JScrollPane scrollLista;
```

```
public VentanaPrincipal(){  
    lista = new ListaPersonas();  
    inicio();  
    setTitle("Personas");  
    setSize(270,350);  
    setLocationRelativeTo(null);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setResizable(false);  
}
```

```
private void inicio() {  
    contenedor = getContentPane();  
    contenedor.setLayout(null);  
    nombre = new JLabel();  
    nombre.setText("Nombre:");  
    nombre.setBounds(20, 20, 135, 23);  
    campoNombre = new JTextField();
```

```
    campoNombre.setBounds(105, 20, 135, 23);
```

```
    apellidos = new JLabel();  
    apellidos.setText("Apellidos:");  
    apellidos.setBounds(20, 50, 135, 23);  
    campoApellidos = new JTextField();
```

```
    campoApellidos.setBounds(105, 50, 135, 23);
```

```
    teléfono = new JLabel();  
    teléfono.setText("Teléfono:");  
    teléfono.setBounds(20, 80, 135, 23);  
    campoTeléfono = new JTextField();
```

```
    campoTeléfono.setBounds(105, 80, 135, 23);
```

```
    dirección = new JLabel();  
    dirección.setText("Dirección:");
```

```
direccion.setBounds(20, 110, 135, 23);
campoDireccion = new JTextField();
```

```
campoDireccion.setBounds(105, 110, 135, 23);
```

```
añadir = new JButton();
añadir.setText("Añadir");
añadir.setBounds(105, 150, 80, 23);
añadir.addActionListener(this);
```

```
eliminar= new JButton();
eliminar.setText("Eliminar");
eliminar.setBounds(20, 280, 80, 23);
eliminar.addActionListener(this);
```

```
borrarLista= new JButton();
borrarLista.setText("Borrar Lista");
borrarLista.setBounds(120, 280, 120, 23);
borrarLista.addActionListener(this);
```

```
listaNombres = new JList();
```

```
listaNombres.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
modelo = new DefaultListModel();
```

```
scrollLista = new JScrollPane();
```

```
scrollLista.setBounds(20, 190 ,220, 80);
```

```
scrollLista.setViewportView(listaNombres);
```

```
contenedor.add(nombre);
contenedor.add(campoNombre);
contenedor.add(apellidos);
contenedor.add(campoApellidos);
contenedor.add(teléfono);
contenedor.add(campoTeléfono);
contenedor.add(dirección);
contenedor.add(campoDirección);
contenedor.add(añadir);
contenedor.add(eliminar);
contenedor.add(borrarLista);
contenedor.add(scrollLista);
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent evento) {
```

```
if (evento.getSource() == añadir) {
```

```
    añadirPersona();
```

```
}
```

```
if (evento.getSource() == eliminar) {
```

```
    eliminarNombre(listaNombres.getSelectedIndex() );
```

```
}
```

```
if (evento.getSource() == borrarLista) {
```

```
    borrarLista();
```

```
}
```

```
}
```

```
private void añadirPersona() {
```

```
    Persona p = new Persona(campoNombre.getText(),
```

```
        campoApellidos.getText(),
```

```
        campoTeléfono.getText(), campoDirección.getText());
```

```
    lista.añadirPersona(p);
```

```
    String elemento = campoNombre.getText() + "-" +
```

```
        campoApellidos.getText() +
```

```
        "-" + campoTeléfono.getText() + "-" + campoDirección.
```

```
    getText();
```

```
    modelo.addElement(elemento);
```

```
    listaNombres.setModel(modelo);
```

```
    campoNombre.setText("");
```

```
    campoApellidos.setText("");
```

```
    campoTeléfono.setText("");
```

```
    campoDirección.setText("");
```

```
}
```

```
private void eliminarNombre(int indice) {
```

```
    if (indice >= 0) {
```

```
        modelo.removeElementAt(indice);
```

```
        lista.eliminarPersona(indice);
```

```
    } else {
```

```
        JOptionPane.showMessageDialog(null, "Debe seleccionar un elemento", "Error",
```

```
        JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
}
```

```
private void borrarLista() {  
    lista.borrarLista();  
    modelo.clear();  
}  
}
```