

## ACTIVIDAD 4: 20%

**Integrantes: Lilibeth Barceló Acosta-David Gavalo Molina-Wilson Ocampo Cárdenas**

### Código punto #1

<https://github.com/Guilsin/ACTIVIDAD-4/tree/main/punto1>

//Clase cuenta

```
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 Cantidad 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 CuentaAhorros
public class CuentaAhorros extends Cuenta {
    private boolean activa;

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

```

```
}
```

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

```
public void extactoMensual() {  
    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

```
public class CuentaCorriente extends Cuenta {  
    float sobregiro;  
  
    public CuentaCorriente(float saldo,float tasa) {  
        super(saldo,tasa);  
        sobregiro = 0;  
    }  
  
    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=$"+saldo);
    System.out.println("Cargo mensual=$"+comisiónMensual);
    System.out.println("Número de
transacciones="+ (numeroConsignaciones+numeroRetiros));
    System.out.println("Valor de
sobregiro=$"+ (numeroConsignaciones+numeroRetiros));
    System.out.println();
}
}

```

//Clase PruebaCuenta

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

```
public class PruebaCuenta {
```

```
    public static void main(String args[]) {
```

```
        Scanner entrada = new Scanner(System.in);
```

```
        System.out.println("Cuenta de ahorros");
```

```
        System.out.println("Ingrese saldo inicial=$");
```

```

        float saldoInicialAhorros = entrada.nextFloat();

        System.out.println("Ingrese tasa de interés=");

        float tasaAhorros= entrada.nextFloat();

        CuentaAhorros cuenta = new CuentaAhorros(saldoInicialAhorros,
tasaAhorros);

        System.out.println("Ingresar cantidad a consignar=$");

        float cantidadDepositar =entrada.nextFloat();

        cuenta.consignar(cantidadDepositar);

        System.out.println("Ingresar cantidad a retirar= $");

        float cantidadRetirar=entrada.nextFloat();

        cuenta.retirar(cantidadRetirar);

        cuenta.extactoMensual();

        cuenta.imprimir();

    }

}

```

---

## Código punto #2

<https://github.com/Guilsin/ACTIVIDAD-4/tree/main/punto2>

//Clase ApartamentoApartaestudio

package punto2;

public class apartamentoApartaestudio extends viviendaApartamento {

protected static double valorArea = 1500000;

//constructor apartaestudio:

public apartamentoApartaestudio(int identificador, int area, String direccion, int numeroHabitaciones, int numeroBaños){

```

        //los apartaestudios solo tienen 1 baño y 1 habitación
        super(identificador, area, direccion,1,1);
    }

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

//Clase ApartamentoFamiliar
package punto2;

public class apartamentoFamiliar extends viviendaApartamento {

    //atributos:
    protected static double valorArea = 2000000;
    protected int valorAdministracion;

    //constructor apartamento familiar:
    public apartamentoFamiliar(int identificador, int area, String direccion, int
numeroHabitaciones, int numeroBaños, int valorAdministracion){
        super(identificador, area, direccion, numeroHabitaciones, numeroBaños);
        this.valorAdministracion = valorAdministracion;
    }

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

```

```

        System.out.println();
    }
}

//Clase casaConjunto
package punto2;

public class casaConjunto extends casaUrbana {

    //atributos:
    protected static double valorArea = 2500000;
    protected int valorAdministracion;
    protected boolean tienePiscina,tieneCancha;

    public casaConjunto(int identificador, int area, String direccion, int numeroHabitaciones,
int numeroBaños, int numeroPisos, int valorAdministracion, boolean tienePiscina, boolean
tieneCancha){
        super(identificador, area, direccion, numeroHabitaciones, numeroBaños,
numeroPisos);
        this.valorAdministracion = valorAdministracion;
        this.tienePiscina = tienePiscina;
        this.tieneCancha = tieneCancha;
    }

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

```



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

//Clase casaIndependiente

package punto2;

public class casaIndependiente extends casaUrbana{

//atributos:

protected static double valorArea = 3000000;

public casaIndependiente(int identificador, int area, String direccion, int  
numeroHabitaciones, int numeroBaños, int numeroPisos){

super(identificador, area, direccion, numeroHabitaciones, numeroBaños,  
numeroPisos);

}

void imprimir(){

super.imprimir();

System.out.println();

}

}

//Clase casaRural

package punto2;

public class casaRural extends viviendaCasa {

//atributos:

```

protected static double valorArea = 1500000;

protected int distanciaCabecera, altitud;

//método constructor casa rural:

public casaRural(int identificador, int area, String direccion, int numeroHabitaciones, int
numeroBaños, int numeroPisos, int distanciaCabecera, int altitud){

    super(identificador, area, direccion, numeroHabitaciones, numeroBaños,
numeroPisos);

    this.distanciaCabecera = distanciaCabecera;

    this.altitud = altitud;

}

void imprimir(){

    super.imprimir();

    System.out.println("Distancia a la cabecera municipal = " + distanciaCabecera +
"km");

    System.out.println("Altitud sobre el nivel del mar = " + altitud);

    System.out.println();

}

}

//Clase casaUrbana

package punto2;

public class casaUrbana extends viviendaCasa {

    //método constructor de la casa urbana:

    public casaUrbana(int identificador, int area, String direccion, int numeroHabitaciones,
int numeroBaños, int numeroPisos){

```

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

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

//Clase inmueble

package punto2;

public class inmueble {

//atributos

protected int identificador, area ;

protected String direccion;

protected double precioVenta;

//método constructor:

inmueble(int identificador, int area, String direccion){

this.identificador = identificador;

this.area = area;

this.direccion = direccion;

}

//metodo calculador de precio de venta:

double calcularPrecioVenta(double valorArea){

precioVenta = area\*valorArea;

```

        return precioVenta;
    }

    //metodo muestra en pantalla los datos del inmueble
    void imprimir(){
        System.out.println("Identificador inmobiliario = " + identificador);
        System.out.println("Área = " + area);
        System.out.println("Dirección = " + direccion);
        System.out.println("Precio de venta = $" + precioVenta);
    }
}

//Clase inmuebleLocal
package punto2;

public class inmuebleLocal extends inmueble {

    //atributos:
    enum tipo{INTERNO,CALLE};
    protected tipo tipoLocal;

    //constructor inmueble local:
    public inmuebleLocal(int identificador, int area, String direccion, tipo tipoLocal){
        super(identificador, area, direccion);
        this.tipoLocal = tipoLocal;
    }

    void imprimir(){

```

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

//Clase inmuebleVivienda

package punto2;

```
public class inmuebleVivienda extends inmueble{  
    //atributos  
    protected int numeroHabitaciones, numeroBaños;  
  
    //método constructor de vivienda:  
    public inmuebleVivienda(int identificador, int area, String direccion, int  
numeroHabitaciones, int numeroBaños){  
        super(identificador, area, direccion); //invoca al método constructor de la clase padre  
        this.numeroHabitaciones = numeroHabitaciones;  
        this.numeroBaños = numeroBaños;  
    }  
  
    void imprimir(){  
        super.imprimir(); //invoca al método imprimir de la clase padre  
        System.out.println("Número de habitaciones = " + numeroHabitaciones);  
        System.out.println("Número de baños = " + numeroBaños);  
    }  
}
```

//Clase localComercial

```
package punto2;
```

```
public class localComercial extends inmuebleLocal {
```

```
    //atributos:
```

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

```
    protected String centroComercial;
```

```
    //método constructor del local comercial:
```

```
    public localComercial(int identificador, int area, String direccion, tipo tipoLocal, double valorArea, String centroComercial){
```

```
        super(identificador, area, direccion, tipoLocal);
```

```
        this.centroComercial = centroComercial;
```

```
    }
```

```
    void imprimir(){
```

```
        super.imprimir();
```

```
        System.out.println("Centro comercial = " + centroComercial);
```

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

```
    }
```

```
}
```

```
//Clase localOficina
```

```
package punto2;
```

```
public class localOficina extends inmuebleLocal {
```

```
    //atributos:
```

```
    protected static double valorArea = 3500000;
```

```
protected boolean esGobierno;
```

```
//método constructor de oficina:
```

```
public localOficina(int identificador, int area, String direccion , tipo tipoLocal, boolean  
esGobierno){
```

```
    super(identificador, area, direccion, tipoLocal);
```

```
    this.esGobierno = esGobierno;
```

```
}
```

```
void imprimir(){
```

```
    super.imprimir();
```

```
    System.out.println("Es oficina gubernamental? = " + esGobierno);
```

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

```
}
```

```
}
```

```
//Clase main
```

```
package punto2;
```

```
public class main {
```

```
    public static void main(String[] args) {
```

```
        apartamentoFamiliar apto1 = new apartamentoFamiliar(103067,120,"Avenida Las  
palmas 69-420",3,2,2000000);
```

```
        System.out.println("Datos del inmueble:");
```

```
        apto1.calcularPrecioVenta(apt01.valorArea);
```

```
        apto1.imprimir();
```

```

        apartamentoApartaestudio aptestudio1 = new
apartamentoApartaestudio(12354,50,"Avenida Caracas 30-15",1,2);
        System.out.println("Datos del inmueble:");
        aptestudio1.calcularPrecioVenta(aptestudio1.valorArea);
        aptestudio1.imprimir();
    }
}

```

//Clase viviendaApartamento

package punto2;

public class viviendaApartamento extends inmuebleVivienda {

//método constructor de apartamento:

```

    public viviendaApartamento(int identificador, int area, String direccion, int
numeroHabitaciones, int numeroBaños){
        super( identificador, area, direccion, numeroHabitaciones, numeroBaños);
    }

```

```

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

```

//Clase viviendaCasa

package punto2;

public class viviendaCasa extends inmuebleVivienda{



```
//atributos

protected int numeroPisos;


//método constructor de casa:

public viviendaCasa(int identificador, int area, String direccion, int numeroHabitaciones,
int numeroBaños, int numeroPisos){

    super( identificador, area, direccion, numeroHabitaciones, numeroBaños);

    this.numeroPisos = numeroPisos;

}


void imprimir(){

    super.imprimir();

    System.out.println("Número de pisos = " + numeroPisos);

}

}
```

---

### **Código punto #3**

<https://github.com/Guilsin/ACTIVIDAD-4/tree/main/punto3>

//Clase Animal

```
package punto3;


public abstract class Animal {

    protected String sonido;

    protected String alimentos;

    protected String hábitat;

    protected String nombreCientífico;


    public abstract String getNombreCientífico();


    public abstract String getSonido();

}
```

```
        public abstract String getAlimentos();

        public abstract String getHábitat();

    }
}
```

//Clase Cánido

```
package punto3;
```

```
public abstract class Cánido extends Animal {

}
```

//Clase Felino

```
package punto3;
```

```
public abstract class Felino extends Animal{

}
```

//Clase Felino

```
package punto3;
```

```
public class Gato extends Felino{

    public String getSonido() {

        return "Maullido";

    }

}
```

```
        public String getAlimentos() {  
            return "Ratones";  
        }  
  
        public String getHábitat() {  
            return "Doméstico";  
        }  
  
        public String getNombreCientífico() {  
            return "Felis silvestris catus";  
        }  
    }  
}
```

//Clase León

package punto3;

```
public class León extends Felino{  
    public String getSonido() {  
        return "Rugido";  
    }  
  
    public String getAlimentos() {  
        return "Carnívoro";  
    }  
  
    public String getHábitat() {  
        return "Praderas";  
    }  
}
```

```
    }

    public String getNombreCientífico() {
        return "Panthera leo";
    }

}
```

//Clase Lobo

package punto3;

```
public class Lobo extends Cánido {
    public String getSonido() {
        return "Aullido";
    }

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

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

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

```
}
```

```
//Clase Perro
```

```
package punto3;
```

```
public class Perro extends Cánido {
```

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

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

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

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

```
}
```

```
//Clase PruebaAnimal
```

```
package punto3;
```

```

public class PruebaAnimal {
    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 León();

        for(int i = 0; i < animales.length; i++) {
            System.out.println(animales[i].getNombreCientífico());
            System.out.println("sonido=" + animales[i].getSonido());
            System.out.println("Alimentos=" + animales[i].getAlimentos());
            System.out.println("Hábitat=" + animales[i].getHábitat());
            System.out.println();
        }
    }
}

```

---

#### **Código punto #4**

<https://github.com/Guilsin/ACTIVIDAD-4/tree/main/punto4>

//Clase Ciclista

package punto4;

```

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 getPosiciónGeneral(int posiciónGeneral) {
```

```

        return posiciónGeneral;
    }

    protected void setPositionGeneral(int posiciónGeneral) {
        posiciónGeneral = posiciónGeneral;
    }

    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);
    }
}

```

//Clase Contrarrelojista

package punto4;

```

public class Contrarrelojista extends Ciclista{
    private double velocidadMáxima;

```



```

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

    protected double getVelocidadMáxima() {
        return velocidadMáxima;
    }

    protected void setVelocidadMáxima(double velocidadMáxima) {
        this.velocidadMáxima=velocidadMáxima;
    }

    protected void imprimir() {
        super.imprimir();
        System.out.println("Aceleración promedio="+ velocidadMáxima);
    }

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

}

//Clase Equipo
package punto4;

```

```
import java.util.*;

public class Equipo {

    private String nombre;

    private static double totalTiempo;

    private String país;

    Vector<Ciclista> listaCiclistas;


    public Equipo(String nombre,String país) {

        this.nombre = nombre;

        this.país = país;

        totalTiempo = 0;

        listaCiclistas= new Vector<Ciclista>();

    }


    public String getNombre() {

        return nombre;

    }


    public void setNombre(String nombre) {

        this.nombre= nombre;

    }


    private String getPaís() {

        return país;

    }


    private void setPaís(String país) {

        this.país=país;

    }

}
```

```
}
```

```
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 entrada = new Scanner(System.in);
```

```
    String nombreCiclista = entrada.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("País="+país);
    System.out.println("Total tiempo del equipo="+totalTiempo);
}
}

```

//Clase Escalador

package punto4;

```

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";
}

}

//Clase pruebaCiclista
package punto4;

```

```

public class PruebaCiclista {
    public static void main(String args[]) {
        Equipo equipo= new Equipo("Sky","Estados Unidos");
        Velocista velocista = new Velocista(123979,"Geraint Thomas",320,25);
        Escalador escalador= new Escalador(123980,"Egan Bernal",25,10);
        Contrarrelojista contrarrelojista = new Contrarrelojista(123981,"Jonathan
Castroviejo",120);
        equipo.añadirCiclista(velocista);
        equipo.añadirCiclista(escalador);
        equipo.añadirCiclista(contrarrelojista);
        velocista.setTiempoAcumulado(365);
        escalador.setTiempoAcumulado(385);
        contrarrelojista.setTiempoAcumulado(370);
        equipo.calcularTotalTiempo();
        equipo.imprimir();
        equipo.listarEquipo();
    }
}

```

//Clase Velocista

```
package punto4;
```

```

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

```

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

protected double getPotencialPromedio() {
    return potencialPromedio;
}

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

protected double getvelocidadPromedio() {
    return velocidadPromedio;
}

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

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

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

---

### **Código punto #5**

<https://github.com/Guilsin/ACTIVIDAD-4/tree/main/punto5>

```
//Clase ListaPersonas  
package punto5;  
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();  
    }  
}
```



```
    }  
}
```

//Clase Persona

package punto5;

```
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;  
    }  
}
```

//Clase Principal

package punto5;

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

```
        miVentanaPrincipal.setVisible(true);  
    }  
}
```

//Clase ventanaPrincipal

```
package punto5;
```

```
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:");  
    dirección.setBounds(20, 110, 135, 23);  
    campoDirección = new JTextField();  
    campoDirección.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.setViewportViewView(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();
    }
}

```