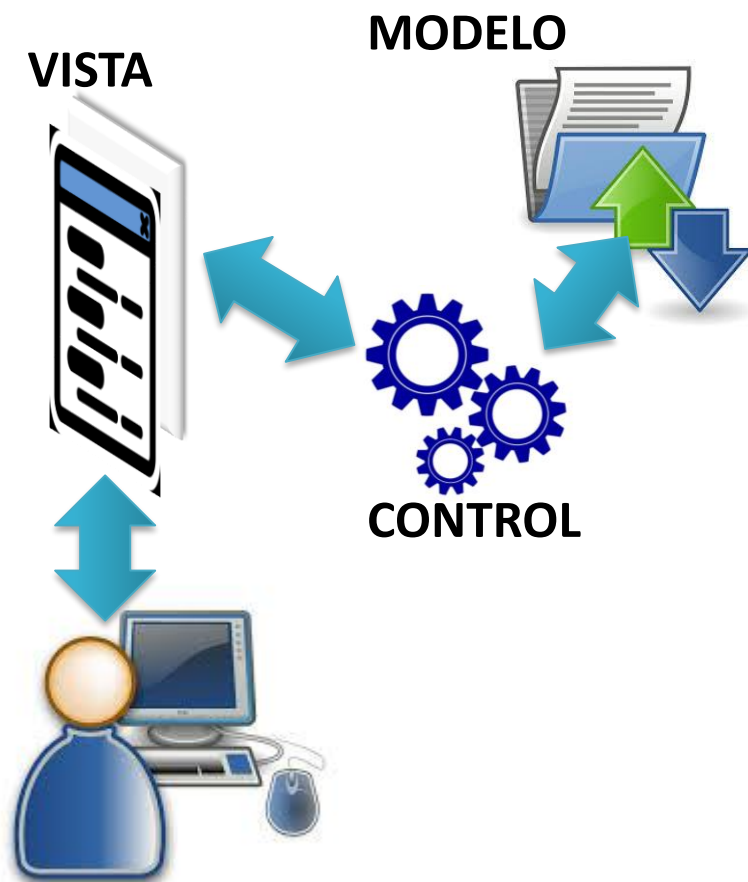




# Universidad Distrital Francisco José de caldas

## Tecnología en Sistematización de Datos

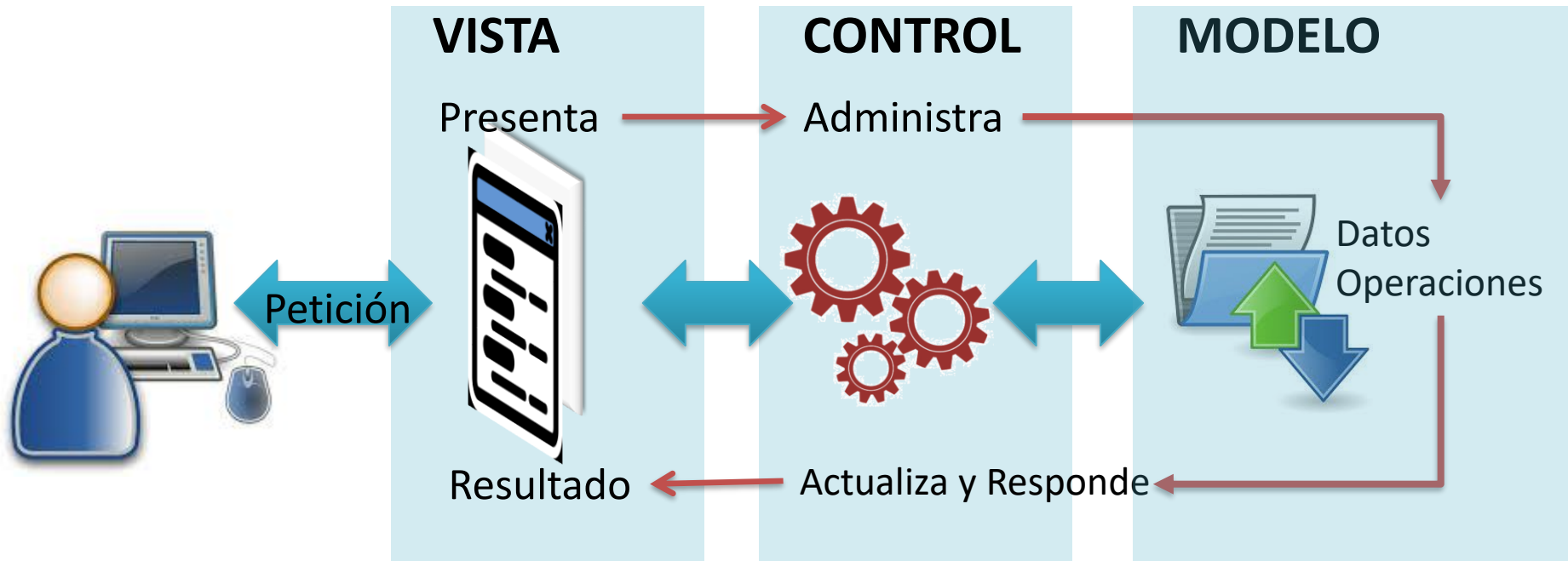


## Modelo Vista Controlador

Programación Multinivel  
Sonia Alexandra Pinzón Nuñez  
2021



# CAPAS DEL MVC

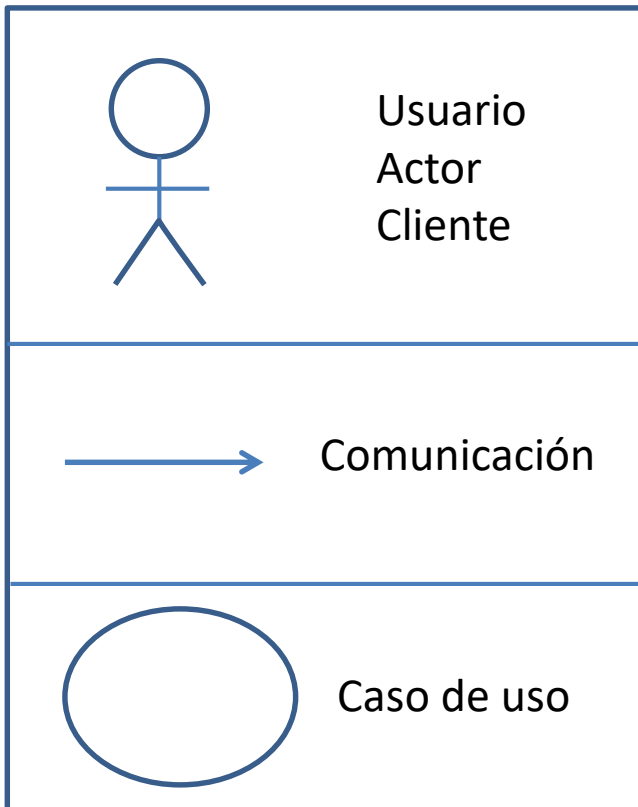




# Diagrama Casos de Uso

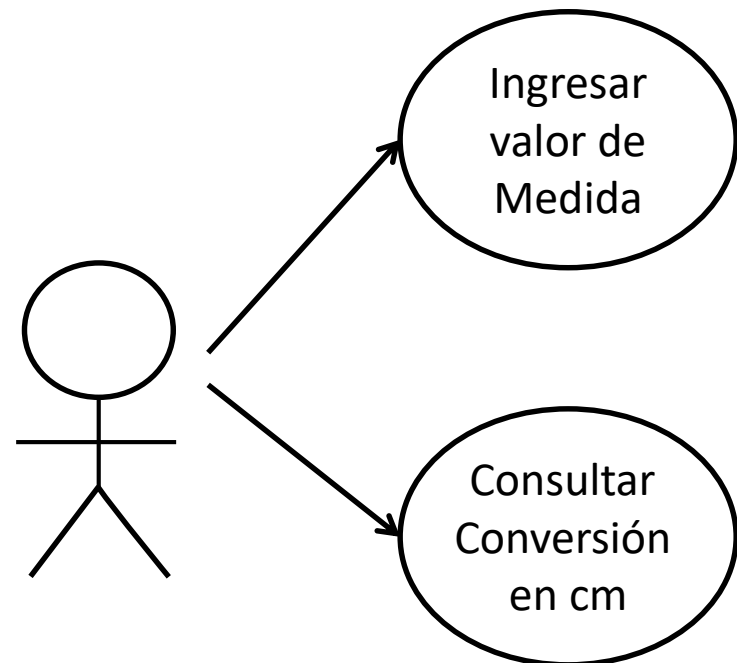
Corresponde a la representación de las acciones que hace el usuario con la aplicación. (Como se usa).

## Estereotipos



## Ejemplo

Convertir una medida dada en metros a centímetros

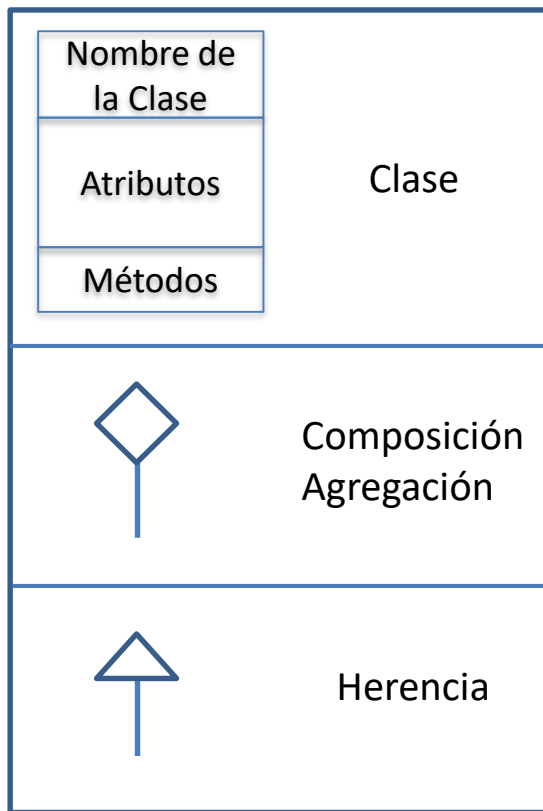




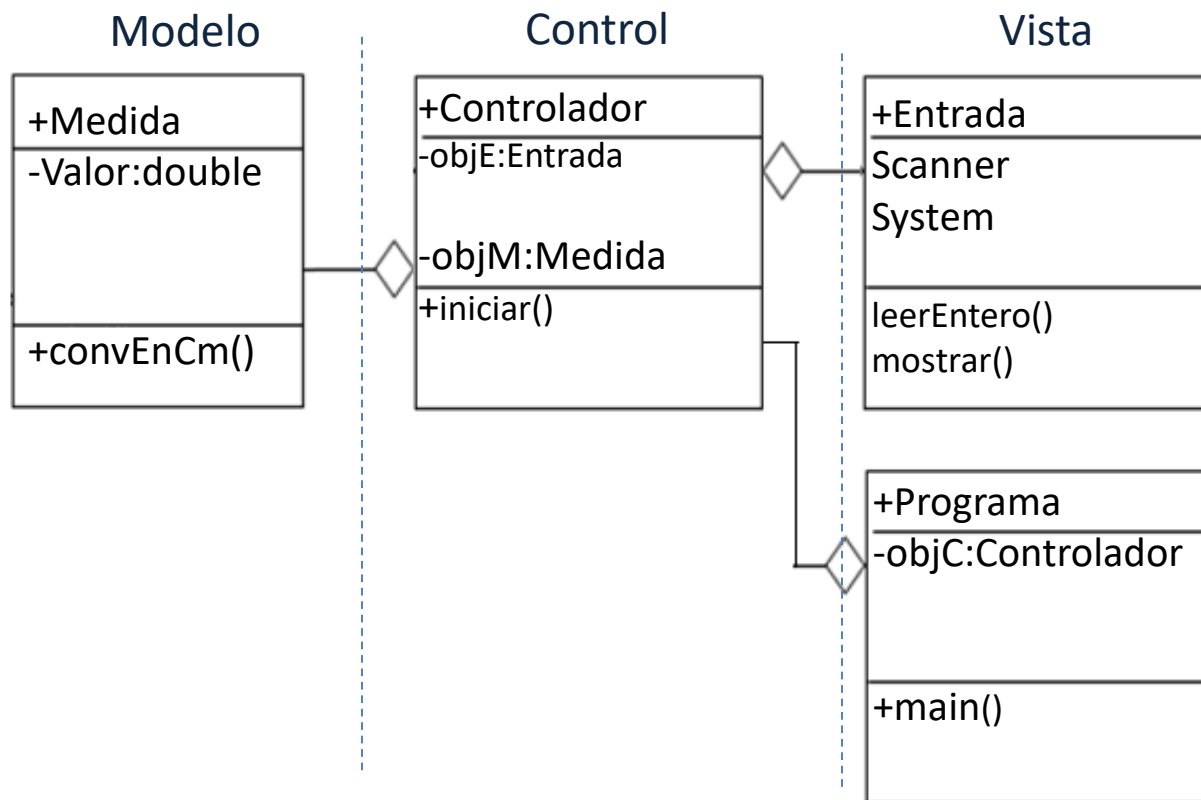
# Diagrama de Clases

Representa las relaciones entre las clases definidas para el sistema (lógica del negocio)

## Estereotipos



## Ejemplo

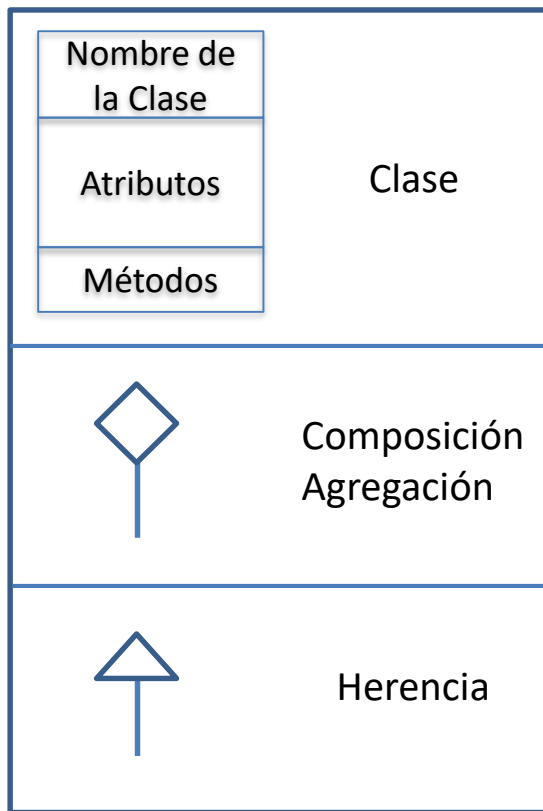




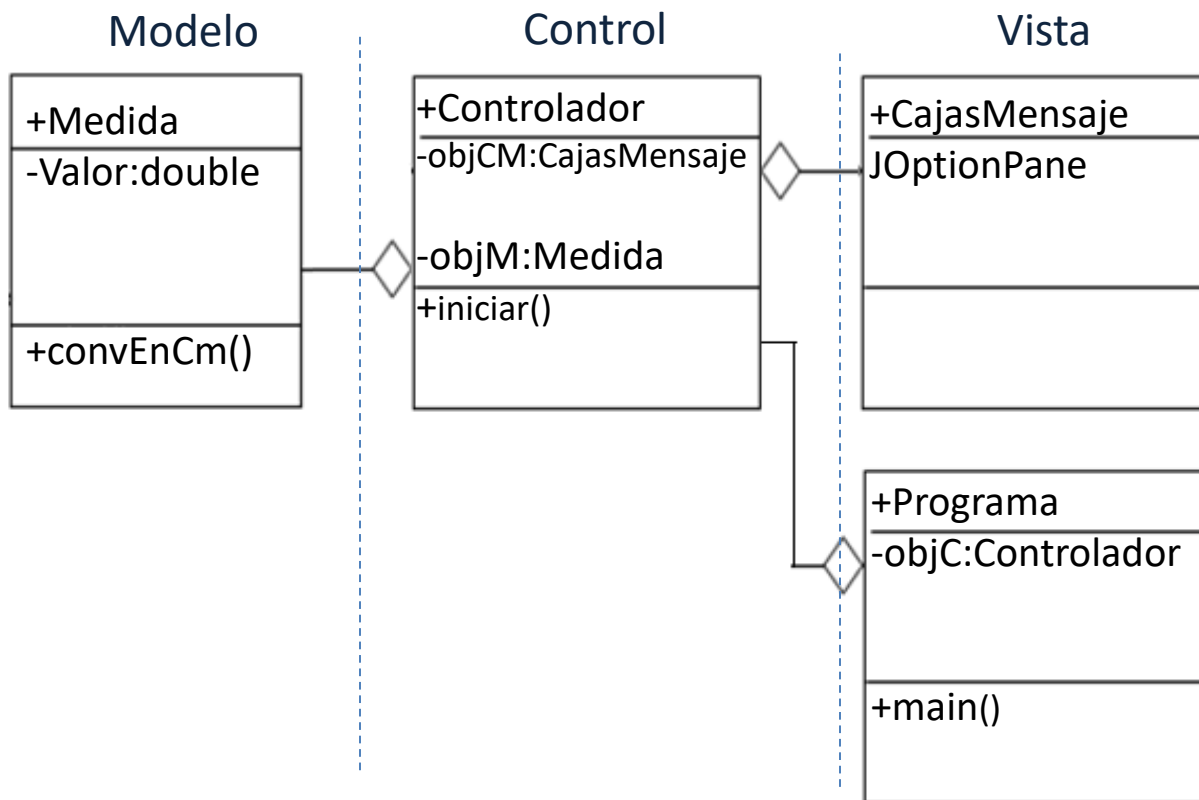
# Diagrama de Clases

Representa las relaciones entre las clases definidas para el sistema (lógica del negocio)

## Estereotipos



## Ejemplo

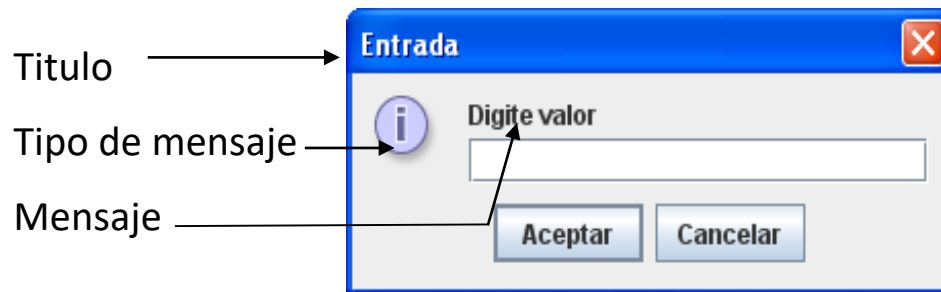




# Diseño de Interfaz - Cajas de Mensaje

## Clase JOptionPane

### ENTRADA DE DATOS `showInputDialog()`



ICONO	TIPO DE MENSAJE	VALOR
	ERROR_MESSAGE	0
	INFORMATION_MESSAGE	1
	WARNING_MESSAGE	2
	QUESTION_MESSAGE	3

```
JOptionPane.showInputDialog(contenedor, mensaje, titulo, tipo de mensaje);
```

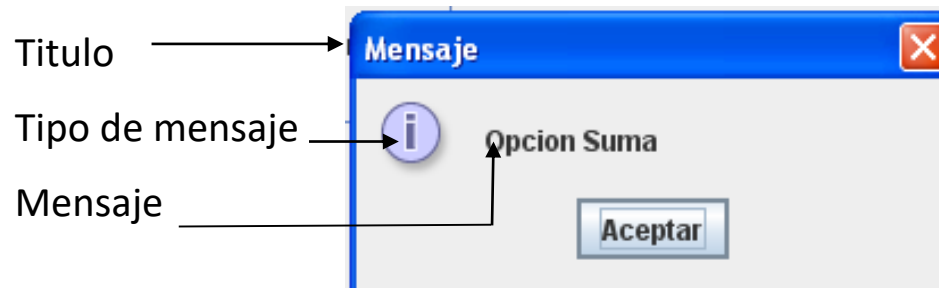
```
String dato= JOptionPane.showInputDialog(this,  
    "Digite valor",  
    "Entrada",  
    JOptionPane.INFORMATION_MESSAGE);
```



# Diseño de Interfaz - Cajas de Mensaje

## Clase JOptionPane

**MENSAJE** `showMessageDialog()`



`JOptionPane.showMessageDialog(contenedor, mensaje, titulo, tipo de mensaje);`

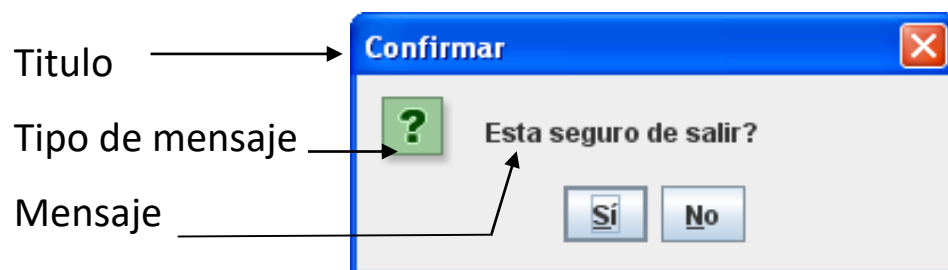
```
JOptionPane.showMessageDialog(null,  
                             "Opción Suma",  
                             mensaje,  
                             1);
```



# Diseño de Interfaz - Cajas de Mensaje

## Clase JOptionPane

**CONFIRMAR** showConfirmDialog()



`JOptionPane.showConfirmDialog(contenedor, mensaje, titulo, tipo de Botones, tipo de mensaje);`

```
int resp= JOptionPane.showConfirmDialog(null,  
    "Desea terminar el proceso?",  
    titulo, JOptionPane.YES_NO_OPTION,  
    JOptionPane.QUESTION_MESSAGE);  
if(resp== JOptionPane.YES_OPTION)  
    System.exit(0);
```





# Implementación:

Codificar las clases definidas en el diagrama de clases

## Modelo

```
public class Medida {  
    private double valor;  
  
    public Medida(double valor) {  
        this.valor = valor;  
    }  
    public Medida() {  
        this.valor = 0;  
    }  
    public double getValor() {  
        return valor;  
    }  
    public void setValor(double valor) {  
        this.valor = valor;  
    }  
    @Override  
    public String toString() {  
        return "valor =" + valor;  
    }  
    public double convEnCm() {  
        return this.valor*100;  
    }  
}
```



# Implementación:

## Vista Cajas de Mensaje

```
public class CajasMensaje {  
    private String titulo;  
    public CajasMensaje(String titulo) {  
        this.titulo = titulo;  
    }  
    public CajasMensaje() {  
        this.titulo = "";  
    }  
    public String getTitulo() {  
        return titulo;  
    }  
    public void setTitulo(String titulo) {  
        this.titulo = titulo;  
    }  
    @Override  
    public String toString() {  
        return "titulo=" + titulo ;  
    }  
    public int leerEntero(String msj){  
        return Integer.parseInt(JOptionPane.showInputDialog(null,msj,titulo,1));  
    }  
    public double leerDecimal(String msj){  
        return Double.parseDouble(JOptionPane.showInputDialog(null,msj,titulo,1));  
    }  
    public String leerTexto(String msj){  
        return JOptionPane.showInputDialog(null,msj,titulo,1);  
    }  
    public void mostrar(String msj)  
    {  
        JOptionPane.showMessageDialog(null, msj,titulo,1);  
    }  
}
```



# Implementación:

## Control

```
package control;
import modelo.Medida;
import vista.*;
/**
 * @author Sonia Pinzón
 */
public class ControladorCM{
    Medida objM;
    CajasMensaje objE;
    public ControladorCM(Medida objM, CajasMensaje en) {
        this.objM = objM;
        this.objE = en;
    }
    public ControladorCM() {
        this.objM = new Medida();
        this.objE = new CajasMensaje();
    }
    public void iniciar(){
        objE.setTitulo("Conversión a Cms");
        objM.setValor(objE.leerDecimal("Digite medida en metros "));
        objE.mostrar("La medida "+ objM.getValor()+
            "mts es igual a "+objM.convertirEnCM()+"cms ");
    }
}
```

Método que  
desarrolla o inicia  
la acción →



# Implementación:

Vista

Programa  
autónomo

```
package proymedidamvc;
import control.*;
/**
 *
 * @author Sonia Pinzón
 */
public class ProyMedidaMVC {

    public static void main(String[] args) {
        ControladorCM objC= new ControladorCM();
        objC.iniciar();
    }

}
```



# Bibliografía

---

- Documentación Java Oracle paquete Swing.  
Disponible en:  
<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>.
- Deitel y Deitel. Programación Java. Editorial McGraw Hill.