Tema 7
Modelo-Vista-Controlador

Proyecto de Análisis y Diseño de Software

2º Ingeniería Informática

Universidad Autónoma de Madrid

Modelo-Vista-Controlador (MVC)

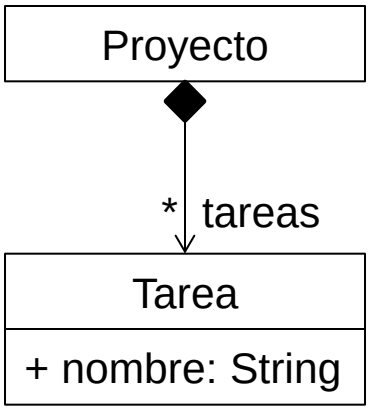
- Modelo (clases del dominio)
- Vistas (clases que heredan de JComponent, ej. JButton…)
- Controladores (clases que implementan interfaces de tipo EventListener, ej. ActionListener, MouseListener…)

1. El controlador se registra en la interfaz
2. El usuario realiza acción en la interfaz
3. El controlador trata el evento:
   1. cambia el modelo (si hace falta)
   2. genera la nueva vista
4. La nueva vista toma los datos a mostrar del modelo

Modelo

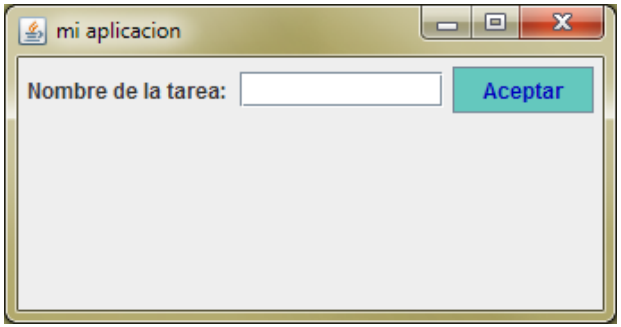
```
public class Proyecto {

   private List<Tarea> tareas = new ArrayList<Tarea>();
   // ...

   public void addTarea(Tarea t) {
      tareas.add(t);
   }
}
```

```
public class Tarea {

   private String nombre;
   // ...

   public Tarea (String nombre) {
      this.nombre = nombre;
   }

   public String getNombre() {
      return nombre;
   }
}
```



Vista

```
public class AltaTarea extends JPanel {

   private JTextField nombreTarea;
   private JButton    botonAceptar;
   // ...

   public AltaTarea () {
      // asignar layout
      // ...

      // crear componentes
      JLabel etiqueta = new JLabel("Nombre de la tarea: ");
      nombreTarea     = new JTextField(10);
      botonAceptar    = new MiBoton("Aceptar");

      // añadir componentes al panel
      this.add(etiqueta);
      this.add(nombreTarea);
      this.add(botonAceptar);
   }

   // método para asignar un controlador al botón
   public void setControlador(ActionListener c) {
      botonAceptar.addActionListener(c);
   }

   // método que devuelve el nombre de una tarea (contenido del campo JTextField)
   public String getNombreTarea () {
      return nombreTarea.getText();
   }
}
```



Controlador

```
public class ControlAltaTarea implements ActionListener {

   private AltaTarea vista;
   private Proyecto modelo;

   public ControlAltaTarea(AltaTarea vista, Proyecto modelo) {
      this.vista  = vista;
      this.modelo = modelo;
   }

   @Override
   public void actionPerformed(ActionEvent arg0) {

      // validar valores en la vista
      String nombreTarea = vista.getNombreTarea();
      if (nombreTarea.equals("")) {
         JOptionPane.showMessageDialog(vista, "Debe introducir un nombre.", "Error", JOptionPane.ERROR_MESSAGE);
         return;
      }

      // modificar modelo
      Tarea tarea = new Tarea(nombreTarea);
      modelo.addTarea(tarea);

      // mostrar nueva vista
      DetalleProyecto nuevaVista = this.getPanelDetalleProyecto();
      nuevaVista.update(tarea);
      nuevaVista.setVisible(true);
      vista.setVisible     (false); // otra posibilidad es usar un CardLayout
   }
   // ...
}
```

Poniendo todo junto…

```
public class VentanaProyectos extends JFrame {

  public VentanaProyectos () {

    // vista
    AltaTarea vista = new AltaTarea();

    // modelo
    Proyecto modelo = new Proyecto();

    // controlador
    ControlAltaTarea controlador = new ControlAltaTarea(vista, modelo);

    // configurar la vista con el controlador
    vista.setControlador(controlador);

    // ...
  }
}
```