E.T.S. INGENIERÍA INFORMÁTICA

Gestión de la Información

UNIVERSIDAD DE MÁLAGA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

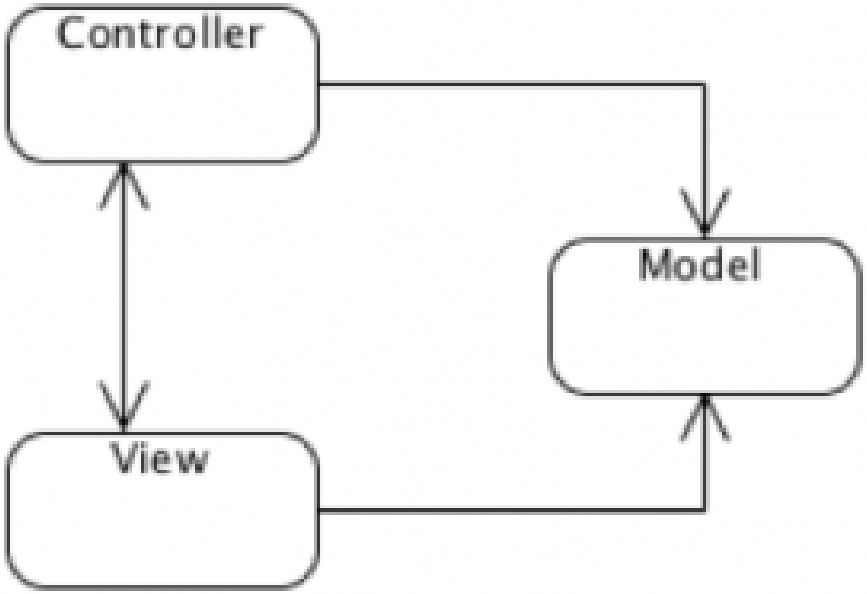
Tema 1

Arquitectura de bases de datos para el patrón MVC (Modelo-Vista-Controlador)

José Luis Pastrana Brincones (pastrana@lcc.uma.es)

El Patrón Modelo-Vista-Controlador (MVC)

- El Módelo Vista Controlador es el patrón arquitectural más usado en la ingeniería del software.
- Separa el modelos de datos, el modelo de la capa de presentación (vista) y la parte de control.



El Patrón Modelo-Vista-Controlador (MVC)

- Descrito por primera vez en 1979 para Smalltalk
  - http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html
- Utilizado en múltiples frameworks
  - Java Swing
  - Java Enterprise Edition (J2EE)
  - XForms (Formato XML estándar del)
  - GTK+ (Gnome para X Window)
  - ASP.NET MVC Framework (Microsoft)
  - Google Web Toolkit (GWT, para crear aplicaciones Ajax con Java)
  - Apache Struts (framework para aplicaciones web J2EE)
  - Ruby on Rails (framework para aplicaciones web con Ruby)
  - Etc.

El Patrón Modelo-Vista-Controlador (MVC)

- El Modelo:
  - Gestiona la información
  - Advierte a las otras capas de cambios en sus datos.
  - Representa el dominio de datos.
- La vista:
  - Representa gráficamente el modelo para que el usuario pueda interactuar él.
  - Es la interfaz de datos.
- El controlador:
  - Recibe las peticiones de la vista.
  - Responde actualizando el modelo de datos.

El Patrón Modelo-Vista-Controlador (MVC)

- Modelo–Vista–Controlador
  - Un modelo
  - Varias vistas
  - Varios controladores
- Las vistas y los controladores suelen estar muy relacionados
- Los controladores tratan los eventos que se producen en la interfaz gráfica (vista)
- Esta separación de aspectos de una aplicación da mucha flexibilidad al desarrollador

El Patrón Modelo-Vista-Controlador (MVC)

- Flujo de control

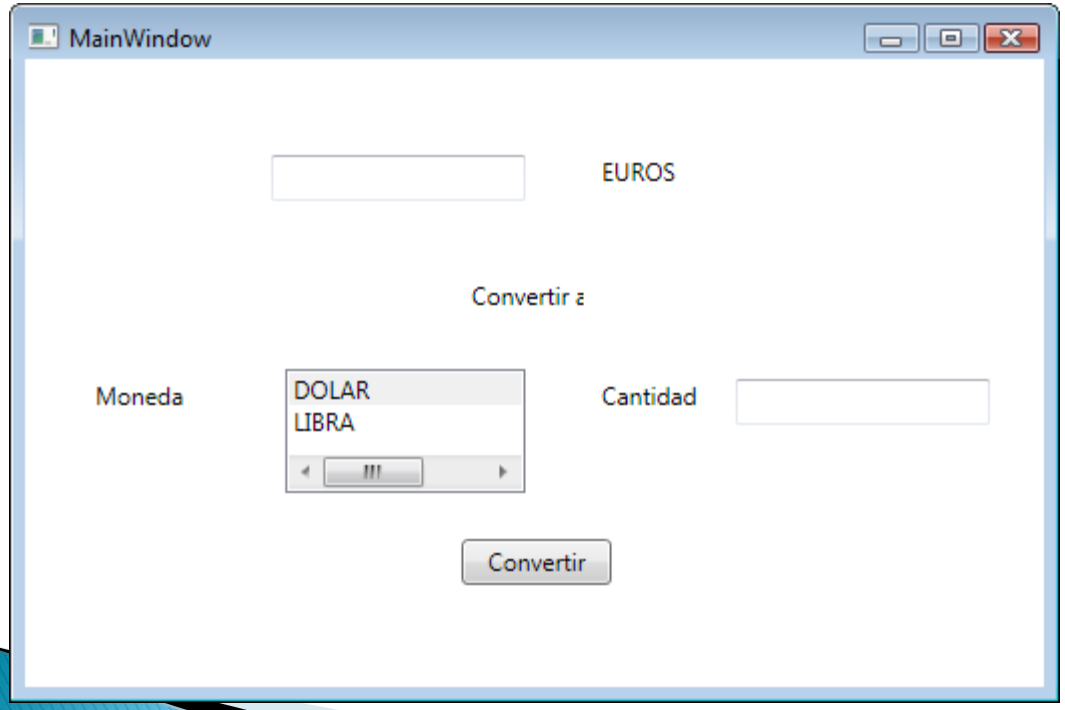
1. El usuario realiza una acción en la interfaz.
2. El controlador trata el evento de entrada.
3. El controlador notifica al modelo la acción.
   - Puede implicar un cambio del estado del modelo
4. Se genera/actualiza la vista.
   - La vista toma los datos del modelo
   - El modelo no tiene conocimiento directo de la vista
5. El interfaz de usuario espera otra interacción del usuario y comenzará otro nuevo ciclo.

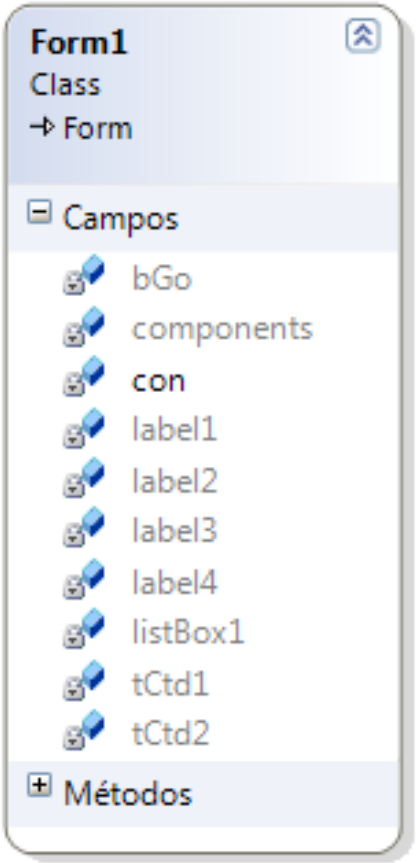
MVC en aplicaciones

- Vista:
  - página HTML, JFrame, Windows Form, WPF, …
- Controlador:
  - Hebra de tratamiento de eventos, que captura y propaga los eventos a la vista y al modelo
- Modelo:
  - Capa de Datos: información almacenada en una base de datos, en ficheros o en XML.
  - Reglas de negocio que transforman esa información.

MVC: Ejemplo

- Queremos desarrollar una aplicación para la conversión de euros a diferentes monedas.
- VISTA



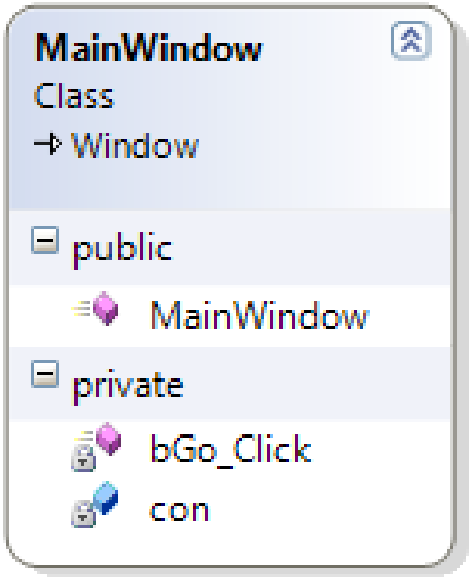


MVC: Ejemplo

- Controlador

```
public partial class MainWindow : Window
{
    Conversion con;
    public MainWindow()
    {
        con = new Conversion();
        InitializeComponent();

        foreach (string m in con.ListaMonedas())
        {
            listBox1.Items.Add(m);
        }
        listBox1.SelectedIndex = 0;
    }
    private void bGo_Click(object sender, RoutedEventArgs e)
    {
        string moneda = listBox1.SelectedItem.ToString();
        double ctd = double.Parse(tCtd1.Text);
        double res = con.Convertir(moneda, ctd);
        tCtd2.Text = res.ToString();
    }
}
```



MVC: Ejemplo

- Modelo

```
class Conversion
{
    DataClasses1DataContext dc;
    public Conversion()
    {
        dc = new DataClasses1DataContext();
    }
    public string[] ListaMonedas()
    {
        . . .
    }
    public double Convertir(string moneda, double ctd)
    {
        var val = from conversion in dc.conversion
                  where (conversion.Moneda == moneda)
                  select conversion.valor;
        double cambio = val.First();
        return ctd * cambio;
    }
}
```