

Tema V

Arquitectura de programas MVC (R-1.0)

Programación en Entornos Interactivos.

29 de febrero de 2012

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Alicante



Resumen

Modelo–Vista–Controlador. Introducción. MVC en pocas palabras.
Concepto de Modelo. Concepto de Vista. Concepto de Controlador.
Ejemplo.



Introducción.

- MVC surge junto con Smalltalk durante los años 70.
- Es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido.
- No es necesario el uso de un lenguaje orientado a objetos para emplearlo, aunque esta metodología lo haga más sencillo.

MVC en pocas palabras.

- La idea clave de MVC consiste en dividir el código de una aplicación en capas, concretamente 3:
 - ① **Modelo**
 - ② **Vista**
 - ③ **Controlador**
- Cada una de estas capas puede ser sustituida en cualquier momento sin afectar a las otras, p.e., tener diferentes vistas para un mismo modelo.
- Esta división del código garantiza mayor facilidad de portabilidad y de adaptación a los requerimientos del usuario.

Concepto de Modelo. –Capa de la Aplicación–

- Es la representación 'software' del problema a resolver, sus datos, funciones, etc. . . – personas, coches, asientos contables. . . –
- Proporciona los métodos necesarios para que:
 - Se puedan consultar los datos del modelo.
 - se puedan modificar los datos del modelo.
- Los modelos no se comunican con las vistas¹, de este modo conseguimos una mayor independencia entre el código que constituye cada capa.
- Un modelo puede tener asociadas varias vistas.

¹Aunque en ocasiones determinadas puede resultar interesante.

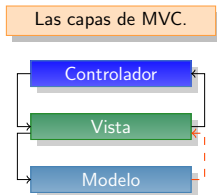
Concepto de Vista. –Capa de Presentación–

- Sirve para mostrar al usuario los ‘datos’ del modelo que le interesan en cada caso –el nombre de una persona, la velocidad de un coche, el importe de un asiento. . . –
- Una vista no tiene pq ser solamente en modo gráfico, puede ser en modo texto. . .
- Las vistas se comunican con los modelos de forma bi-direccional –solicitan información, pueden modificar información–
- Las vistas se pueden ‘anidar’ dando lugar a lo que se llama una vista principal –*top-view*– compuesta de subvistas.

Concepto de Controlador. –*Capa de Interacción*–

- Contiene el código que hace de interfaz entre los dispositivos de entrada –teclado, raton, etc. . . – y las capas de Vistas y Modelos.
- Este es el código que permite al usuario interactuar con las Vistas.
- Normalmente no tendremos que escribir código relacionado con esta capa ya que el código que iría aquí es el que proporciona la biblioteca gráfica utilizada (Xforms, Qt, Gtk, etc. . .).

Una imagen vale más que...



La conexión entre el modelo y las vistas asociadas sólo tiene sentido si los datos del modelo se van a modificar internamente por algún cálculo y queremos que se actualicen las vistas asociadas a ese modelo.

Ejemplo. (I)

- En este ejemplo tenemos un **modelo** de una **Persona**.
- La información relativa a una Persona en esta aplicación consta de su **nombre** y su **edad**.
- Sobre este modelo se definen **dos vistas**:
 - ① **Vista** de texto **ascii**: sólo permite consultar el estado del Modelo.
 - ② **Vista** que hace uso de la biblioteca **GTK** para mostrar el estado del Modelo y **permitir al usuario cambiar el nombre y la edad de la Persona** reflejada en esta vista.

Ejemplo. (II)

- La comunicación entre las Vistas y el Modelo puede ser para solicitar información o para actualizar la información del Modelo con la que el usuario ha cambiado en la Vista.
- Se permite la comunicación entre el Modelo y las Vistas asociadas a él sólo para indicarles que el estado ha cambiado y que deben actualizar lo que muestran al usuario.

