

Práctica 8

1 FICHA TRABAJO

Autor	Sergio Martín Santana
Fecha entrega y defensa	13/01/2016
Tiempo Necesario para realizar el trabajo (Aproximado)	
<i>Realización de trabajo de búsqueda de información</i>	8 h
<i>Codificación del problema</i>	4 h
<i>Realización del informe</i>	2 h
Total	14 h

2 INTRODUCCIÓN

El modelo MVC (Modelo-Vista-Controlador), es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

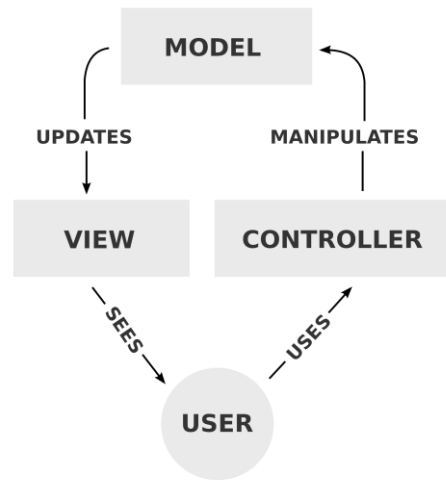
De manera genérica, los componentes de MVC se podrían definir como sigue:

- El **Modelo**: Es la representación de la información con la cual el sistema opera.
- El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.
- La **Vista**: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

3 CÓDIGO CREADO

EL código generado no se diferencia en gran medida del generado para la práctica 7, salvo que en este caso hemos agregado una nueva clase e interfaz. Controlador e iControlador, para crear con ellas el controlador ya que tanto la vista como el modelo estaba ya creado anteriormente. Una vez creado hemos realizado tres funciones en la clase Controlador: update, Get Event y setModel.

- Update, simplemente actualiza la vista cuando existe algún cambio en la misma.
- GetEvent, cuenta con una cláusula if else, donde dependiendo el evento que se le pase, realiza una funcion (las funciones vienen a ser las que se realizaban dentro del action listener del componente grafico)
- SetModel, crea el modelo cuando se inicializa la aplicación.



4 INTERFAZ

