

Modelo-Visor-Controlador

José Luis León Bayas

October 2016

Es un patrón que permite trabajar por separado en las diferentes áreas de un programa, entre las cuales estan: el modelo, la vista y el controlador.

Nos permite enfocar nuestra máxima atención en cada uno de los componentes y tener un código más organizado y limpio, además nos permite reutilizar código y el mantenimiento o cambio es mucho más facil, también nos ofrece la ventaja de ser multiplataforma.

1. Modelo

Esta sección de un programa contiene la estructura o núcleo del mismo, con ello determina la funcionalidad del programa ya que conlleva las principales instrucciones y tratamiento de datos, mantiene el estado actual del mismo y además es independiente de la vista y el controlador, ello implica no conoce nada acerca de la vista y el controlador, sin embargo se puede encontrar relacionado con el controlador de modo que con la interacción del usuario se pueda cambiar el estado del modelo o crear uno nuevo para la situación, y con ello cambiar la vista. [2] [3]

Es la parte donde se realizan los cambios en caso de ser necesarios, ya que esto modifica a las dos partes consiguientes, logrando de este ser un modelo más eficiente y fácil de cambiar.

2. Vista

Esta parte se enfoca principalmente en la presentación de los datos de la aplicación, es su apariencia y la representación del modelo que se muestra al usuario. [1]

La vista debe cambiar de acuerdo a como cambie el modelo (y es allí donde surge la única interacción entre ambos), para ello existen dos métodos:

1. Push: los valores se guardan tanto en la vista como en el modelo de este modo la vista cambia a la par que el modelo. [1]
2. Pull: la vista comprueba continuamente los cambios del modelo para de este modo cambiar con ella. [1]

3. Controlador

El controlador es el encargado de leer y traducir las interacciones del usuario con la vista de modo que los valores o interacciones se posicionen en el modelo logrando de este

modo y de ser necesarios cambiar el mismo lo que conlleva a un cambio en la vista del usuario. [1]

Referencias

[1] bferro. Aplicando mcv en java.

[2] N Gotooru. What is mvc pattern?

[3] Universidad Carlos III. Patron de arquitectura modelo vista controlador (mvc).