

Modelo Vista Controlador


Dpto. Lenguajes y Sistemas Informáticos
Universidad de Granada

Programación y Diseño Orientado a Objetos

Créditos


- Las siguientes imágenes e ilustraciones son libres y se han obtenido de:

- ▶ Emojis, <https://pixabay.com/images/id-2074153/>

- ▶  <https://pixabay.com/images/id-1299287/>

- ▶  <https://pixabay.com/images/id-341444/>

- ▶  <https://pixabay.com/images/id-1020156/>

- ▶  <https://pixabay.com/images/id-3383459/>

- El resto de imágenes e ilustraciones son de creación propia, al igual que los ejemplos de código

Objetivos

- Entender qué es un patrón de diseño y su utilidad
- Entender el patrón Modelo Vista Controlador y el reparto de responsabilidad que realiza

Contenidos

1 Patrón de diseño

2 Modelo Vista Controlador

Patrón de diseño

- Un patrón de diseño describe un problema que ocurre numerables veces en nuestro entorno
- Describe además el núcleo de una solución a ese problema de forma que sea reutilizable
- Permite aprovechar soluciones previas probadas y validadas a problemas conocidos

Patrón de diseño



=



=



Modelo Vista Controlador (MVC)

- Es un patrón de tipo arquitectónico, al igual que la programación por capas, arquitectura orientada a servicios, etc.
- Elementos:

Modelo: Clases que representan la lógica del problema

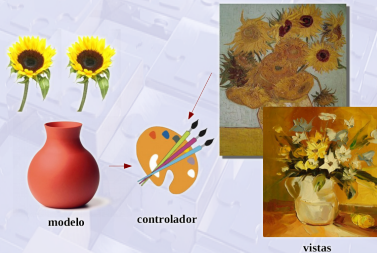
Vista: Una representación visual de los datos del modelo para mostrarlos al usuario

- ▶ La interacción del usuario se produce con elementos de la vista

Controlador: Actúa de intermediario entre la vista y el modelo

Modelo Vista Controlador

- Para un mismo modelo se pueden tener diferentes vistas



- La información fluye en ambas direcciones

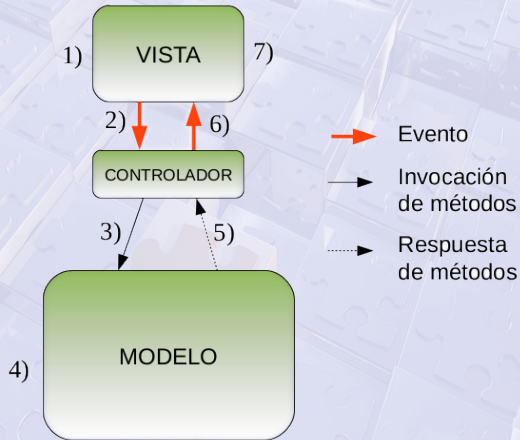


La responsabilidad del Controlador

- Ante una orden del usuario, siempre a través de la vista, que implique cambios en el modelo
 - ▶ Es el controlador quien la ejecuta actuando sobre el modelo
- ← Cuando se producen cambios en el modelo, estos cambios deben verse reflejados en las vistas correspondientes
 - ▶ El controlador puede actuar de intermediario en este proceso
- ⇔ La mayoría de las veces, una acción del usuario implica un recorrido de *ida y vuelta*.

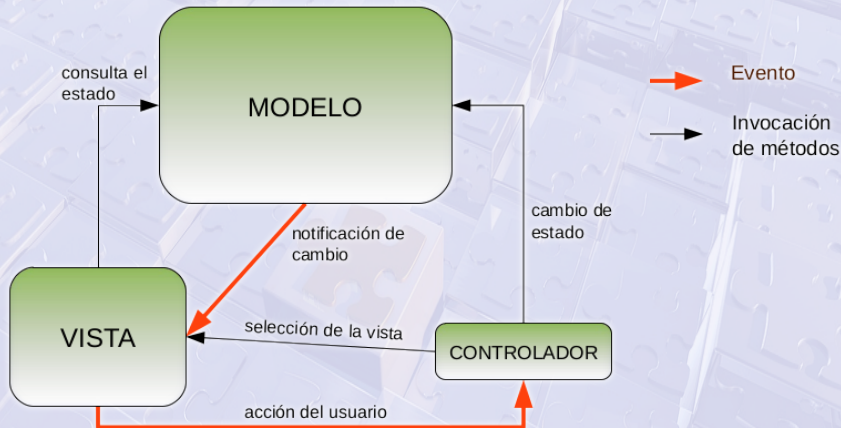
Esquema básico de MVC

- Sin comunicación directa entre modelo y vista



Esquema alternativo de MVC

- El modelo puede informar directamente a la vista para que se actualice



Modelo Vista Controlador

→ *Diseño* ←

- En lecciones anteriores se han comentado los principios de alta cohesión y bajo acoplamiento
- No solo se tienen en cuenta en el diseño a nivel de clases
- También deben contemplarse a otros niveles
- Esta arquitectura Modelo Vista Controlador es un ejemplo de ello

Modelo Vista Controlador

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Granada

Programación y Diseño Orientado a Objetos