

T3.2 PATRÓN MODELO-VISTA-CONTROLADOR

El patrón Modelo-Vista-Controlador se originó para implementar interfaces de usuario en los que las responsabilidades están bien distribuidas entre distintas partes (componentes) del diseño. Así, se decidió distinguir tres responsabilidades distintas:

- Lógica de negocio: **Modelo**.
- Presentación: **Vista**.
- Gestión de eventos de usuario: **Controlador**.

Modelo.

Contiene el núcleo de la funcionalidad (dominio) de la aplicación y representa la lógica de negocio de la aplicación. Encapsula el estado de la aplicación, ya que encapsular el modelo de la aplicación en componentes facilita la depuración, mejora la calidad y favorece la reutilización de código. El Modelo es independiente de la Vista y el Controlador.

Vista.

Representa la lógica de presentación de la aplicación; es la presentación del modelo. Los componentes de la vista extraen el estado actual del sistema del modelo y proporcionan la interfaz del usuario. La Vista puede acceder al Modelo, pero nunca cambiar su estado, y puede ser notificada cuando hay un cambio de estado en el Modelo.

Controlador.

Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente. Es responsable de recibir los eventos, determinar el procesador del evento, invocar al procesador y finalmente provocar la generación de la vista apropiada. En una aplicación web Java la tecnología más adecuada para implementar los controladores son los Servlets.

