4. Patrón MVC

El patrón MVC (Modelo-Vista-Controlador) es un patrón de arquitectura de software que separa los datos de una aplicación, la lógica de negocio y la interfaz de usuario. Su uso está muy extendido porque mejora la calidad y eficiencia del código.

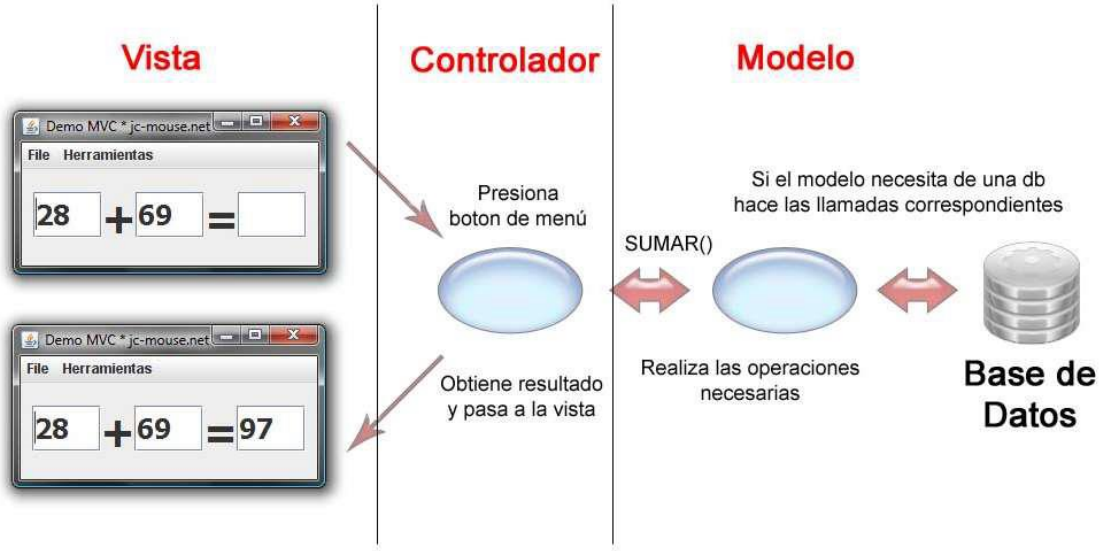
Este patrón organiza una aplicación en tres bloques, que son independientes entre sí:

- Modelo: representa la lógica de la aplicación y sus datos.
- Vista: representa la parte del código con la que interacciona el usuario. Es decir, está formada por las diferentes interfaces de nuestra aplicación.
- Controlador: es el bloque que se encarga de comunicar la vista con el modelo.

Entre las ventajas de este patrón encontramos:

- Hay una separación entre el almacenamiento de los datos y su representación visual.
- Facilita la reutilización de código.
- Facilita el mantenimiento, porque los cambios que se produzcan en una parte de la aplicación no afectarán a las demás.

La siguiente figura describe el flujo de una aplicación diseñada bajo el patrón MVC:



El funcionamiento sería el siguiente:

1. El usuario realiza una petición (evento) en la interfaz de la aplicación (pulsa un botón, un link…). Esta petición es recogida por la vista y la envía al controlador.

2. El controlador examina la petición y decide lo que debería hacer la aplicación. Entonces, el controlador llama al modelo para que ejecute una acción determinada.
3. El modelo ejecuta la acción (llamando a la base de datos si es necesario) y le devuelve el resultado al controlador.
4. El controlador le indica a la vista los resultados que debe mostrar.
5. La vista muestra estos resultados en la interfaz gráfica, de forma que el usuario los pueda visualizar.

EJERCICIOS PROPUESTOS

1. Desarrolla una aplicación, siguiendo el patrón MVC, que sume dos números y muestre el resultado. Apóyate en la interfaz **VentanaSuma.java**

2. Desarrolla una aplicación, siguiendo el patrón MVC, que muestre los datos de un empleado con un número determinado. Apóyate en la base de datos **empleados** y en la interfaz **VentanaDatosEmpleado.java**.