

Hoja de ejercicios 3. Patrón MVC

Ejemplo 1: proyecto MVC1.

MVC para una aplicación de escritorio con un formulario y sin base de datos.

Construir un programa para realizar la suma de dos números enteros con las siguientes características:

- La vista será un formulario con dos `JText` para escribir los números, un botón de sumar y un `JText` no editable para mostrar el resultado. En la vista haremos que los controles tengan acceso público para manipularlos desde el controlador.
- El modelo se encargará de almacenar los números, tendrá los `getter` y `setter` y un método para realizar la operación deseada.
- El controlador conectará la vista con el modelo. Es necesario:
 - o Que el constructor inicialice a la vista y al modelo. Posteriormente iniciar la variable que permite el acceso tanto a la vista como al modelo.
 - o Crear un método que inicialice a la vista.
 - o Implementar la interfaz `ActionListener`, para que la clase esté preparada para “escuchar” las acciones de la vista.
 - o Sobrescribir el método `actionPerformed`, que puede reconocer que componente de la vista generó la acción a través de la variable `ActionEvent`.

Hacer el ejemplo paso a paso.

¿Qué ventaja aporta? Separar la interfaz de usuario del modelo de almacenamiento facilita la reusabilidad del código y la localización de errores.

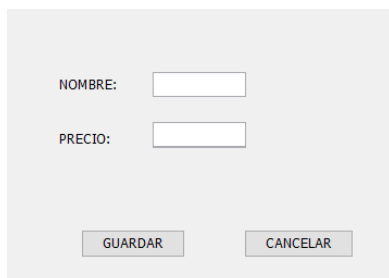
EJERCICIO:

Vamos a realizar una pequeña aplicación para gestionar los productos de una tienda.

Los productos se almacenan en un fichero de nombre “productos.dat”.

De cada producto nos interesa conocer: Nombre. Precio.

1. Escribe un programa que sea capaz de almacenar o guardar un producto en el fichero. En la ventana o `JFrame` se pueden introducir el nombre y el precio y guardarlos en el fichero.



The image shows a simple Java Swing window with a light gray background. It contains two text input fields. The first field is preceded by the label 'NOMBRE:' and the second by 'PRECIO:'. Below these fields are two buttons: 'GUARDAR' on the left and 'CANCELAR' on the right.

2. Incorpora un `JList` que liste o muestre los productos almacenados en el fichero.