***MODELO VISTA CONTROLADOR***

**Problema que aborda**

Algunas veces podemos ver que todo nuestro proyecto está desarrollado en una misma clase, librería, página…etc. Con esto conseguimos que nuestra aplicación sea muy difícil de mantener e incluso casi imposible de reutilizar.

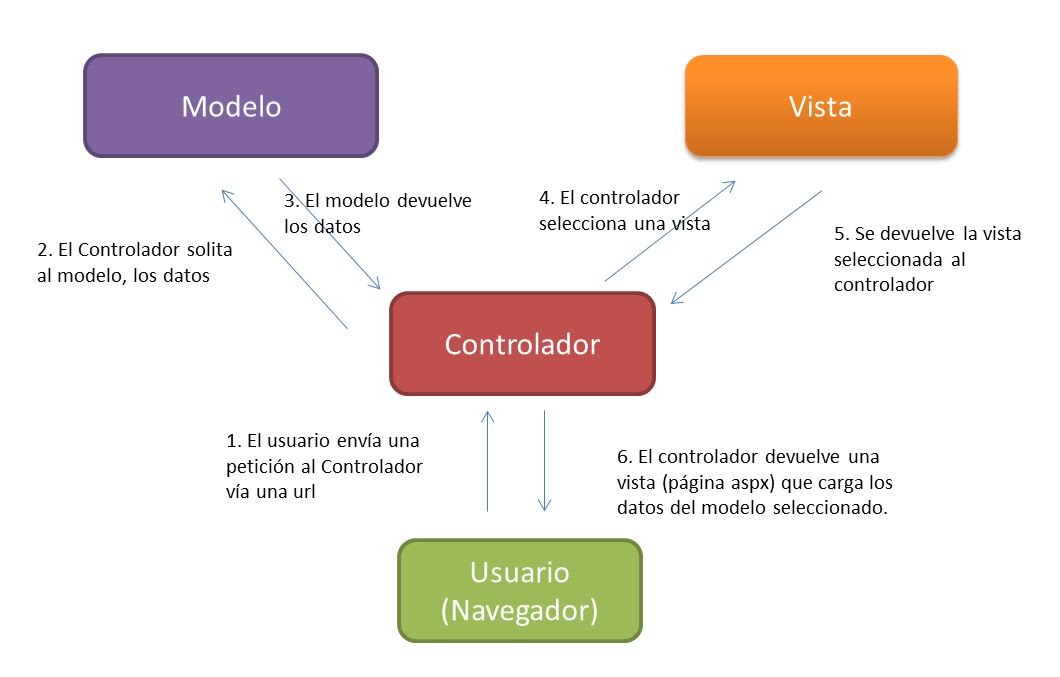
**Solución propuesta**

Con el modelo vista controlador, lo que conseguimos es separar las diferentes responsabilidades que tiene una aplicación en elementos de tres tipos:

-**Modelo**: contiene una representación de los datos que maneja el sistema y su lógica de negocio.

-**Vista**: es la interfaz de usuario, que se compone de la información que se envía al cliente y de los mecanismos de interacción.

-**Controlador**: actúa de intermediario entre el modelo y la vista, gestionando el flujo de información entre ellos.



**Consecuencias**

Ventajas:

-Separación clara de dónde tiene que ir cada tipo de lógica.

-Sencillez para crear distintas representaciones de los mismos datos.

-Facilidad para la realización de pruebas unitarias.

-Reutilización de los componentes.

Desventajas:

-La cantidad de archivos que se deben mantener incrementa considerablemente.

-La curva de aprendizaje es más alta que utilizando otros modelos.

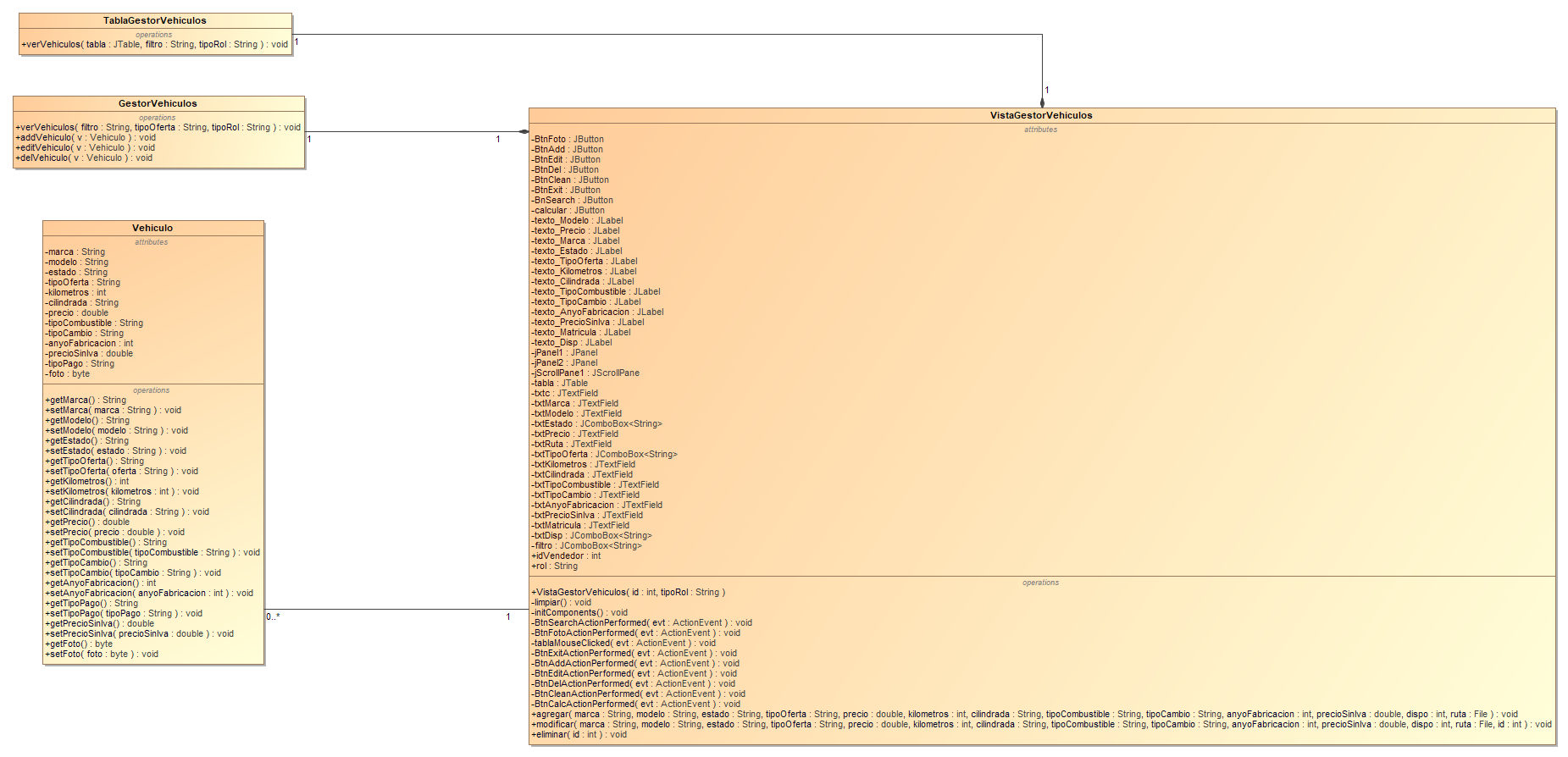
-Su separación en capas, aumenta la complejidad del sistema.

**Aplicación a nuestro proyecto**

Dentro de nuestro proyecto podemos ver:

-Modelo: podemos ver el paquete llamado modelo en el que se encuentran las clase base de nuestra aplicación como Vehiculo, Usuario, Operacion, GestorVehiculos, GestorUsuarios, GestorOperaciones.

-Vista, controlador: la vista la podemos ver en el paquete vista, el controlador lo tenemos integrado junto a las siguiente clases, que serían las que usan este modelo; VistaGestorVehiculos, VistaGestorUsuarios, VistaGestorOperaciones, VistaGestorVehiculosMecanico, VistaVenta, VistaVenta2, VistaAlquiler, VistaAlquiler2.



En la imagen podemos ver una parte de nuestro diagrama de clases donde se aplica el modelo vista controlador, tenemos las clases de vehículo y GestorVehículos que estarían dentro del modelo como clases base.

A continuación, tenemos la clase VistaGestorVehículos, que en nuestra app actuaría como Vista, y dentro estaría integrado el controlador que haría de intermediario entre el modelo y la vista.