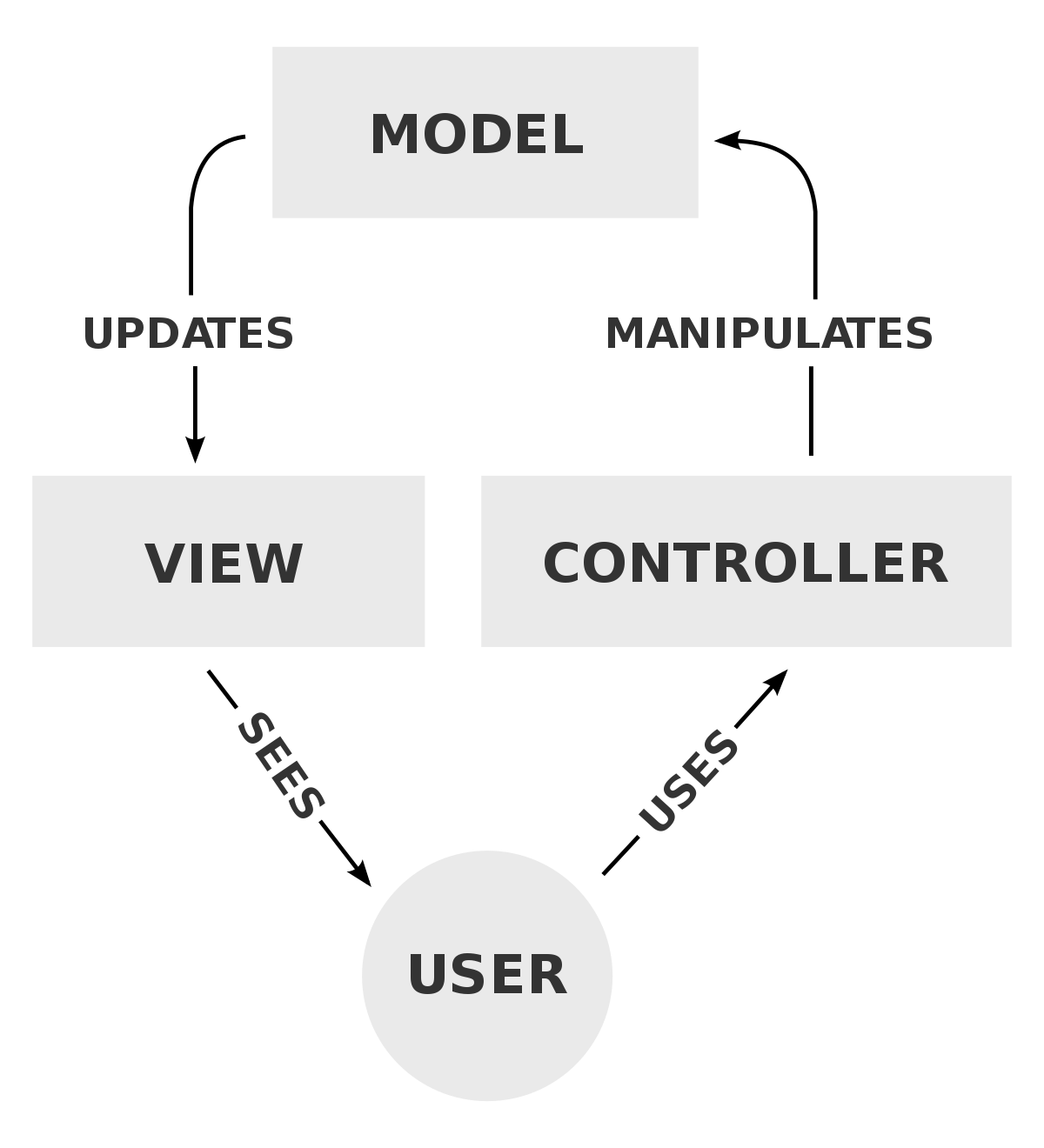
# **¿Qué es MVC?**

MVC es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

A continuación, veamos un esquema que indica la separación y organización de este patrón de arquitectura de software:



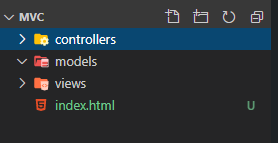
Ahí podemos ver lo siguiente; el usuario usa el controlador, este controlador manipula el modelo y el modelo actualiza la vista la cual es vista por el usuario, claro que a veces el controlador actualiza la vista directamente, pero cuando se trata de manipular data que esté directamente relacionada con información en la base de datos, por ejemplo, el modelo, luego de ser actualizado por el controlador, refleja este cambio en la vista.

Ahora definamos los pilares de este patrón arquitectónico:

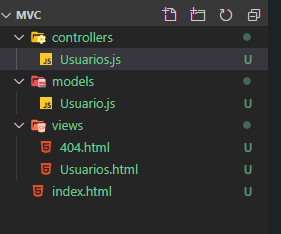
* Modelo – Es la representación de la información con la cuál la aplicación opera, gestiona los accesos a dicha información, tanto lectura como actualización
* Controlador – Responde a eventos (usualmente acciones del usuario) e invoca peticiones al “modelo” cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su “vista” asociada si se solicita un cambio en la forma en que se presenta el “modelo”
* Vista: Presenta el “modelo” (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho “modelo” la información que debe representar como salida.

# **Ejemplo del patrón de arquitectura MVC**

Organización de carpetas en el modelo MVC, fíjese cómo el editor de texto reconoce esta carpeta como parte del modelo MVC y actualiza los íconos para representar el nombre que lo describe.

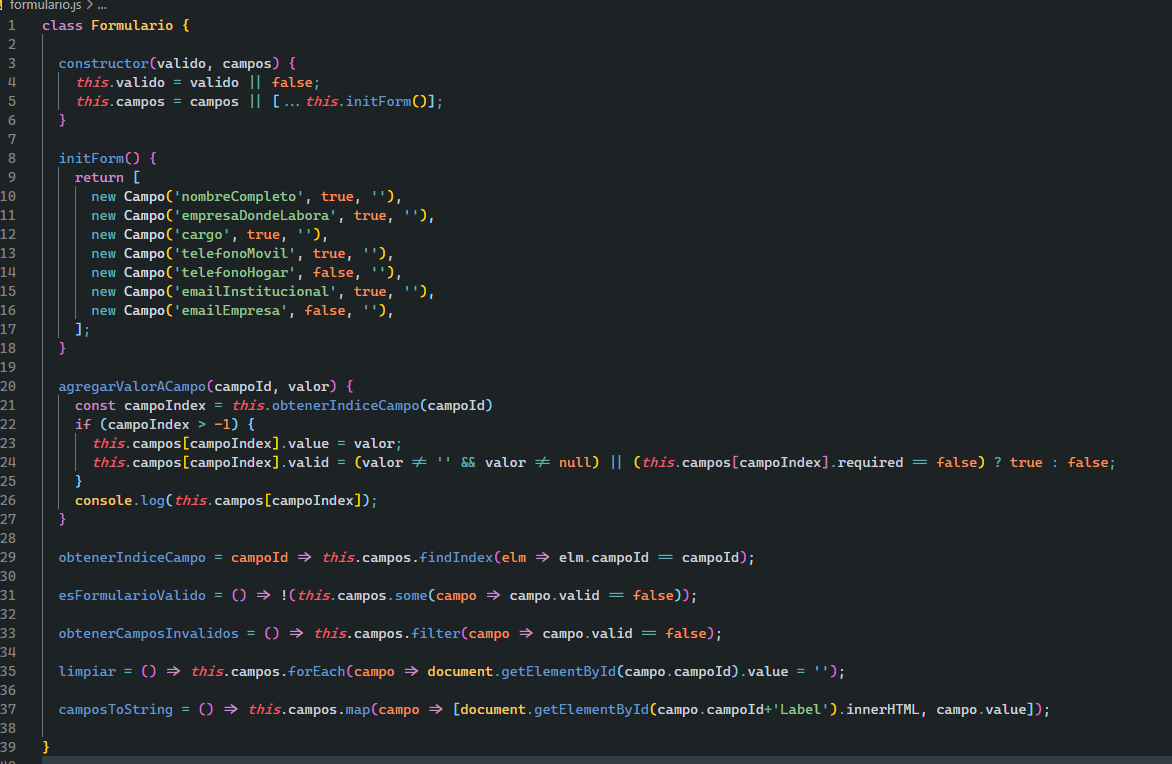


A continuación, la entidad Usuario presente en el modelo MVC, donde se puede ver una vista "Usuarios", un controlador “Usuarios”, donde hay métodos diferentes y el modelo “Usuario” que describe la entidad usuario.

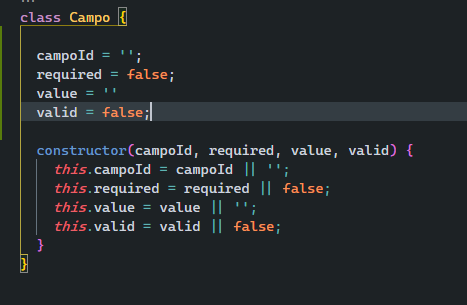


Veamos otro ejemplo:

Aquí podemos ver el controlador de una entidad **Formulario**, instanciando el modelo "**Campo**" con diferentes métodos para manipular el modelo "**Campo**":



A continuación, el modelo "**Campo**":



La vista, donde se presenta dicho campo se manipula de forma dinámica, y es parte del **index.html**