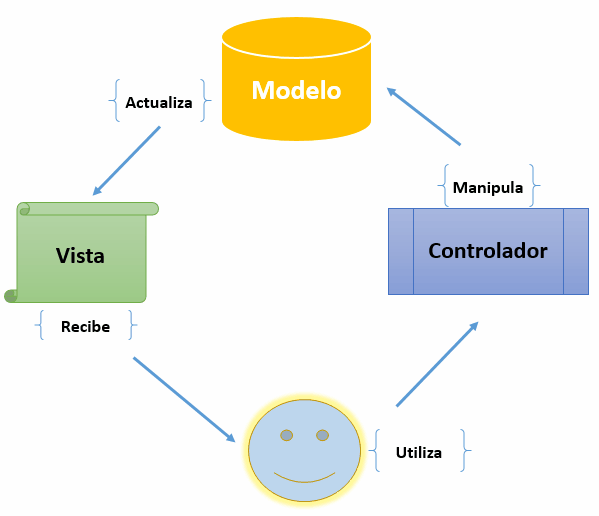
**El modelo MVC (Modelo-Vista-Controlador)**

El modelo MVC fue algo ideado en la década de los 70 para el lenguaje *SmallTalk.* Se ha estado usando en aplicaciones de escritorio para ese y otros lenguajes, pero su aparición en la Web no ha sucedido hasta hace poco. **La idea central detrás de MVC es que se establece una separación clara  entre la gestión de sus datos (el modelo), la lógica de la aplicación (el controlador), y la presentación de los datos para el usuario (la vista**).

El esquema de la figura adjunta explica el circuito de interacción que sigue habitualmente este modelo desde el punto de vista del usuario



La vista obtiene del modelo los datos para mostrar al usuario. Cuando un usuario interactúa con la aplicación, el controlador responde cambiando los datos en el modelo. Por último, el modelo notifica a la vista el cambio producido para que ésta pueda actualizar lo que muestra en la salida.

En la arquitectura propuesta por AngularJS, la vista es el *Document Object Model* (DOM), los controladores son clases de JavaScript, y los datos del modelo se almacenan en propiedades de objetos del DOM.

MVC ofrece una arquitectura ordenada por varias razones: En primer lugar, ofrece un modelo mental por donde empezar, por lo que no tenemos que reinventarlo cada vez. Además, otros miembros del equipo de desarrollo contarán con un punto de partida rápido para comprender lo escrito al haber utilizado la arquitectura MVC al organizar el código. Y esto supondrá innumerables beneficios a la hora de mantenerlo, modificarlo y probarlo.

**Resumiendo las ventajas del modelo MVC**

La **arquitectura MVC** ofrece las siguientes **ventajas**:

* Favorece la implantación del principio de **"Separación de Responsabilidades"** (*Separation of Concerns*).
* **No utiliza el estado de las vistas** o formularios de servidor. Esto hace que el *framework* MVC sea ideal para los desarrolladores que desean un control completo sobre el comportamiento de una aplicación.

**Nota**: En .NET ya está disponible una versión MVC, probada y optimizada para el desarrollo.

* **Se utiliza un patrón *Front Controller*** que procesa las solicitudes de aplicaciones Web a través de un solo controlador. Esto permite diseñar una aplicación que es compatible con una rica infraestructura de enrutamiento (*routing*).
* Proporciona un **mejor soporte para el desarrollo basado en pruebas** (**TDD**). Funciona muy bien en aplicaciones web soportadas por grandes equipos de desarrolladores y diseñadores web que necesitan un alto grado de control sobre el comportamiento de la aplicación.