***DataBinding* (Enlace a datos)**

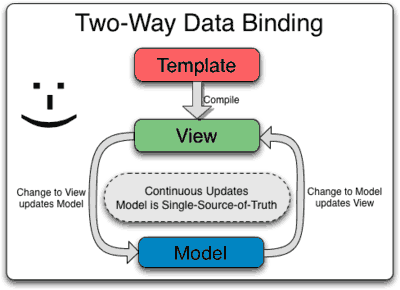
Antes de la aparición de AJAX, era común que las plataformas de servidor fundiesen cadenas HTML con datos antes de enviarlas al cliente. Sobre todo a partir de la aparición de *jQuery*, esto se extendió a los clientes, al permitirnos actualizar solamente las partes necesarias de la página, utilizando principalmente la propiedad *innerHTML* de los elementos del DOM.

El problema surge cuando queremos insertar datos nuevos o actualizados en la Interfaz de Usuario (IU), o cambiar los datos basándonos en las entradas de éste, ya que se precisa un esfuerzo extra para asegurarnos de que los datos estén es un estado correcto, tanto en la IU como en las propiedades JavaScript.

La solución sería poder declarar qué partes del DOM están vinculadas a otras de JavaScript, y hacer que esta sincronización tuviese lugar de forma automática (al estilo de lo que hacen WPF o Silverlight). Y eso es lo que encontramos en **AngularJS**. **La mayor parte del trabajo de mover datos de una capa a otra sucede de forma automática**.

En el modelo de programación con AngularJS, este **enlace a datos tiene lugar mediante la sintaxis *moustache* y se establece en la vista**, de forma que, cuando se recibe una actualización del modelo, se produce internamente un evento, que provoca la actualización de los datos y su consecuente refresco en la Interfaz de Usuario.

Pero Angular hace algo más. Y es que el modelo de *DataBinding* en **AngularJS es bidireccional**. Esto significa que cuando -a su vez- el usuario modifica algo en la interfaz que le compete al modelo de datos, este se actualiza igualmente, como podemos ver en el gráfico adjunto, que encontramos en la documentación oficial del sitio de AngularJS.



**Ventajas del enlace de datos**

Resumiendo las ventajas del enlace a datos, podemos apoyarlo en los siguientes puntos:

* Una aplicación basada en datos se puede crear de forma rápida y eficiente mediante esta técnica.
* La cantidad de código disminuye increíblemente.
* Aumenta la calidad de la aplicación.
* Mediante el uso de eventos se puede obtener el control sobre el proceso de enlace de datos.
* La codificación de la parte de acceso a datos es fácilmente manejable y actualizable.
* Los equipos de programadores obtienen rápidos beneficios de este enfoque y los test se simplifican.