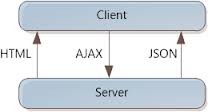
**Aplicaciones Web**

La razón principal del crecimiento de las aplicaciones Web en los últimos años, es ese nuevo concepto de "navegador como plataforma". El mundo de la industria asume que el "sueño de las máquinas virtuales", o sea, la idea de que arquitecturas como J2EE o .NET funcionen en todos los sitios era -en realidad- un sueño imposible. Y más, cuando la barrera entre los dispositivos móviles y los de escritorio se difumina, como sucede hoy en día.

Por otro lado, la nube permite acceso a contenidos desde no importa qué dispositivo y, lo que prima, es el contenido mismo. De ahí la adopción del principio de "hechos consumados". Ni se puede, ni se quiere romper con la inmensa cantidad de información existente. Tenemos que ser capaces de usarla y explotarla, de un modo lo más unificado posible. (De hecho, esto tienen implicaciones en otras áreas como lo que hoy denominamos "*Big Data*". Pero eso es otra historia...).

De esta forma, HTML5 y sus tecnologías supuso el primer paso. Pero crear aplicaciones y gestionar el proceso de construcción como el de cualquier otra aplicación actual, requiere algo más: código reutilizable, sitios y aplicaciones de tipo "*Responsive*", manejo de pruebas unitarias, etc.

Esto dio paso a un nuevo modelo de aplicaciones, el **SPA (*Single Page Applications***), más adecuado y más adaptado a estos principios. Además, los **formatos de transferencia de información** tienden a optimizarse, utilizando propuestas como **JSON**, y los cambios en las **páginas *responsive*** se producen principalmente sobre la información modificada (**peticiones AJAX**).

**AngularJS como solución**

Un grupo de desarrollo de Google se planteó la necesidad de unificar los esfuerzos previos en este sentido (los de otras librerías, como [*Knockout*](http://knockoutjs.com/#_blank)*,* [*Node.js*](http://nodejs.org/#_blank)o[*Ember*](http://emberjs.com/#_blank)), y crear un nuevo conjunto de librerías que recogiese todas las necesidades actuales, y de paso, que lo hiciese permitiendo una fácil integración con los principios de "buenas prácticas" que hoy consideramos fundamentales.

Está creada bajo los principios de que la programación declarativa es ideal para la construcción de interfaces de usuario, mientras que la programación imperativa lo es de cara a la creación de la lógica de negocio. Para los desarrolladores de otros sistemas, como .NET, esto les recordará claramente los principios en los que se basa WPF (*Windows Presentation Foundation*) o el "difunto" Silverlight. Como resultado, se reduce la manipulación del DOM y se incrementan las capacidades de testeo de las aplicaciones. En resumen, **AngularJS aporta**:

* **Visualización óptima** de la información
* Facilidad de lectura y navegación
* **Independencia del dispositivo**
* **"*Feedback*" continuo** (cualquier acción supone una respuesta)
* **Gestión diferida** al cliente, aprovechando los nuevos motores de JavaScript, que encontramos en los navegadores modernos.

**Elementos básicos de Angular**

Para conseguir estos propósitos las librerías de AngularJS ofrecen un conjunto de archivos de código abierto, desarrollados y mantenidos por programadores de Google, con un sitio Web de referencia: [http://angularjs.org](http://angularjs.org#_blank). Allí, podemos encontrar toda la documentación de las API que forman el marco de trabajo de AngularJS, los elementos descargables, el histórico de revisiones y muchas cosas más.

Por completar esta presentación de la librería, podemos resumir **las características más interesantes** de su funcionamiento en estos puntos:

* Un conjunto de librerías con **garantías de mantenimiento y actualización** (por Google) y **libre de royalties**, pensado para las aplicaciones Web
* Un sistema de **enlace a datos** (*DataBinding*) potente y modular que permite el testeo de aplicaciones
* **Control de las estructuras** del DOM
* **Fácil vinculación** de código a elementos del DOM
* Agrupación de elementos HTML en **controles reutilizables** (al estilo de lo que permiten los controles .ASCX de ASP.NET)

Con todo esto, tenemos lo fundamental para la construcción de aplicaciones CRUD (*Create, Read, Update, Delete*), que es el objetivo fundamental de AngularJS. Para otro tipo de aplicaciones de internet, como los juegos o los editores, que precisan una manipulación muy fina del DOM, disponemos de otras librerías como [*jQuery*](http://jquery.com/#_blank), [*Raphael*](http://raphaeljs.com/#_blank), etc.