

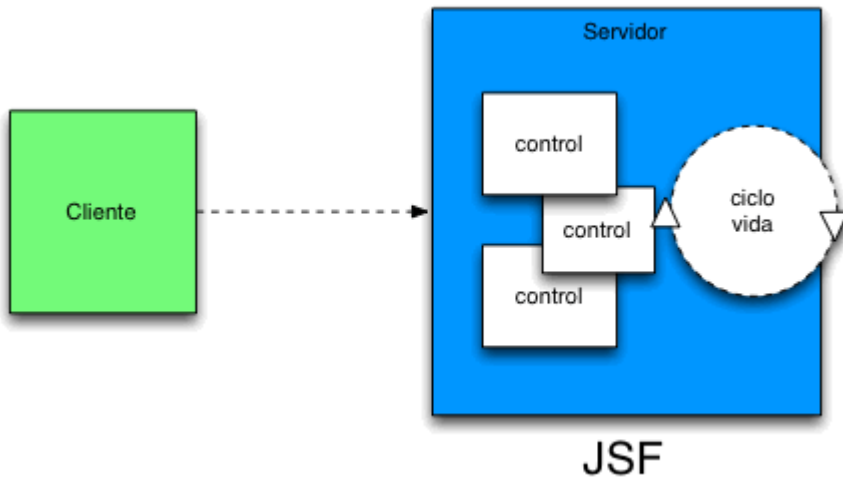
Todos hemos trabajado en unos u otros proyectos con JSF y su tecnología orientada a controles. Una pregunta que cada día me viene más a la cabeza es: ¿Cual es el futuro de JSF?. Para muchas personas la respuesta tiende a ser sencilla . JSF es el estandar de Java EE a nivel de capa de presentación por lo tanto es el futuro. Sin embargo ya hemos tenido en otras ocasiones standards de la plataforma Java cuyo que han pasado a mejor vida o han tenido que reinventarse de forma significativa. Quizás para poder valorar más cual es el estado de JSF debemos realizar una consulta en [Google trends](#) (tendencias) y obtener el siguiente resultado.



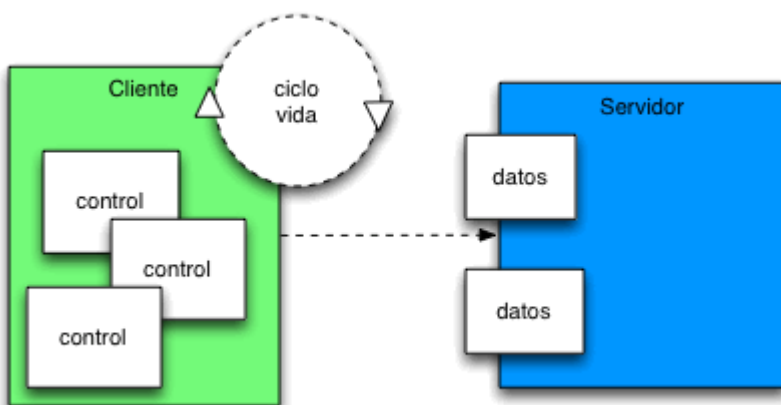
Es evidente que el interés por JSF esta decreciendo dentro de la comunidad. ¿Porqué? esa es una buena pregunta ya que recordemos que JSF es el standard y ha evolucionado de una forma bastante sólida e interesante durante los últimos años . Por decirlo de alguna forma tendríamos que esperarnos una gráfica completamente al revés. ¿Qué es lo que ha pasado?.

JSF y Servidor

JSF siempre ha gestionado su ciclo de vida a través del servidor construyendo los distintos tipos de controles que necesitamos usar. Dejando al cliente unas responsabilidades bastante reducidas

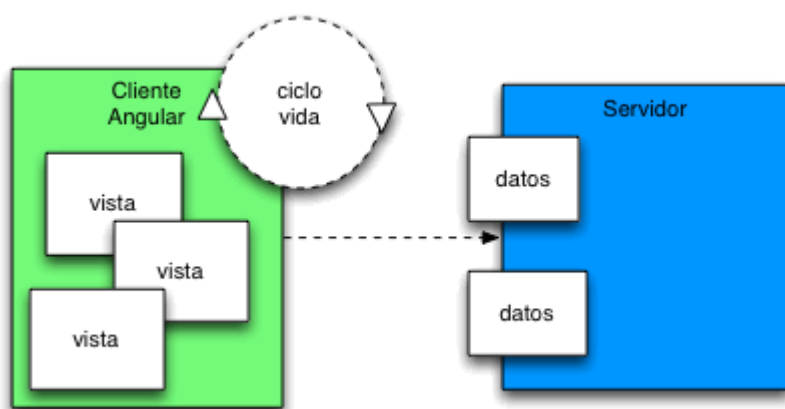


Hoy por hoy muchas aplicaciones han optado por delegar una gran parte de las responsabilidades al cliente . Ejemplos claros son las plataformas móviles con HTML5 ,Arquitecturas SPA y frameworks como JQuery Mobile que usan simplemente el servidor para nutrirse de datos pero que todos los controles vienen definidos directamente en el cliente. Ante este tipo de situaciones JSF pierde parte de su interés.



Framework MVC cliente (Angular, Backbone etc)

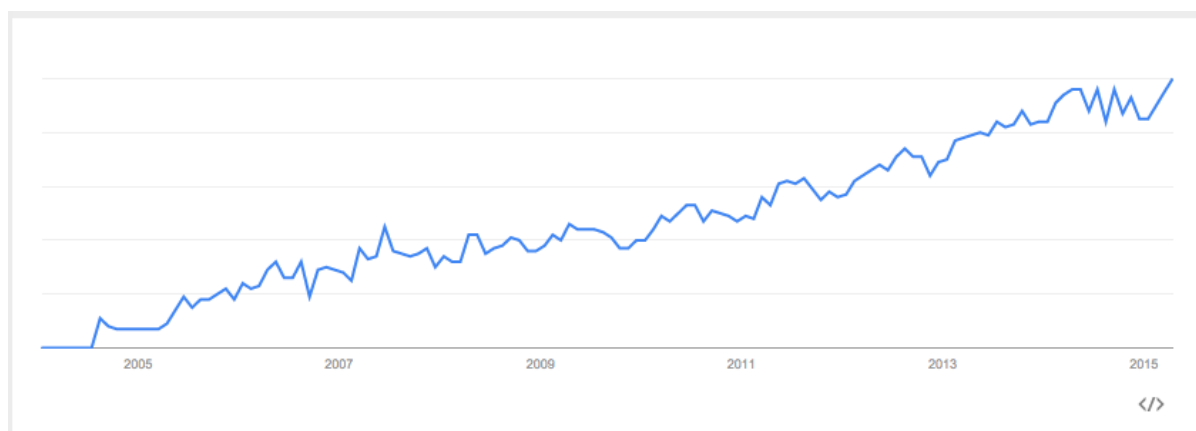
Aparte de lo que acabamos de comentar los nuevos Frameworks MVC JavaScript que se ejecutan en cliente se apoyan en una filosofía similar a las aplicaciones móviles a la hora de asumir responsabilidades de cliente . Con la ventaja además de que están orientados a desarrollar aplicaciones de gran tamaño.



Así pues entre los frameworks móviles y los Frameworks MVC el pastel se tiene que repartir entre mucha mas gente y JSF pierde poco a poco cuota de mercado.

Spring MVC

Spring MVC hace tiempo que compite con JSF pero su enfoque no esta orientado a controles sino que se basa mas en un enfoque MVC puro basado en controladores. La ventaja de este tipo de soluciones es que su integración con las nuevas arquitecturas móviles y JavaScript MVC es más directa. De hecho si revisamos google trends podremos ver que cada día tiene mejor acogida por parte de la comunidad.



JSR 371 MVC (La solución)

Los standards se están haciendo eco de las necesidades de la comunidad y esperemos que la nueva especificación **de JSR 371** que se denomina MVC nos aporte un modelo MVC dentro del standard que permita trabajar de forma integrada con todos los frameworks cliente que existen.

¿Abandonar JSF?

¿Debemos abandonar JSF? .Esta es una pregunta que mucha gente se esta haciendo. No creo que tengamos que ser tan drásticos, JSF sigue siendo una tecnología que podemos usar y que tiene su hueco . Quizás hoy por hoy encaje mas en entornos intranet que internet. Pero lo que cada día esta más claro es que otras arquitecturas están apareciendo en el horizonte y tendremos que abordarlas de una forma u otra. Para que nuestras soluciones mantengan su flexibilidad.

Otros artículos relacionados: [JSF y HTML5](#) , [Spring MVC](#) , [JSF Ajax](#)