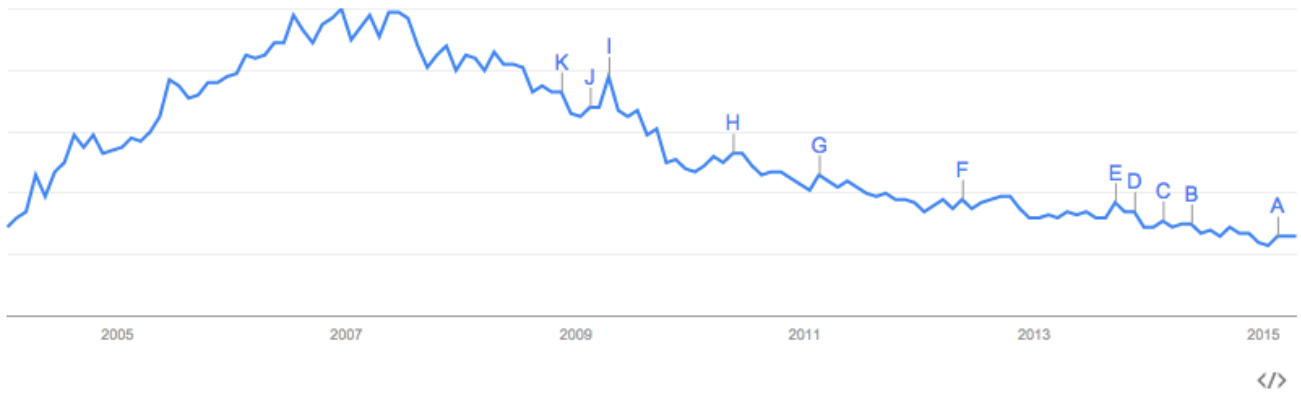
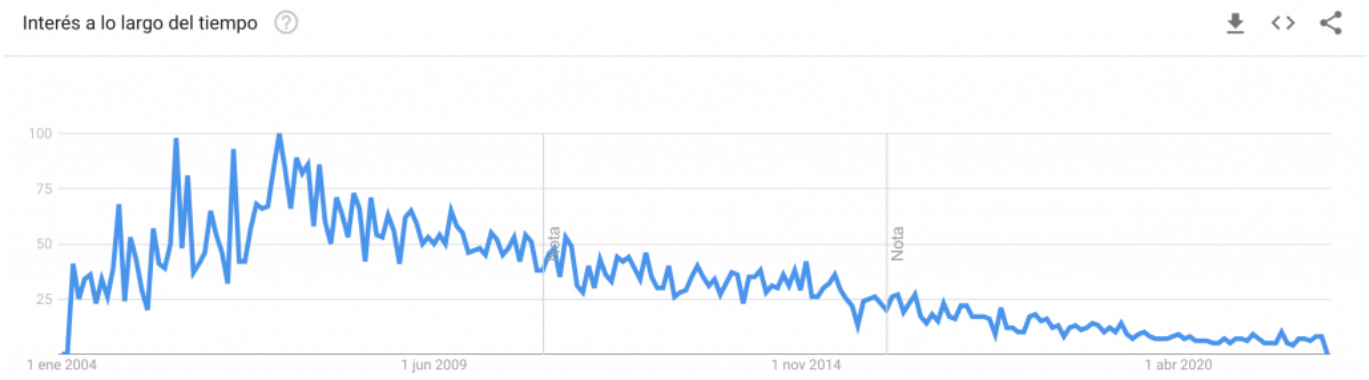
Tabla de Contenidos

- PrimeFaces
- JSF y Servidor
- ¿Entonces JSF ha muerto o esta vivo?
- Java Server Faces 4.0
  - La versión 4
- Otros artículos relacionados:

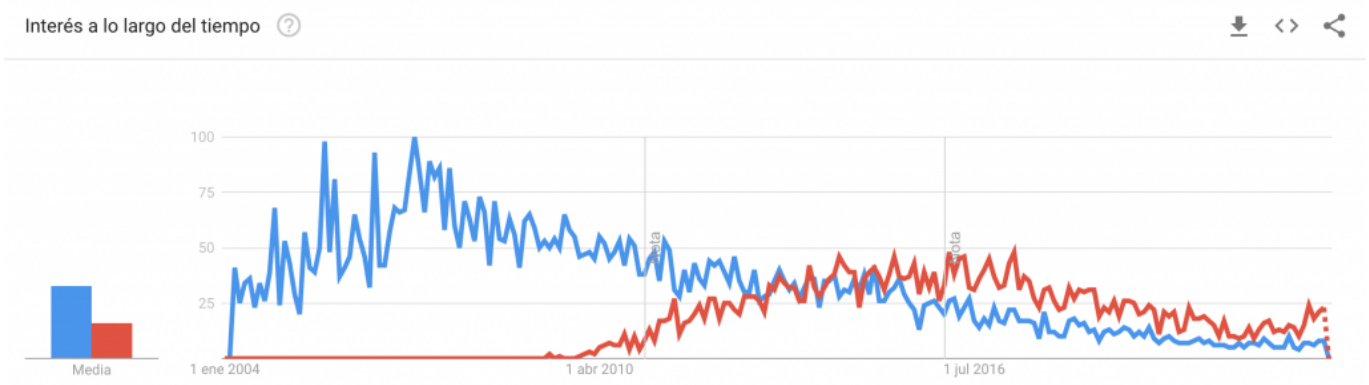
¿JSF y Jakarta EE? .Todos hemos trabajado alguna vez en unos u otros proyectos con JSF (Java Server Faces) y su tecnología orientada a controles. Una pregunta que cada día mas gente se hace es : ¿Cual es el futuro de JSF?. Para muchas personas la respuesta tiende a ser sencilla . JSF es el estandar de Java EE a nivel de capa de presentación por lo tanto tiene futuro. Sin embargo ya hemos tenido en otras ocasiones standards de la plataforma Java que han pasado a mejor vida o han tenido que reinventarse de forma significativa. Vamos a ver un diagrama que en 2015 hablaba sobre cuanto se usaba JSF y cual había sido su evolución usando Google Trends:



En ese momento se veía la tecnología en franco retroceso . Han pasado 7 años desde aquel momento y si volvemos a solicitar la gráfica la situación puede ser mucho más escandalosa.



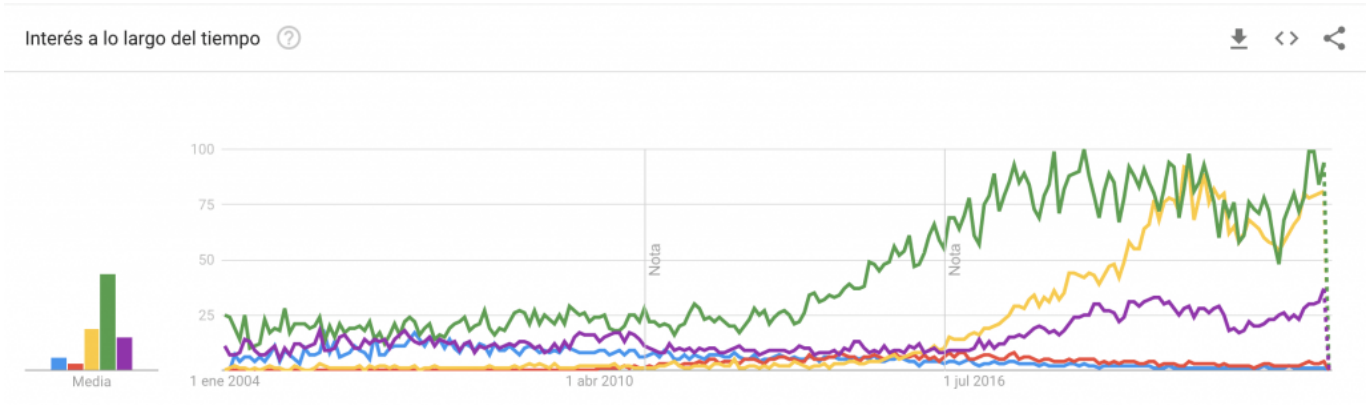
Vista la gráfica nos encontramos un una situación de muerte cerebral de la tecnología o por lo menos eso es lo que parece a tenor de la curva en franco descenso que se muestra. Sin embargo nos guste o nos guste sigue siendo parte de los standares de Java EE . Durante muchos años esta tecnología se ha usado y hay muchas aplicaciones construidas con ella. Si preguntamos a Google Trends por el framework de JSF que más se usa en el mercado hoy en día (PrimeFaces) que engloba no solo la solución de JSF que fue su primer producto sino soluciones adicionales para Vue ,React y Angular basadas todas en un framework de componentes.



PrimeFaces

Si añadimos a la búsqueda PrimeFaces y comparamos la gráfica nos daremos cuenta que hoy por hoy una solución como PrimeFaces que es un producto supera con creces a un

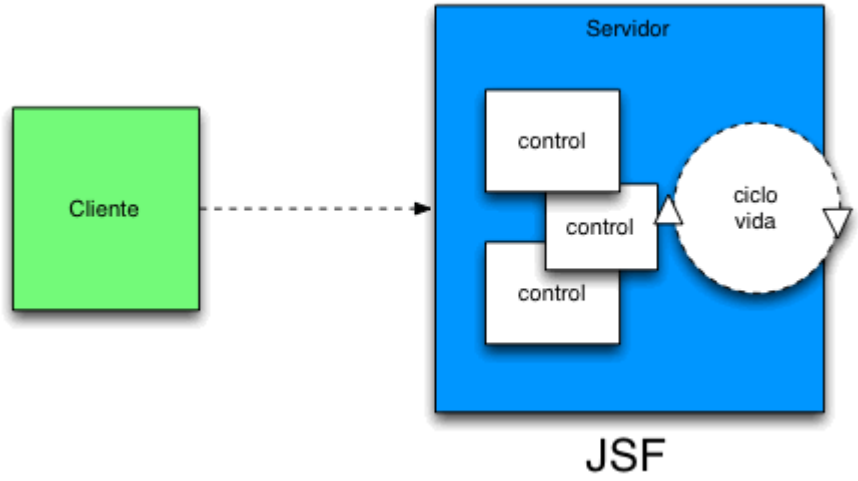
standard como en su momento fue JSF. Es evidente que el interés por JSF ha decrecido de forma absoluta y otras tecnologías han ocupado su lugar a la hora de diseñar soluciones de FrontEnd.



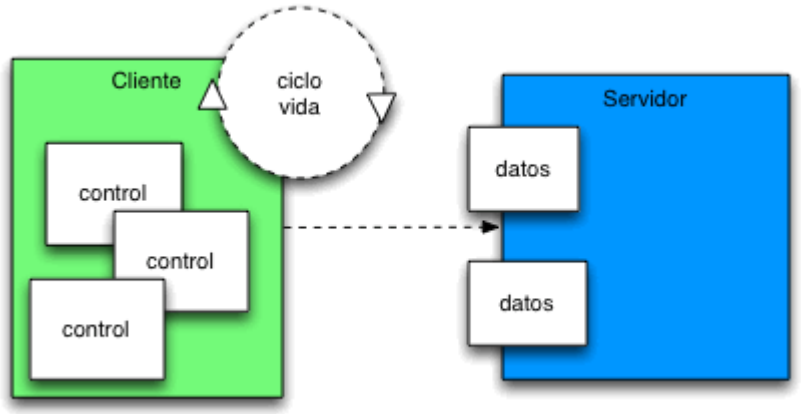
Si miramos la gráfica de Google Trends incluyendo React , Angular y Vue como frameworks de JavaScript . Las cosas se clarifican mucho. JSF e incluso PrimeFaces prácticamente desaparecen de la gráfica. ¿ A que se ha debido esto? .

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 librerías como React , Vue o Angular que se ejecutan en el cliente y que usan el servidor para obtener información de datos e hidratar a los controles de los que disponen.



Apartándonos una gran flexibilidad ya que al publicar la información como datos puros otras aplicaciones se pueden acceder a ellos de una forma transparente. Las arquitecturas REST son mucho más flexibles que estas arquitecturas MVC de Servidor.

¿Entonces JSF ha muerto o esta vivo?

Después de ver las gráficas todos deberíamos entender que JSF ha muerto . Es mucho más interesante desarrollar aplicaciones con Angular o React o Vue a la hora de la plasticidad que ofrecen. Sin embargo sorpresa sorpresa el standard de Jakarta EE que es el nuevo standard de Java anuncia un soporte para JSF 4.0 . Recordemos que hasta hace poco estábamos en la versión 2.3 .

Java Server Faces 4.0

¿Qué sentido tiene esto? . Bueno primero hay que puntualizar que JSF no era una tecnología horrible y que en muchas ocasiones para soluciones complejas en una intranet aportaba una solución razonable muy integrada con los estándares .y la sigue aportando . Segundo y lo más importante JSF es el standard y al ser el standard muchas empresas apostaron por ello por su “longevidad”, porque es algo que en el tiempo perdura y nos permite estar mas tranquilos sobre la evolución tecnológica de nuestra plataforma .

La versión 4

El hecho de que JSF tenga una versión 4.0 dentro de Jakarta . Es una apuesta fuerte de la comunidad Java por la longevidad de la tecnología . Por las miles de aplicaciones internas que muchas empresas tienen desarrolladas sobre Java Server Faces y que necesitan mantener durante un futuro. No todo el mundo se puede permitir inversiones gigantes para actualizar el stack tecnológico por completo. JSF seguirá muy vivo en nuestros servidores durante mucho tiempo ya que los estándares la apoyan y los estandares mandan.

Otros artículos relacionados:

1. JSF y HTML5
2. Spring MVC
3. JSF Ajax