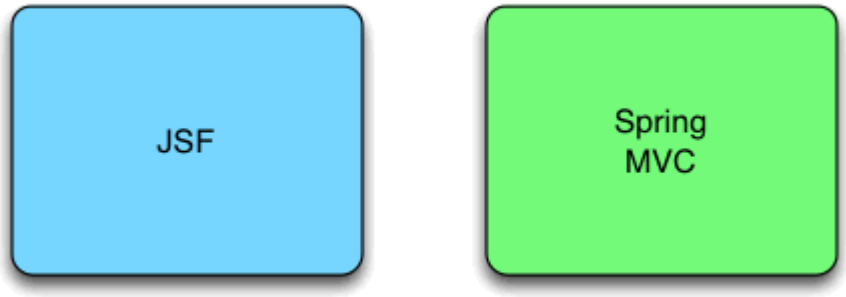
Tabla de Contenidos

- JSF Servidor y Controles
- Spring MVC
- ¿Standards o No?
- Java EE 8 y MVC

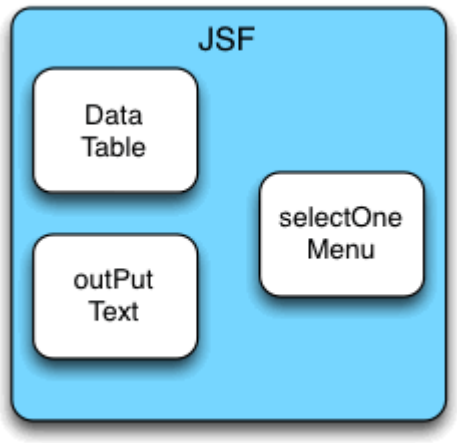
Acabamos de ver como usar Java equals y hashcode

Comparar JSF con Spring MVC es algo que ocurre muy habitualmente ya que son las dos tecnologías de capa de presentación mas utilizadas en los proyectos Java EE. La última vez que leí las estadísticas Spring MVC tenía un 40% y JSF un 22% . Con lo que mas del 60% de las aplicaciones se realizan con estos dos frameworks.



JSF Servidor y Controles

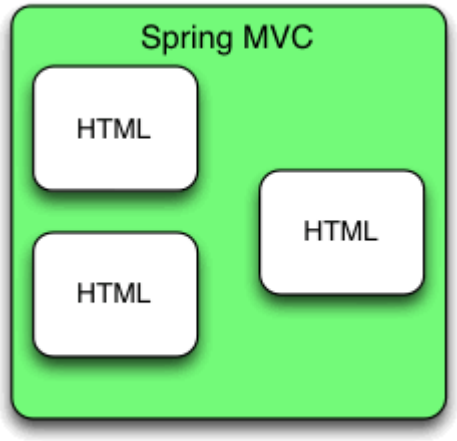
JSF destaca por ser un framework orientado a controles y disponer de varias librerías de controles como RitchFaces, IceFaces y PrimeFaces cada una de las cuales aporta su enfoque particular. Eso sí todas bajo el paraguas del standard y de como este funciona.



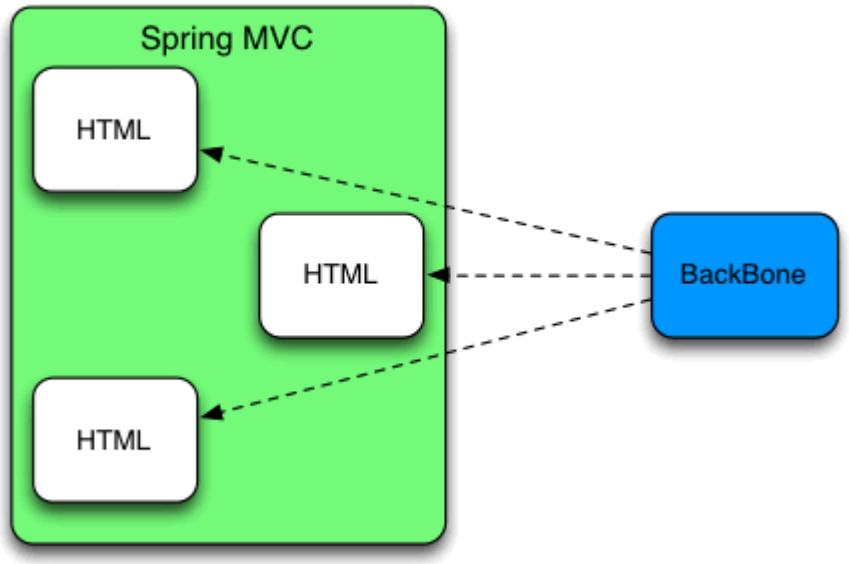
La última evolución del standard ha aportado cosas muy interesantes como data-atributos y Stateless Views que permiten una gran integración con HTML5.

Spring MVC

Por el contrario Spring MVC ha optado por una filosofía diferente en la cual lo más importante es el patrón MVC pero en vez de ser un framework orientado a controles a optado por generar páginas HTML sencillas permitiendo que otros frameworks se puedan acoplar a él de una forma sencilla.

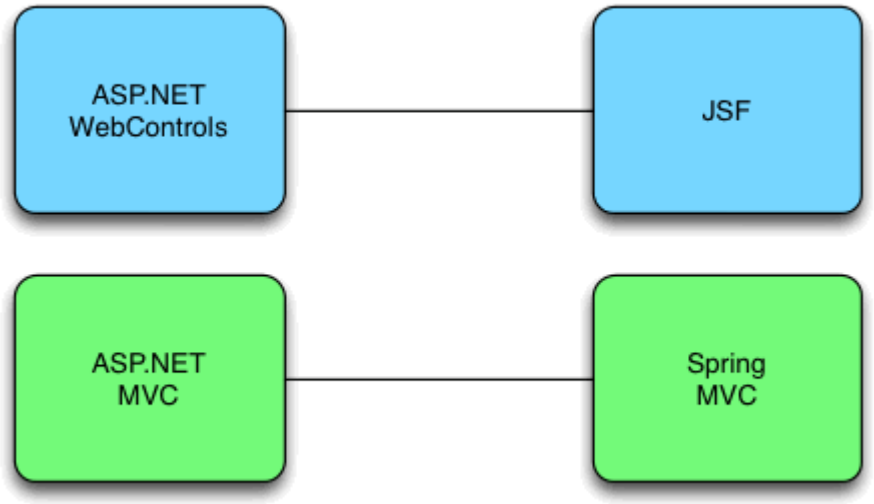


Esto permite que incluso los nuevos frameworks MVC de cliente asuman responsabilidades que antes Spring MVC gestionaba.



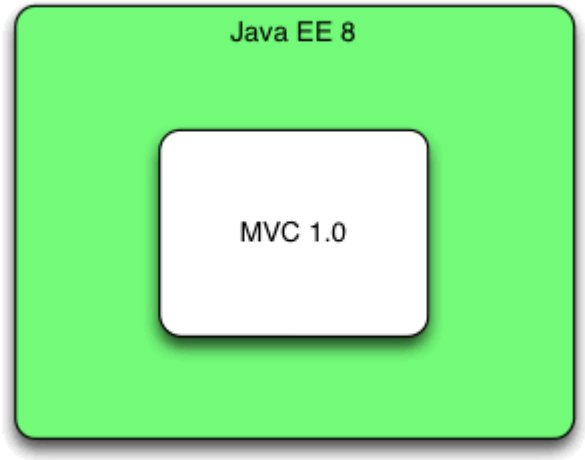
¿Standards o No?

Esta es una de las preguntas más clásicas cuando comparamos ambos frameworks .Spring MVC esta muy bien , pero JSF es parte de los standards. A veces es bueno mirar a la competencia y darse cuenta que en el mundo .NET existen también dos frameworks ASP.NET WebControls y ASP.NET cada uno de los cuales cubre un posible enfoque. Así pues es evidente que hay un hueco para Spring MVC aunque no este dentro de los standards.



Java EE 8 y MVC

¿Es suficiente el pero de Spring MVC para incluir su “filosofía” como parte de los standards?. La respuesta parece que es SÍ y que en Java EE 8 tendremos una respuesta clara con una nueva especificación que por ahora se denomina MVC 1.0 . Yo creo que todos lo vamos ha agradecer.



Quizás los únicos que lo agradezcan un poco menos sea el propio Spring ya que en un número importante de ocasiones se opta por él debido a Spring MVC. Siempre es bueno

saber que la gente que gestiona las especificaciones de Java EE escucha a la comunidad.