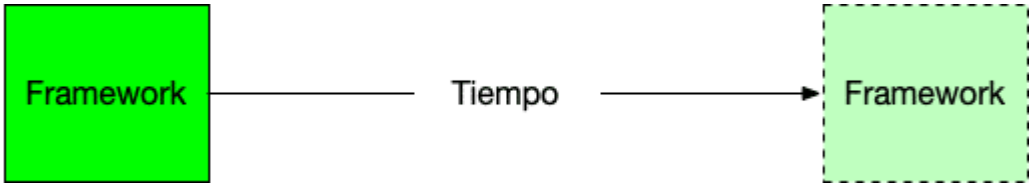
Hoy en día muchos trabajamos con Frameworks Java . Algunos usaran Spring , otros usaran JPA otros EJB y JAX-RS pero todos nos apoyamos en ellos para simplificar la forma de desarrollar nuestra aplicaciones. Sin embargo los frameworks aunque aportan muchas cosas y nos simplifican la vida tambien implican obligaciones . Una de las obligaciones más importantes es que debemos de trabajar de una forma muy determinada con ellos . Esto siempre lo tenemos que tener en mente ya que tiene implicaciones para el ciclo de vida no tanto de una aplicación sino de la arquitectura que estemos diseñando.

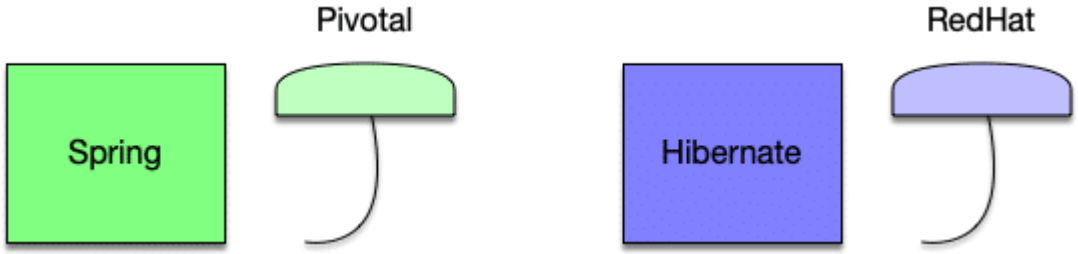
Frameworks Java Arquitectura y tiempo

Nuestras aplicaciones no son eternas y el tiempo implicará la necesidad de actualizar o incluso de reconstruirlas de cero .



Hay a veces que simplemente no interesa actualizar una aplicación porque existen tecnologías mejores para hacerlas o porque el framework que hemos usado para construirla ha muerto.

Siempre es muy difícil saber que frameworks aguantaran el paso del tiempo y cuales no. Sin embargo hay algunas cosas que siempre están claras hay frameworks que nacen y se desarrollan de forma totalmente independiente “Struts” es un ejemplo de ello y hay frameworks que nacen bajo el paraguas de un fabricante “JSF” o “Spring” son dos de ellos .



Siempre existen más posibilidades de que un framework aguante mas tiempo si se

encuentra bajo el paraguas de un fabricante que sino se encuentra bajo el ya que en muchas ocasiones el esfuerzo que hay que hacer para mantener el framework al día es ingente . Un ejemplo de esto lo podemos ver en Spring Framework y como ha evolucionado para adaptarse a competidores como Node con Spring WebFlux.

Frameworks de Cliente y Servidor

Tengamos también en cuenta que existe una distancia importante entre los frameworks de Cliente y los frameworks de Servidor ya que estos últimos cada día se encargan mas de la gestión pura de la información y su evolución será más pausada. Por el otro lado los frameworks de Cliente siempre están ligados a la capa de presentación y a dispositivos que cambian continuamente. Por lo tanto en estos últimos siempre es más difícil que nuestra elección se mantenga y es mas importante utilizar frameworks que estén bajo un paraguas fuerte como es el caso de Angular o el caso de React unos bajo el paraguas de Google y otros bajo el paraguas de FaceBook.

Conclusiones

Pensemos siempre muy detenidamente la elección de nuestros frameworks ya que esto tendrá fuertes implicaciones en el futuro de nuestras arquitecturas . Valoremos sobre manera el uso de frameworks que estén bajo el paraguas de un fabricante y en el caso de elegir uno que no tengamos claro el porque lo hacemos.

Otros artículos relacionados

- Arquitecturas FrameworkLess
- Los Frameworks y su lado oscuro
- SOFEA arquitecturas y flexibilidad
- http://Spring Framework