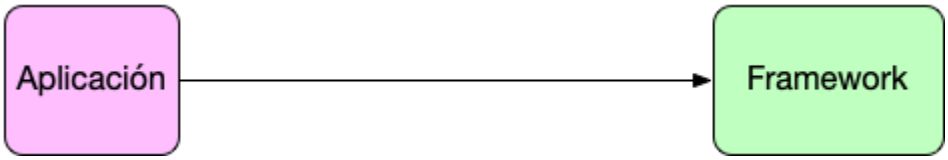
¿Framework vs Frameworkless? Esta pregunta a veces cuesta un poco responderla . Siempre hay dos vertientes en el mundo del desarrollo claramente diferenciadas. Una es la que apuesta por el uso masivo de Frameworks y otra es la que apuesta por el uso puntual de Frameworks . Un Framework no es ni mas ni menos que un conjunto de clases o componentes que nos permiten desarrollar nuestras aplicaciones de una forma mucho mas rápida y productiva a cambio de seguir una serie de reglas o convenciones.

¿Porque no usar Frameworks?

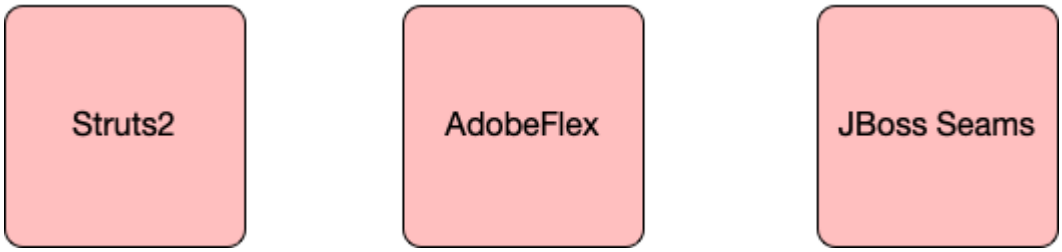
Con lo cual la pregunta más lógica es ¿Por qué no usar frameworks? . La realidad es que todo el mundo los usa . Lamentablemente el uso de frameworks implica que nos quedamos acoplados a él.



Es decir la ganancia de productividad no es “gratuita” sino que asumimos un sacrificio que quedamos fuertemente ligados al uso de este framework.

Frameworks y Acoplamiento

Esto en un principio no parece problemático . Pero lo puede llegar a ser sí el framework en un momento concreto desaparece . En ese momento tendremos un gran problema. No hay que echarle mucha imaginación para ver qué paso con Struts 2 o qué pasó con JBoss Seams o con Adobe Flex. Son frameworks que en su momento tuvieron un fuerte auge y que luego pasaron a la historia.



Aquellas empresas que desarrollaron mucho sobre dichas plataformas hoy tienen un problema o un problemon.

¿Frameworks vs FrameworkLess?

Es realmente difícil tomar decisiones sobre este tema . Si tuviera que dar una opinión es que normalmente prefiero el uso de frameworks . Eso sí, siempre teniendo mucho cuidado sobre cuales elegir e intentando que estén bajo el paraguas de compañías fuerte Spring Framework (pivotal) o Hibernate (JBoss) , Angular (Google). No hay que tener ninguna duda que la selección de frameworks de cliente es más peligrosa .Primero porque existen más y segundo porque el universo de JavaScript avanza a pasos agigantados y solo deja muertos en el camino. Lo que puede ser válido hoy puede no ser valido mañana. Hay equipos que optan por arquitecturas fFameworkless apoyandose en React o Vue mezclados con otras librerías tipo Axios .

Frameworkless y sus ventajas

Las ventajas de las arquitecturas Frameworkless están relacionadas con la longevidad de las tecnologías que usan . Una librería tiene un ciclo de vida mucho más largo que un Framework y esto muchas veces importa. De ahí que cada día nos encontremos con soluciones basadas en React o Vue que optan por este enfoque que ademas es más fácil de integrar en arquitecturas clásicas o legacy.

Framework vs Frameworkless y su cultura

Los frameworks no son tan longevos como las librerías normalmente . Eso sí hay excepciones ,Spring o Hibernate llevan 20 años en el sector. No es el mismo caso de Angular . Una de las ventajas que para mi siempre tienen los Frameworks es que generan una “cultura” de conocimiento y construcción de aplicaciones “comunes” a toda la organización .

Rotaciones y Cultura

Eso hoy en día es importante este concepto de cultura con el nivel de rotación de personal tan alto existente . Que todo el mundo trabaja de la misma forma y que todo el mundo trabaje con un Framework Standard hace mucho más sencillo encontrar nuevos desarrolladores . Un enfoque frameworkless corre el riesgo de ser muy a medida y tener mas complicaciones a la hora de localizar las personas adecuadas.

Conclusiones (Framework vs Frameworkless)

Hoy por hoy el apostar por frameworks Standard puede ayudarnos a gestionar de forma más uniforme la integración de personas en los equipos.

Otros artículos relacionados

- Java Collections Framework y su estructura
- ¿Cómo elegir un buen Framework?
- Framework vs Libreria dos conceptos importantes