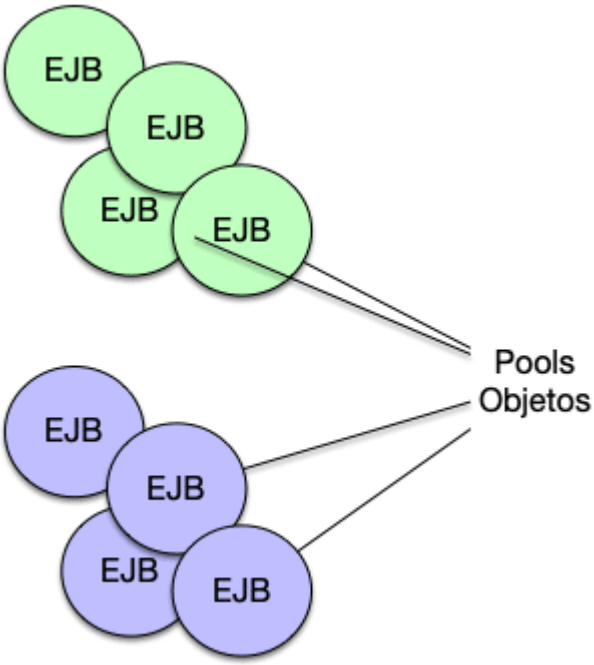
Tabla de Contenidos

- Java EE y EJBs
- JSR una pieza clave
  - ¿Spring Boot es más moderno?

Java EE y el futuro es una pregunta cada día más importante para mucha gente . Hoy por hoy en el ecosistema de Java siempre nos hemos encontrado con dos enfoques muy diferentes . Gente que apostó por Spring Framework y gente que apostó por los standards de Java EE tipo EJB ,CDI , JAX-RS etc . Ultimamente parece que es Spring el que gana la batalla , solo hay que ver las solicitudes de Spring Boot , Spring MicroServices etc en el mundo de Java EE a nivel de ofertas de trabajo. ¿En qué situación se encuentra Java EE?. Como ya explique hace poco en el blog ha cambiado su nombre a Jakarta EE.

Java EE y EJBs

Hace un par de días una persona me pregunto porque los EJBs no encajan en las nuevas arquitecturas de Microservicios. Bueno la realidad no es que no encajen cómo tal ya que los EJBs es una tecnología orientada a generar un pool de objetos que sean capaces de manejar un conjunto de peticiones elevadas.



En muchas casuísticas los MicroServicios no necesitan de este tipo de componente. Aunque en otras ocasiones puede venir bien . Desde mi punto de vista la clave es , cuantas librerías necesita un MicroServicio para funcionar correctamente. La respuesta suele ser que cuantas menos use mejor ya que la imagen Docker que se generará será de menor tamaño y la memoria consumida también . Por ejemplo podemos ver que Payara aporta un soporte para EJB Lite en su Arquitecturas de MicroServicios mientras que da la sensación que Quarkus realiza un apoyo más fuerte a CDI sin hacer uso de EJBS generando imágenes más ligeras.

JSR una pieza clave

La realidad es que todo esto de lo que estamos hablando no importa demasiado hoy en día aunque nos lo parezca. ¿Se van a usar EJBs , no se van a usar? . Lo que hoy importa realmente es que las arquitecturas clásicas están en un proceso de rediseño importante, todos nos estamos dando cuanta de ello . ¿Cómo encaja esto en nuestra plataforma ,en lo que tenemos desplegado? . La realidad es que terminará encajando porque el uso de los standards (JSR) ya sea EJB Lite , CDI , JAX-RS seguirá. Otra cosa es como los nuevos

servidores de aplicaciones los usen o los incluyan pero lo importante es que los standards nos dan seguridad en el tiempo y nos permiten tener aplicaciones que sigan funcionando durante décadas , si décadas . Esto es lo que mucha gente necesita a día de hoy . Los EJBs son de 2001 y hoy siguen usándose , los Servlets igual así que la apuesta fue correcta.

¿Spring Boot es más moderno?

Esta es otra pregunta habitual , a mucha gente le parece que Spring Boot es más moderno que va por delante etc . La realidad es que la gente de Spring puede siempre hacer lo que quiera ya que están en una parte dentro de los standards y en otra parte muy fuera. Siempre han seguido su propio camino . Los standards siempre son más lentos en evolución pero siempre acaban llegando hay que tener algo de paciencia.

Otros artículos relacionados

- ¿Que es Spring Boot?
- ¿Que es Spring WebFlux?
- MicroServices con Payara