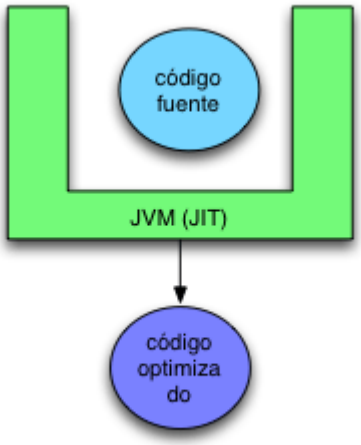
Todos hemos oído hablar de la lentitud de Java, siempre ha sido evidente que no puede ejecutarse de la misma forma que se ejecuta una aplicación nativa y su rendimiento es algo menor. Con los años muchas cosas han mejorado y las máquinas JIT (Just In Time) han conseguido mejorar el rendimiento del lenguaje. Las máquinas JIT son las encargadas de generar un código nativo muy optimizado , el siguiente artículo nos lo explica a detalle.



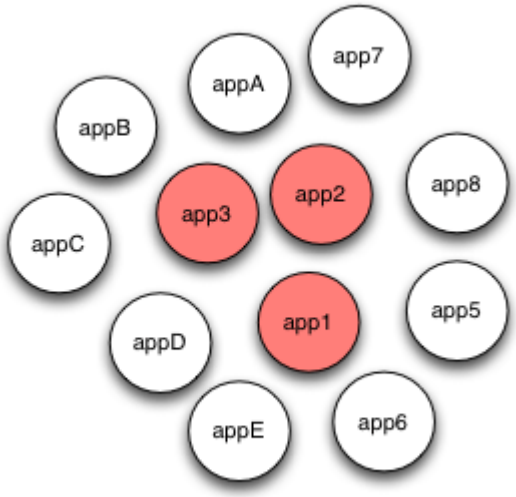
Es cierto que Java se ha apoyado también en la mejora de rendimiento de los procesadores, solo hay que ver la plataforma Android para darnos cuenta de lo que era y de lo que es hoy. Sin embargo la lentitud es una de las claves del éxito de Java pero en otro sentido.

La lentitud de Java y el mundo enterprise

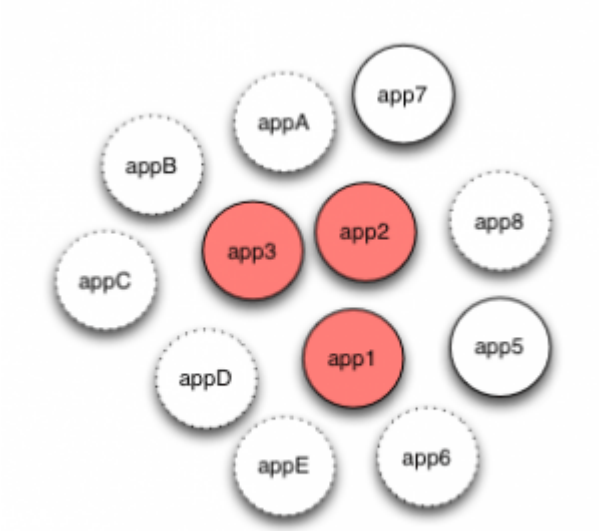
Lo que más destacaría de Java es su enfoque (enterprise) hacia la empresa algo que es clave ya que las empresas necesitan una evolución más pausada de los frameworks y de las tecnologías asociadas a ella. Si miramos el mundo de JavaScript Angular va ya por la versión 19 en los pocos años que han pasado. Sin embargo Spring Framework va por la versión 6 . Nos puede parecer una curiosidad pero React va por la versión 18 y JPA va por la versión 3. En Java muchos frameworks y standards se desarrollan con la intención de que duren muchos años y se siga dando mantenimiento de ello y eso es clave para dotar a las empresas grandes de seguridad en la construcción de aplicaciones.

Las empresas y sus corazones

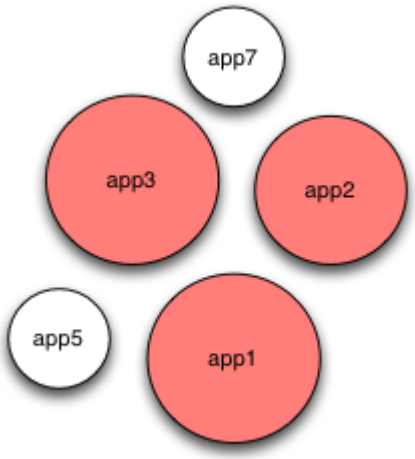
Es cierto que todas las empresas tienen en explotación decenas o cientos de aplicaciones, sin embargo hay un conjunto de aplicaciones (grandes o pequeñas) que son el núcleo de su negocio y por decirlo de alguna forma son su “corazón” sin ellas la empresa tendrá grandes dificultades para la gestión y supervivencia diaria.



Con el paso del tiempo muchas aplicaciones de la empresa caen en desuso, pero las aplicaciones “enterprise” sobreviven.



Estas aplicaciones pueden tener 10-30 años de vida , con el tiempo habrán evolucionado y su tamaño aumenta.



Estas aplicaciones necesitan evolucionar , pero necesitan poderlo hacer de una forma “pausada” sin grandes saltos. Es aquí en donde Java y Java EE y Spring destacan , ¿Podemos usar el mismo Servlet que hace 10 años en una aplicación de hoy?. La respuesta es SI. Es cierto que podemos usar otras tecnologías más avanzadas Pero si que es cierto que los Servlets y los JSP siguen estando disponibles . ¿Podemos usar JSF o EJBS a día de hoy?. La respuesta es SI. De hecho todavía vamos por EJB 4.0 que indica que la evolución no ha sido revolucionaria sino mas bien lenta. Esto es clave para el mundo “enterprise” ya que genera

una “cultura” entre los desarrolladores que se extiende en el tiempo y permite que todos nos entendamos en diferentes lineas temporales. Podemos hoy en día usar Swing? , pues si que podemos, es verdad que JavaFX es una alternativa pero Swing sigue estando disponible. Eso es clave para la apuesta tan fuerte que muchas empresas han hecho sobre la tecnología Java en las últimas décadas y les ha permitido sobrevivir

La necesidad de seguir trabajando

Las empresas necesitan que sus aplicaciones enterprise/core funcionen correctamente y puedan evolucionar . Es en este entorno en el que la inversiones no se pueden congelar ya que si las aplicaciones core no funcionan correctamente la empresa no podrá operar. Es en este entorno en donde la forma pausada de evolucionar de Java se convierte en una ventaja competitiva sobre el resto . ¿ Era Angular 2.0 compatible con Angular 1.3? .JavaScript a veces esta en las antipodas en cuando a mantenimiento de Frameworks y Librerias.

Otros artículos relacionados :

- Arquitecturas y Modularidad
- Arquitecturas MVC y Rest