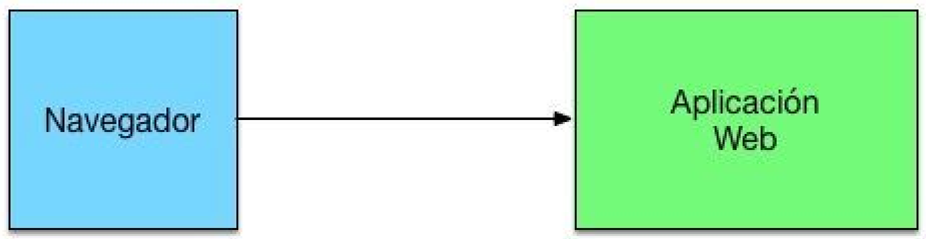
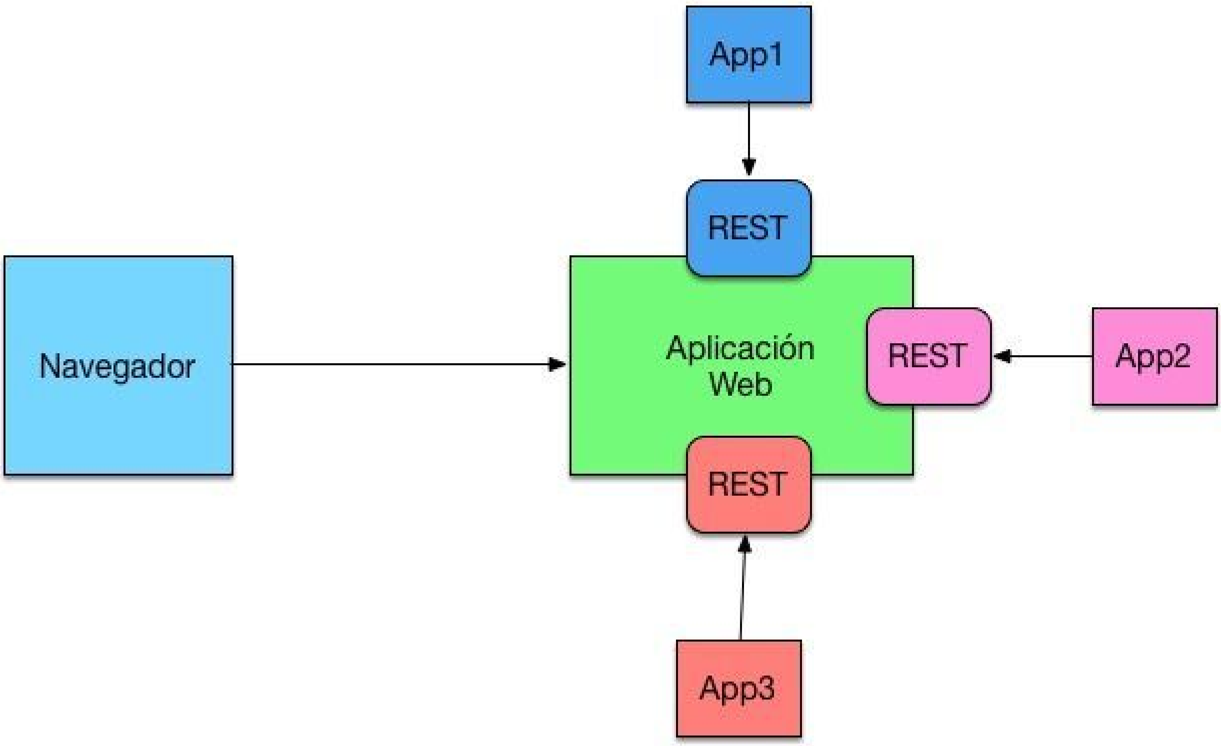
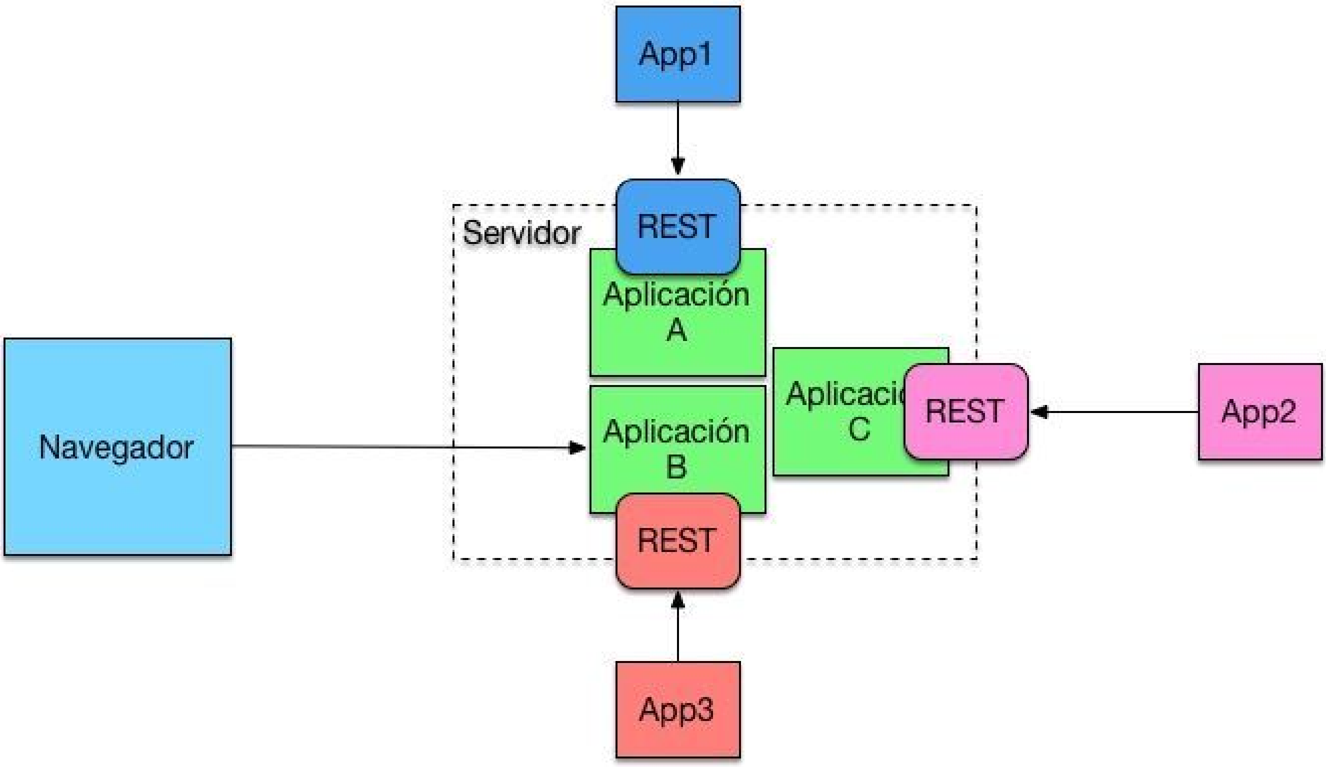
¿Como funcionan los Reactive MicroServices? . El mundo de los microServicios es difícil de entender. Todo el mundo quiere empezar con ello . Pero evolucionar hacia una arquitectura de este estilo no es una tarea fácil. Para entender un poco mejor el concepto de Reactive MicroServices, debemos revisar las arquitecturas que tenemos en estos momentos . Lo más habitual es que tengamos construidas aplicaciones web monolíticas. Un navegador accede a la aplicación.



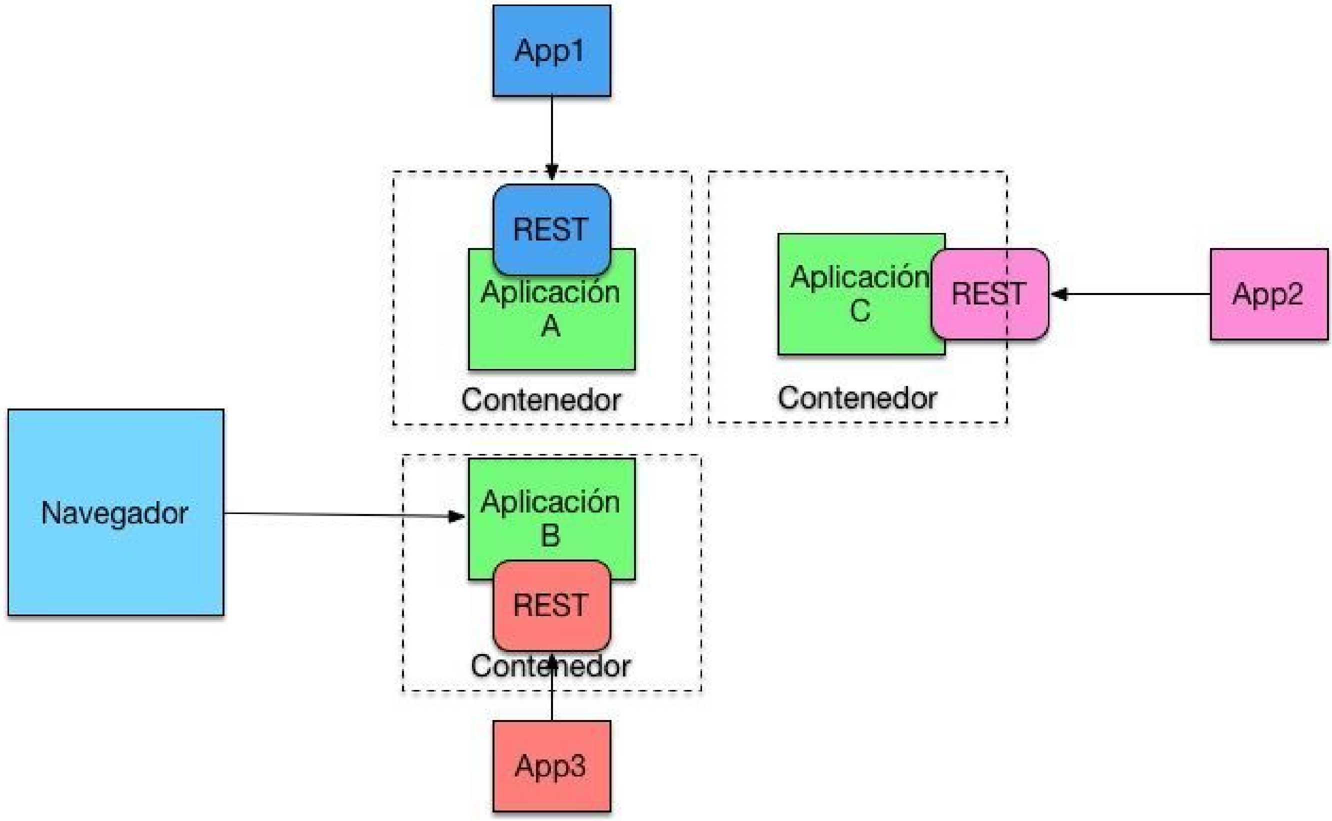
La realidad es que esto siempre incluye algo más, ya que nuestra aplicación tiene que dar acceso a varios clientes móviles o a otras aplicaciones. Así pues habremos publicado de una forma u otra X servicios REST.



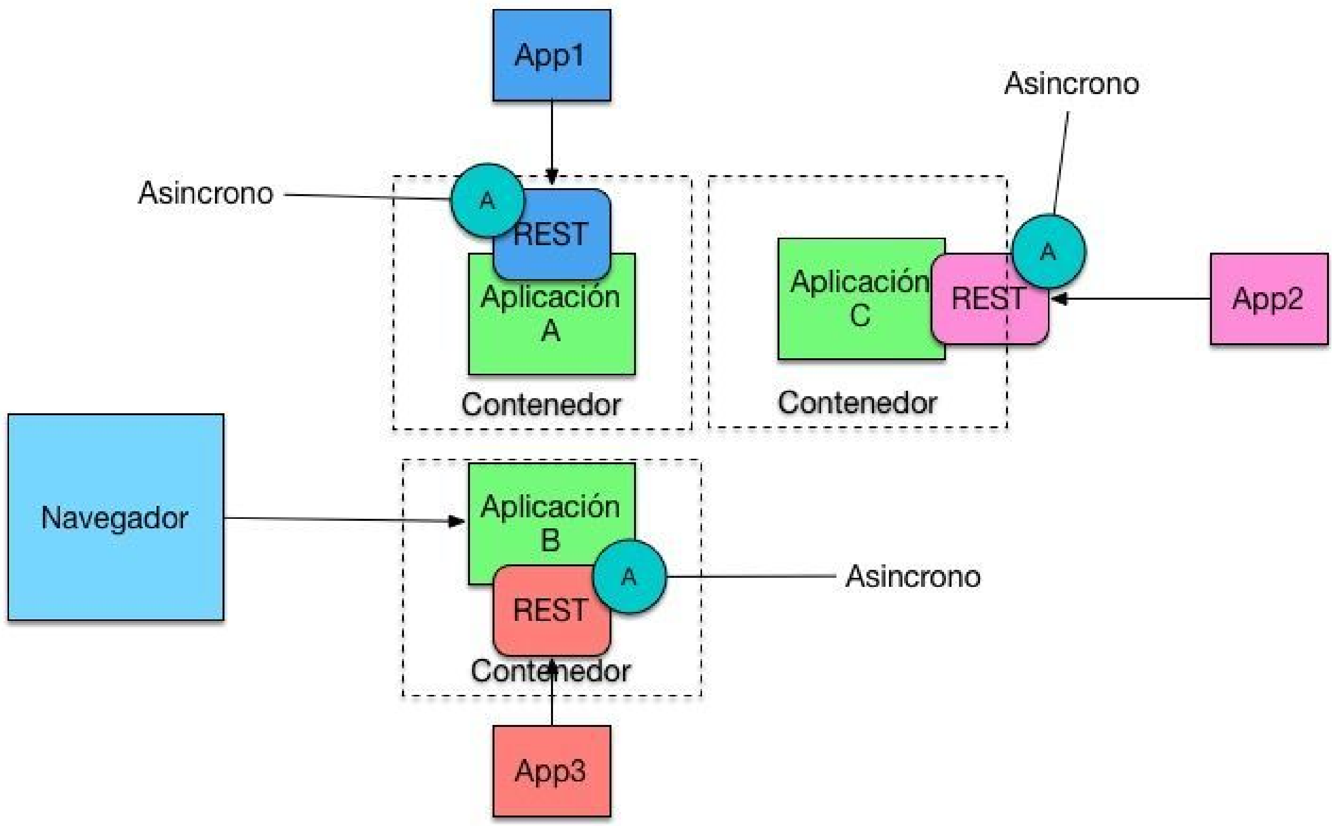
Es aquí donde empiezan los problemas . Cuando tenemos unos pocos servicios REST no pasa nada . Pero cuando tenemos muchos independientes entre si , nos encontraremos que para actualizar la App1 debemos redesplegar nuestra aplicación. No parece algo complejo pero a más aplicaciones conectadas vía REST mayores serán los despliegues y mas constantes. Esto es un problema. El siguiente paso natural es modularizar de alguna forma nuestra aplicación y convertirla en un conjunto de aplicaciones pequeñas independientes.



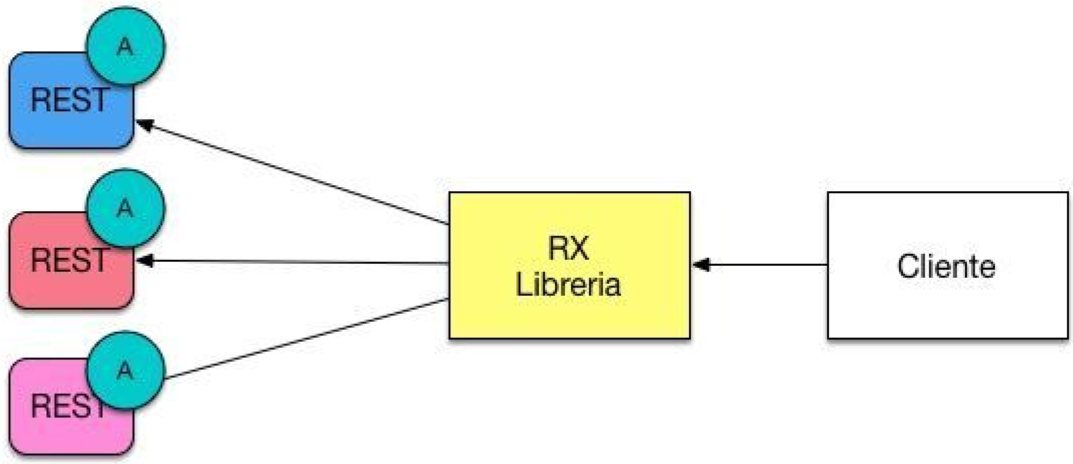
El problema es que estas aplicaciones independientes , no son tan independientes como pensamos. Todas se ejecutan en el mismo servidor .Así que si una aplicación se come la memoria o la CPU de la máquina el resto de las aplicaciones sufrirán . El siguiente paso es generar contenedores tipo Docker y aislar las aplicaciones haciéndolas completamente independientes.



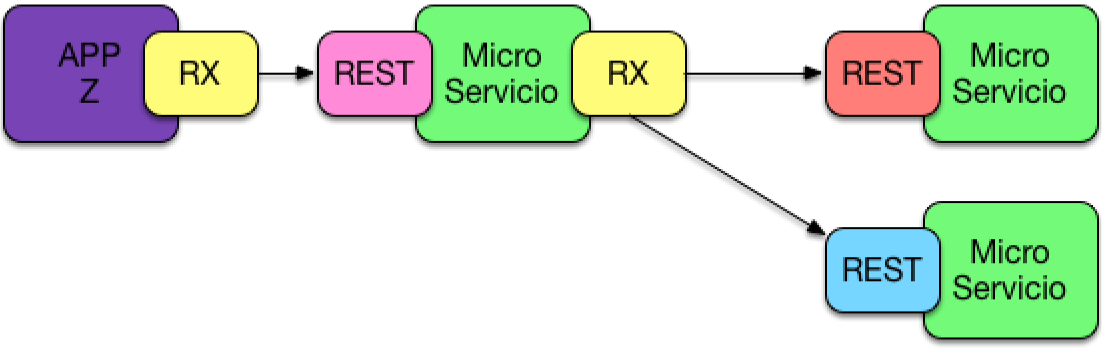
Si un contenedor falla no afecta a los demás. Hemos avanzado otro paso . El siguiente problema empieza cuando tengamos que construir aplicaciones que necesitan consumir servicios REST de forma masiva. Algo que con la arquitectura que estamos diseñando será una consecuencia directa. Todas las peticiones que realizamos son asíncronas.



Así pues necesitaremos utilizar programación Reactiva para realizar una comunicación asíncrona óptima y correcta entre los distintos APIs REST de los diferentes Microservicios que creamos.



No solo eso sino que es posible que nuestros propios microservicios sean clientes de otras aplicaciones a las cual accedemos vía REST.



Las extensiones RX nos las vamos a encontrar en muchas plataformas cuando desarrollemos estas nuevas Arquitecturas. Eso sí la evolución hacia arquitecturas de microservicios es un reto y habrá que abordarlo con mucha calma .Todavía quedan muchos temas por clarificar.

Otros artículos relacionados : RxJava , MicroServicios , Docker