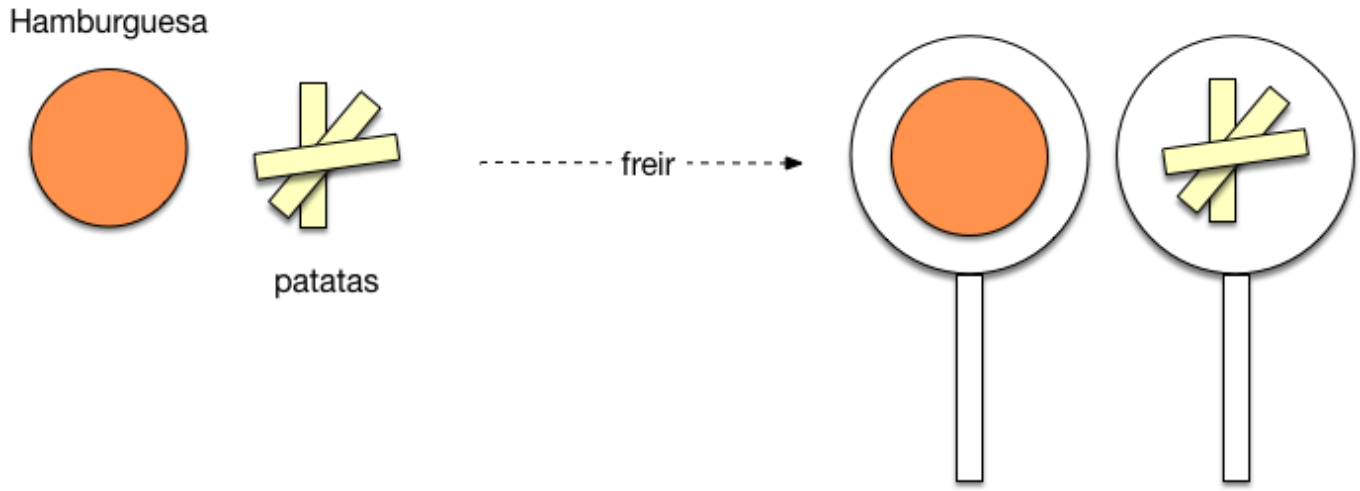
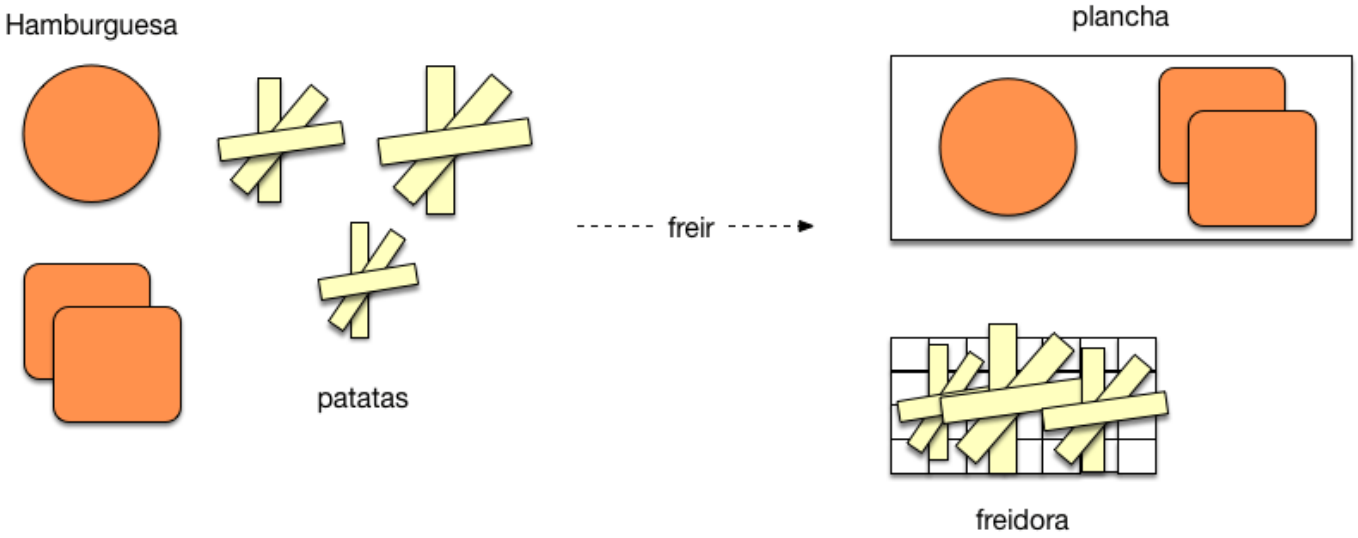
La diferencia entre framework vs libreria es a veces difícil de entender sobre todo cuando uno esta empezando. ¿Qué diferencia existe entre un framework vs librería? .Vamos a intentarlo explicar con un ejemplo ajeno a la programación .Supongamos que nosotros queremos cocinar una hamburguesa con patatas. En principio se trata de una operación muy muy sencilla . Necesitamos la hamburguesa , las patatas y si lo queremos hacer rápido un par de sartenes para freir cada una de ellas.



Un programa

Hasta aquí todo es correcto , podemos cocinar nuestra hamburguesa con patatas. Aquí es donde podemos realizar un símil con el concepto de construir un programa, simplemente hemos usado los métodos standard para construirlo y punto. ¿Se puede hacer mejor?… uhmm quizás sí la verdad es que freir la hamburguesa quizás no es lo más sano. La podemos hacer a la plancha y nos quedará mejor y más sana. De igual forma quizás sea más limpio freir las patatas en una freidora con aceite nuevo ya que podemos freír unas cuantas más (estas no sobran nunca).

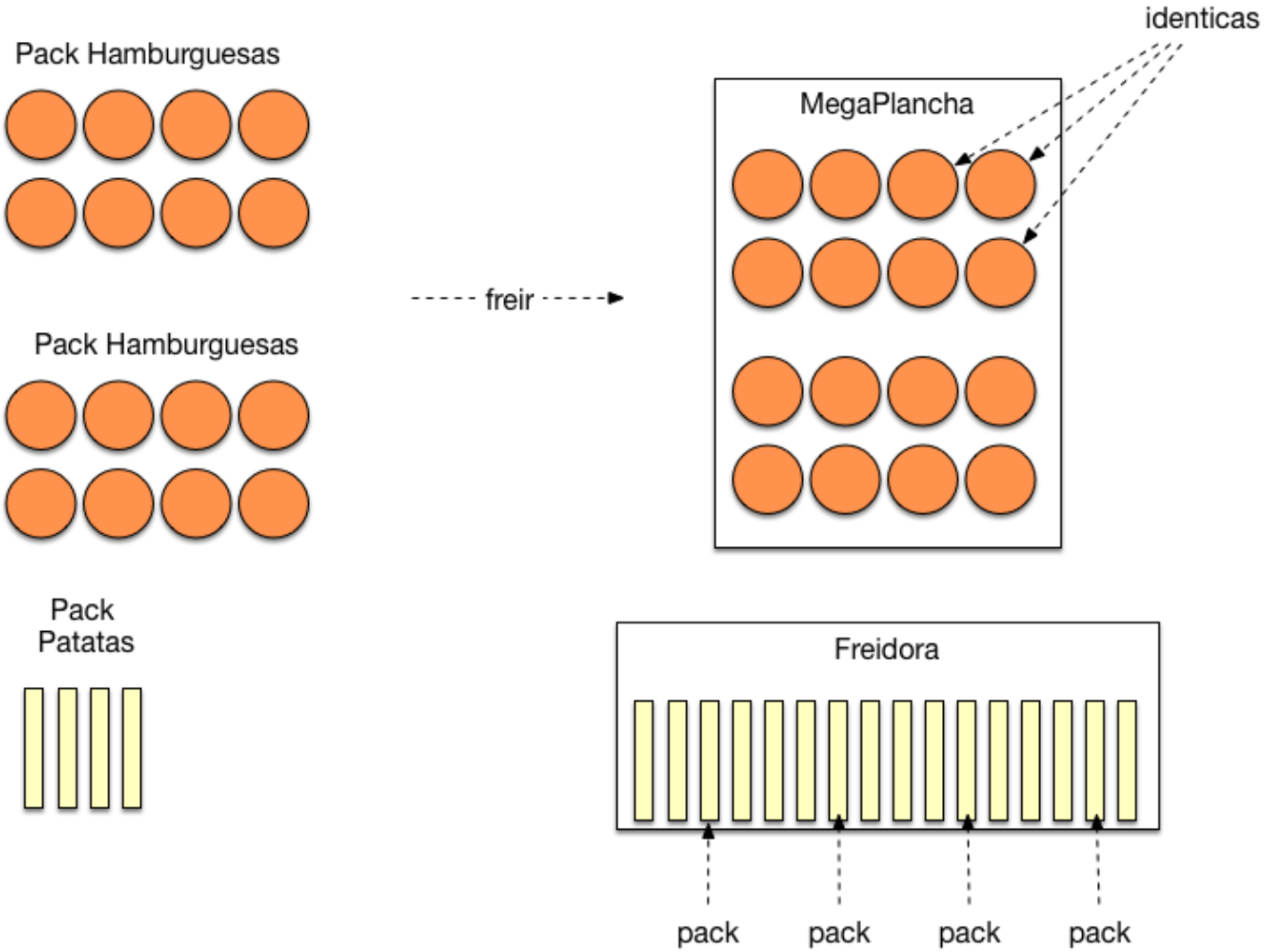


Framework vs Libreria (Libreria)

La plancha ademas admite bien hamburguesas cuadradas que están de moda. Ok todo es correcto , podemos usar la plancha y la freidora y mejorar la forma en la que cocinamos la hamburguesa. Este es el concepto de libreria , un conjunto de código que nos facilita la construcción de nuestro programa. Un ejemplo sería jQuery , podemos hacer las cosas algo mejor y con menos esfuerzo.

Framework vs Libreria (Frameworks)

Esta es la forma en la cual la gente pro hace las hamburguesas con patatas en casa , e incluso puede ser la forma en la que un pequeño restaurante las hace. ¿ Ahora bien cómo las hace McDonalds? . Este tema es muy diferente , McDonalds es posible que use hamburguesas congeladas, todas del mismo tamaño y que las cocine por bloques. No solo eso sino que también usara las mismas patatas congeladas todas idénticas o muy similares que freirá en su freidora por packs.

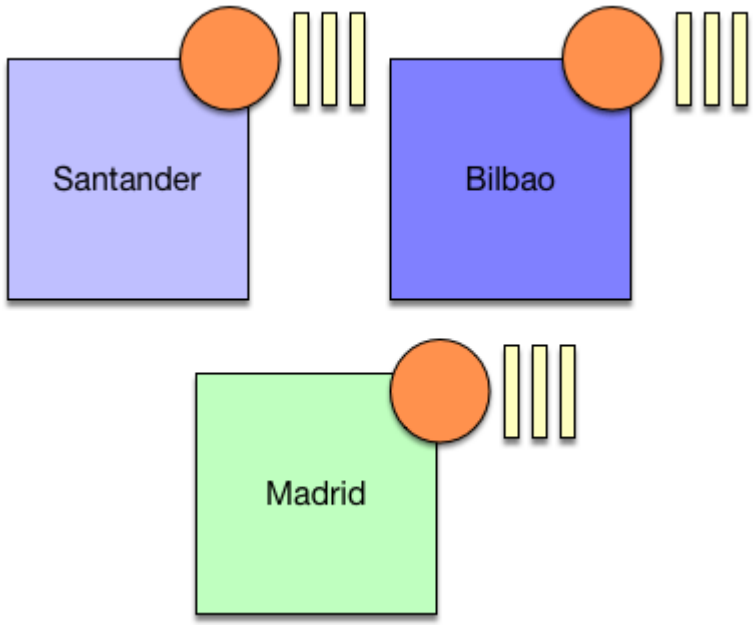


Esto parece en un primer momento que se parece mucho a la solución anterior pero solo que para grandes volúmenes. La realidad es que se parecen en poco . ¿Qué fue de nuestras hamburguesas cuadradas? . Muy sencillo McDonalds las hace solo redondas y punto. ¿Oye y de esas patatas unas más grandes y otras más pequeñas ?. Tampoco existen McDonals nos traen las patatas congeladas y todas son iguales e identicas. ¿Oye que yo quería la hamburguesa muy hecha? . Nada imposible la mega plancha las hace todas a la vez. ¿En qué hemos ganado entonces? .

Frameworks y Restricciones

Pues así visto lo visto parece que hemos perdido ya que estamos más limitados. La realidad

es un poco diferentes , hemos perdido en flexibilidad, esa es la clave, el proceso no es tan flexible. Sin embargo hemos ganado en productividad ya que en poco tiempo cocinamos mucho . Para lograrlo se asumen varias limitaciones las hamburguesas solo son de un tipo, igual pasa con las patatas , la freidora y la megaplancha también son muy específicadas. ¿Qué más hemos logrado? Hemos logrado generar homogeneidad ya que todas las hamburguesas en todos los McDonals nos saben igual .



Da lo mismo donde estemos ubicados. Acabamos de generar una “cultura” . Eso es lo que hace un framework , aportar productividad , define limitaciones y genera homogeneidad a la hora de trabajar dentro de una organización .

10 Frameworks de Java Más Usado

A continuación te comparto una lista con los frameworks más populares en el ecosistema Java, cada uno con su descripción y enlace a la página oficial. ¡Echa un vistazo y descubre cuál es el más adecuado para tu próximo proyecto!

1. Spring Framework

Es el framework más completo para construir aplicaciones en Java, ideal para proyectos grandes y complejos. Te ayuda a gestionar todo lo relacionado con inyección de

dependencias y arquitectura empresarial.

2. Spring Boot
Si quieres desarrollar rápido y sin complicaciones, Spring Boot es perfecto. Hace que crear aplicaciones listas para producción sea súper fácil con una configuración mínima.

3. Hibernate
Este framework se encarga de que trabajar con bases de datos sea mucho más sencillo. Hibernate convierte objetos Java en tablas de bases de datos de manera automática, lo que te ahorra mucho código repetitivo.

4. JSF (JavaServer Faces)
Ideal para crear interfaces de usuario en aplicaciones web usando componentes predefinidos. Simplifica mucho la vida cuando necesitas manejar formularios y vistas en aplicaciones empresariales.

5. Vaadin
Con Vaadin puedes crear aplicaciones web modernas sin tocar JavaScript. Todo lo haces en Java, lo que facilita mucho el desarrollo si prefieres mantener todo en un solo lenguaje.

6. Struts
Es un framework clásico para construir aplicaciones web en Java usando el patrón MVC (Modelo-Vista-Controlador). Ayuda a mantener bien organizada la lógica de tu aplicación.

7. GWT (Google Web Toolkit)
Con GWT puedes escribir todo tu código en Java, y luego el framework lo convierte en JavaScript para que funcione en el navegador. Perfecto si quieres evitar lidiar directamente con JavaScript.

8. Grails
Basado en Groovy, Grails hace que el desarrollo web sea más ágil, especialmente si ya estás familiarizado con Spring y Hibernate. Es muy productivo para proyectos pequeños y

medianos.

9. Micronaut
Micronaut es perfecto para crear microservicios y aplicaciones ligeras. Se arranca rápido y consume menos memoria, lo que es ideal para la nube.

10. Quarkus
Quarkus está diseñado para aplicaciones en la nube y Kubernetes, y su gran ventaja es que arranca súper rápido y usa pocos recursos, ideal para entornos modernos de microservicios.

Con estos frameworks tienes un abanico de posibilidades para desarrollar aplicaciones web, microservicios y más en Java. Cada uno está diseñado para distintos casos de uso, ¡así que elige el que mejor se adapte a tus necesidades!

Otros artículos relacionados

1. Angular 5 Hello World y su funcionamiento
2. React vs Angular 2 , frameworks vs librerias
3. El patrón MVC , arquitectura cliente vs servidor
4. Framework Wikipedia