



GUSTAVO CORONEL
DESARROLLA SOFTWARE



ENTERPRISE JAVA DEVELOPER DESARROLLO WEB CON SPRING BOOT

OVERVIEW DE LA PROGRAMACIÓN REACTIVA

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

gcoronelc@gmail.com

LOGRO ESPERADO

Al finalizar esta lección el participante tendrá una idea clara de que es la programación reactiva, sus beneficios y donde podrá utilizarla.



LO QUE CONOCEMOS

Actividad desarrollada con
mentimeter.

Accede a la siguiente URL:

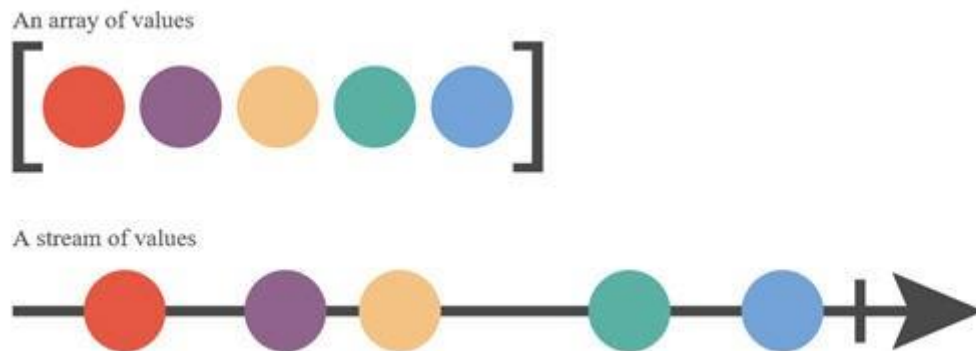
www.menti.com

Luego, ingresa el código que
muestra en pantalla.



NECESIDAD

La programación reactiva, o Reactive Programming, es un paradigma enfocado en el **trabajo con flujos de datos finitos o infinitos de manera asíncrona**, permitiendo que estos datos se propaguen generando cambios en la aplicación, es decir, “reaccionan” a los datos ejecutando una serie de eventos.

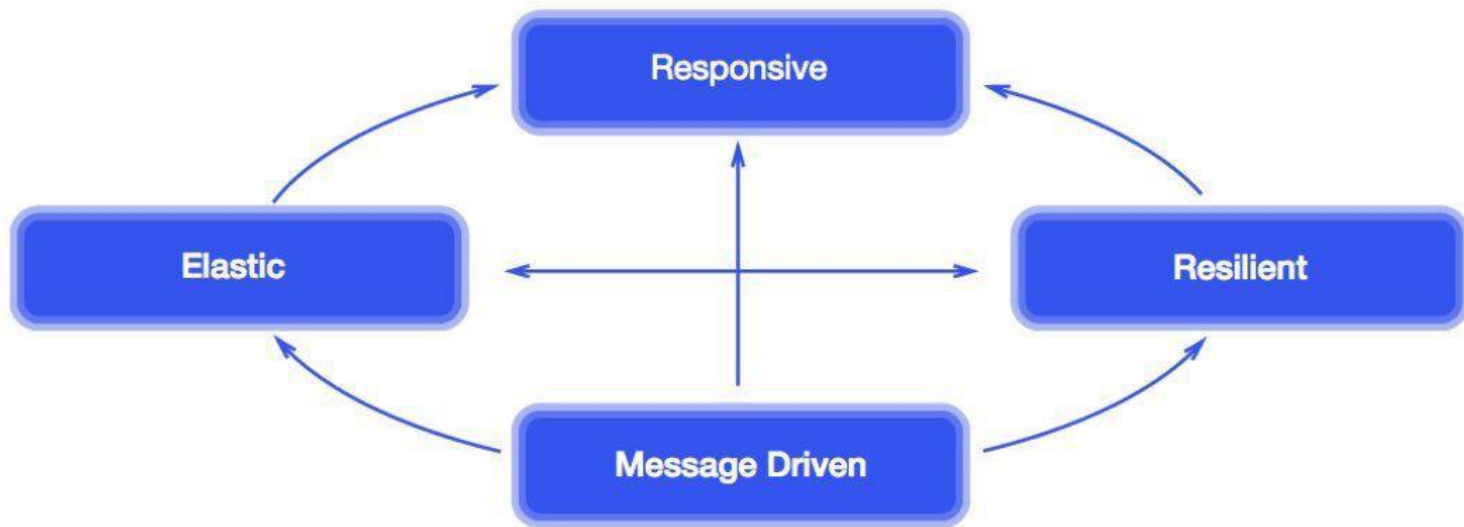


ACTIVIDAD: REACTIVIDAD

	A	B	C	D	E
1					
2		DATOS		SUMA	
3		10			
4		20		75	
5		15			
6		30			
7					

PRINCIPIOS

<https://www.reactivemanifesto.org/>



PRINCIPIOS

Responsivos

Aseguran la calidad del servicio cumpliendo unos tiempos de respuesta establecidos.

Además, define límites en dichos tiempos de respuesta, de forma que los problemas pueden ser detectados rápidamente y tratados de forma efectiva

Resilientes

Se mantienen responsivos incluso cuando se enfrentan a situaciones de error.

PRINCIPIOS

Elásticos

Se mantienen responsivos incluso ante aumentos en la carga de trabajo.

Orientación a mensajes

Minimizan el acoplamiento entre componentes al establecer interacciones basadas en el intercambio de mensajes de manera asíncrona.

Afectando (de manera positiva) todo el sistema.

BENEFICIOS

Escalabilidad:

Usando programación reactiva obtenemos una implementación débilmente acoplada, escalable y que tiende a aislar los fallos. La escalabilidad se refiere a la capacidad de escalar horizontalmente y de forma rápida.

Quiere decir, que se agregan más nodos al mismo, el rendimiento de éste mejora. Por ejemplo, al añadir una computadora nueva a un sistema que balancee la carga entre la antigua y la nueva puede mejorar el rendimiento de todo el sistema.

Esto aplica también en la programación, bien sea con servicios del lado del backend o módulos y/o componentes del lado del frontend.

Ahorro:

La utilización eficiente de los recursos, deriva en gastar menos en servidores y centros de datos. La promesa de la programación reactiva es que se puede hacer más con menos.

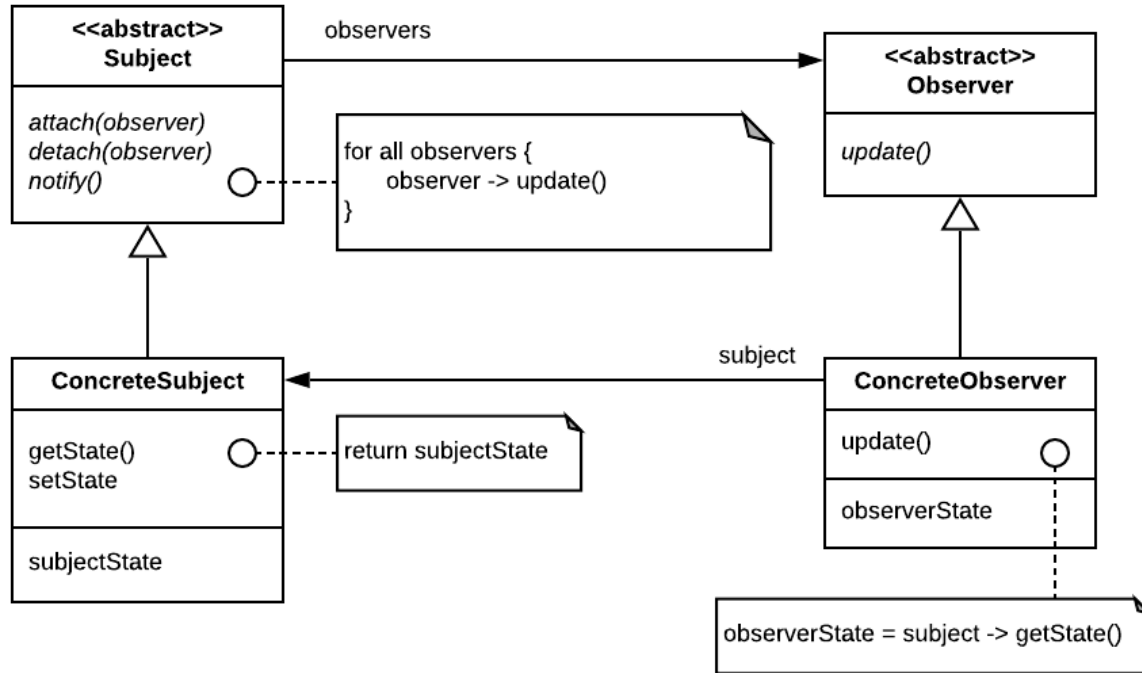
Específicamente puedes procesar cargas de trabajo más altas con menos hilos.

LENGUAJES

<http://reactivex.io/languages.html>

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Lua: RxLua
- Ruby: Rx.rb
- Python: RxPY
- Go: RxGo
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP
- Elixir: reaxive
- Dart: RxDart

PATRÓN DEL OBSERVADOR



ACTORES

- PRINCIPALES

- Flujo de datos
- El Observable
- El Observador

- ADICIONALES

- Dispatchers
- Operadores

EJEMPLOS CON RXJS

En este enlace encontraras ejemplos ilustrativos con RxJS:

<https://rxjs-playground.github.io/>

Gracias