



TA2

Curso: Open Source Software

SECCION: E52B

Integrantes:

- Vargas Ccora, Jhordy (U202112739)
- Acosta Valencia, Moises (U201723069)
- Chuquimani Barrientos, Luiggi (U201921630)
- Cahuana Montes, Juan Antonio (U201919919)
- Luque Callata, Eddy Efrain (U201720945)
- Prialé Revolo, Luis Alberto (U201418997)

Que es programación Reactiva



La programación reactiva es un paradigma **que se centra en trabajar de forma asíncrona** con flujos de datos finitos o **infinitos**. Su concepción y **desarrollo está relacionado con** la publicación del **Manifiesto Reactivo**, que **sentó** las bases de los sistemas **reactivos que debían ser**:

Responsivos

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

Resilientes

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

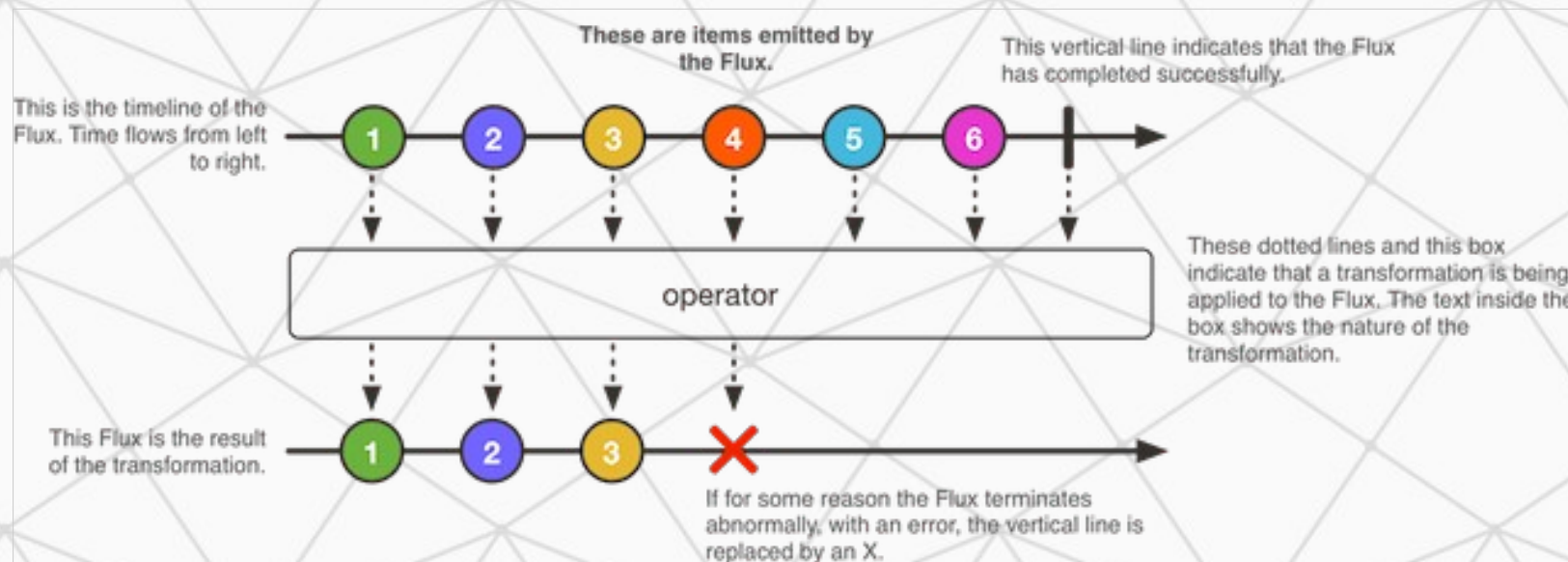
Elásticos

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

Orientados a mensajes

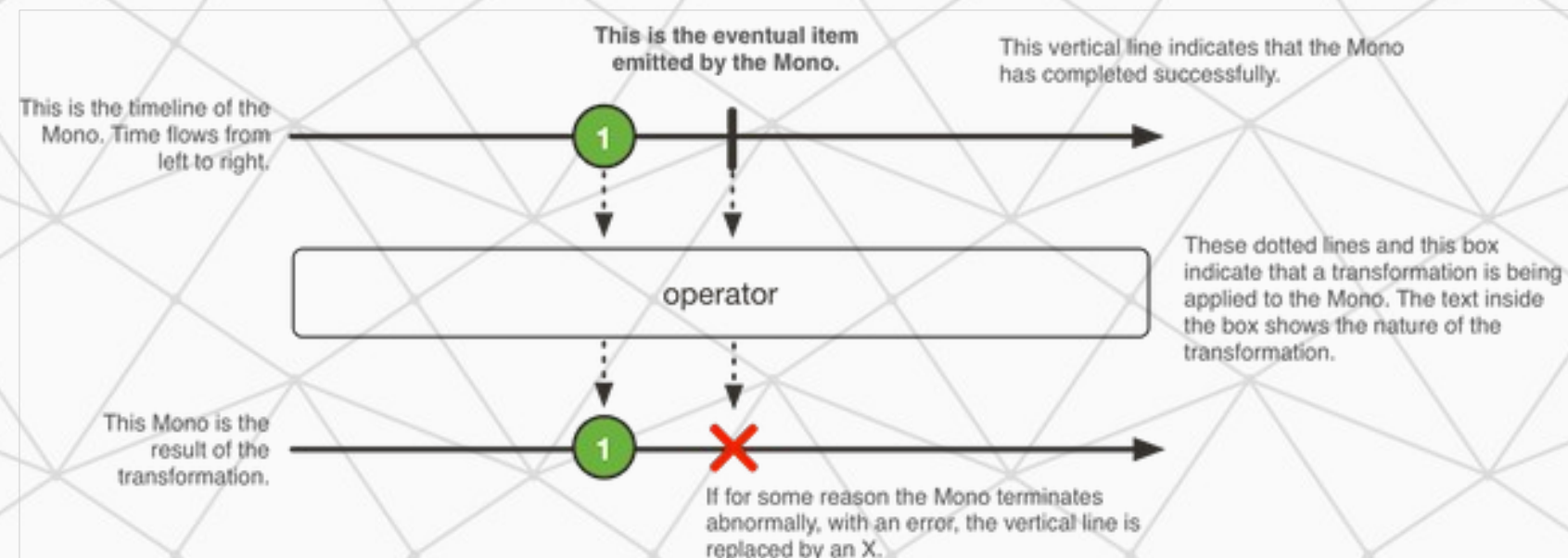
- Minimizan el acoplamiento entre componentes al establecer interacciones basadas en el intercambio de mensajes de manera asíncrona

Un esquema de cómo funciona



FLUX

MONO



Para que sirve, beneficios y desventajas



Para qué sirve

La programación reactiva implica modelar datos y eventos como flujos de datos observables e implementar rutinas de procesamiento de datos para reaccionar a los cambios en esos flujos.

Beneficios

- Mas flexible
- Resistente y escalable
- Más tolerante al fallo
- Usar los recursos de manera eficiente
- Menos latente
- Altamente receptivo

Desventajas

- *Puede ser difícil de aprender para algunos* : La programación reactiva tiene la reputación de ser difícil, por lo que algunos desarrolladores pueden tener dificultades para aprenderla inicialmente.
- *Más memoria* : Dado que la programación reactiva se basa en flujos a lo largo del tiempo, las aplicaciones tenderán a utilizar más memoria.

Introducción a WebFlux



Spring WebFlux proporciona soporte de programación reactivo, asíncrono y sin bloqueo para aplicaciones web en un formato de controlador anotado

Spring WebFlux, un enfoque diferente para crear aplicaciones web basadas en programación reactiva. Aplicaciones reactivas que utilizan WebFlux y la capa HTTP. Es un marco web reactivo y sin bloqueo construido en Project Reactor que admite la contrapresión de flujos reactivos y funciona en servidores sin bloqueo.

Ejemplo de controlador Spring WebFlux

```
@RestController
@RequestMapping("api /v1/room/reservation/ ")
public class ReservationResource {

    @GetMapping(ruta =("/{roomId}")
        public Mono<Reservación> getReservationById(@PathVariable
        Mono< String> roomId) {

        return //Llama a tu capa de servicio aquí
    }
}
```

¿Como se aplica? Ejemplo.



Conceptos

Para su aplicación se requiere manejar los conceptos previamente tratados, mencionamos algunos como:

- Comunicación asíncrona.
- Reactive Streams.
- Programación funcional.
- Otros.

Tecnologías y herramientas

Teniendo en cuenta el problema a solucionar, es importante tener en cuenta las tecnologías a usar:

- Base de datos.
- Arquitectura de la solución.
- Tipo de cliente

¿Como se aplica? Ejemplo.



Tecnologías seleccionadas

Las herramientas seleccionadas son las siguientes:

- Springboot y Webflux
- MongoDB: para un mejor aprovechamiento de funcionalidades reactivas.
- Cliente Angular distribuido o template HTML en el mismo servidor.

Ejemplo:

Problema:

Se busca servir un listado de productos para un sitio e-commerce con una alta demanda donde las respuestas de cada listado pueden tardar hasta 1 por producto.

Solución:

Se implementa programación reactiva no bloqueante para que los clientes puedan visualizar de manera progresiva los productos de una categoría, esta misma solución se puede aplicar luego a paginaciones.

Caso Práctico



Caso

Se requiere tener una listado de productos que puedan servirse a un cliente de manera progresiva sin bloquear los otros recursos necesarios en la aplicación cliente.

Aplicación desplegada:

Cliente 1:

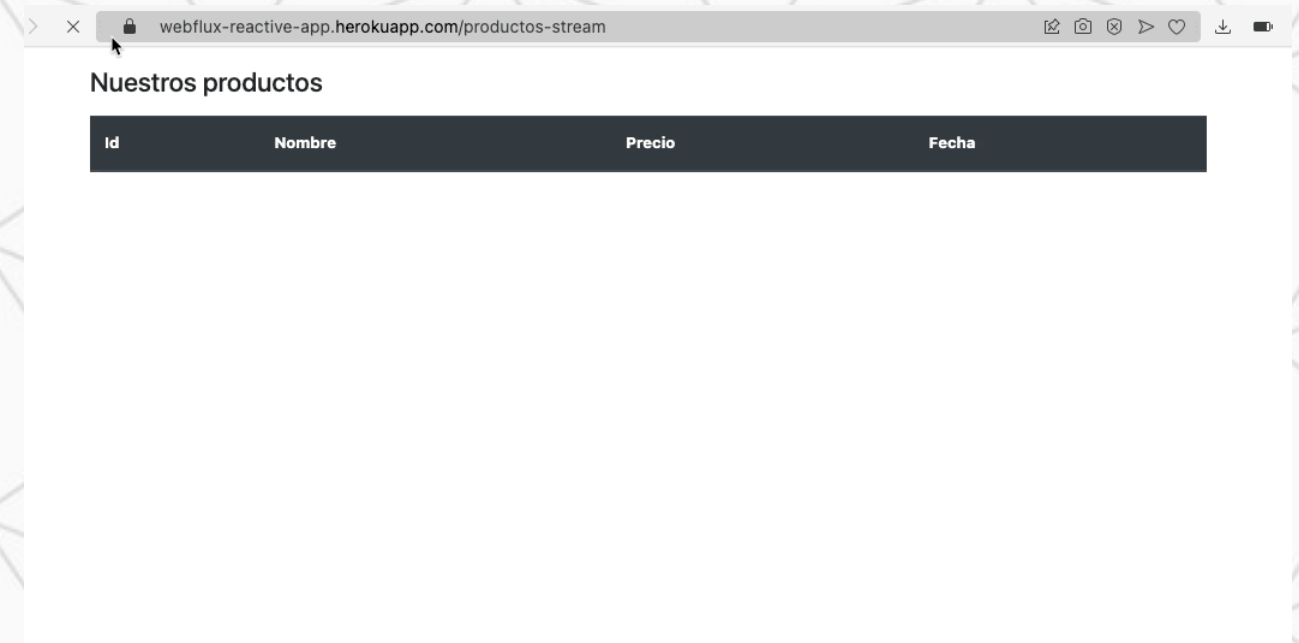
webflux-reactive-app.herokuapp.com/productos-stream

API Rest:

webflux-reactive-app.herokuapp.com/api/productos

Repositorio:

github.com/universidad-upc-open-source/programacion-reactiva



¿Qué es lo que más les ha impresionado o gustado?



Podemos mencionar algunos puntos que más nos costo entender y llamó la atención:

- El patrón observador puede ser implementado en diferentes tecnologías y lados de una aplicación.
- La capacidad de realizar notificaciones desde el servidor cambiando el enfoque pasivo a uno activo.
- Me llamó mucho la atención como ahora con Programación Reactiva puede alterar datos y mostrar a tiempo real información de la BD si poder realizar la consulta de manera manual.

Conclusiones y Recomendaciones.



Conclusiones:

- Se concluye que el paradigma de programación reactiva implementa el patrón observador haciendo que los objetos suscritos sean notificados a los eventos que estos fueron suscritos.
- Un sitio donde en lugar de ir nosotros activamente a buscar los datos, existen unos flujos de datos a los que nosotros reaccionamos cuando recibimos nueva información.

Recomendaciones:

- Para sacar el máximo provecho al paradigma de programación reactiva, se sugiere evaluar el soporte cada componente, para que tanto el cliente del usuario final, servidor y base de datos ofrezcan funcionalidades reactivas.
- Cuando se desarrolla una aplicación reactiva es esencial manejar adecuadamente los conceptos expresados en las especificaciones del manifiesto reactivo.

Recursos y Tecnologías



Tecnologías:

- IntelliJ
- Visual Studio
- MongoDB Atlas Database
- Motor de plantillas Thymeleaf

Recursos:

- <https://docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html>
- <https://www.thymeleaf.org/documentation.html>
- <https://blog.knoldus.com/a-basic-introduction-to-spring-webflux/>

Bibliografía



- Maluenda, R. (10 de marzo de 2017). *¿Qué es la Programación Reactiva? Una introducción* . Servicios de software de perfiles. <https://profile.es/blog/que-es-la-programacion-reactiva-una-introduccion/>
- Pulido, M. (23 de septiembre de 2019). *Todo lo que deberías saber sobre programación reactiva* . Slash Movilidad | Soluciones móviles. <https://slashmobility.com/blog/2019/09/programacion-reactiva/>
- (N.d.). Edu.Co. Retrieved November 14, 2022, from https://bibliotecadigital.udea.edu.co/bitstream/10495/16993/1/ZapataJuan_2019_ArquitecturaDesarrolloSoftware.pdf



Gracias!