

Fundamentos y Componentes de la Programación Reactiva

William Carrillo

Universidad Mariano Gálvez

Facultad de Ingeniería en Sistemas

2 de agosto de 2025

1. Introducción

La programación reactiva es un paradigma orientado a flujos de datos y propagación de cambios, ampliamente utilizado para construir aplicaciones modernas que requieren alta disponibilidad, escalabilidad y capacidad de respuesta. Este enfoque se sustenta en el manejo asíncrono de eventos y en una arquitectura basada en suscripciones, permitiendo la creación de sistemas resilientes, elásticos y orientados a mensajes. Este documento presenta los fundamentos del manifiesto reactivo y explica los principales actores del modelo de flujo de datos: Observable, Observer, Subject, Operator y Scheduler.

2. Fundamentos de las Aplicaciones Reactivas

Según el *Reactive Manifesto* (2014), una aplicación reactiva debe cumplir con los siguientes principios fundamentales:

- **Responsiva:** La aplicación responde de forma rápida y predecible, incluso bajo carga.
- **Resiliente:** El sistema continúa operando correctamente ante fallos parciales mediante estrategias como el aislamiento y la replicación.
- **Elástica:** La aplicación se adapta a variaciones en la carga mediante escalado dinámico.

- **Orientada a mensajes:** La comunicación entre componentes se realiza mediante el intercambio asíncrono de mensajes.

Estos principios buscan garantizar una experiencia de usuario fluida, robusta y escalable.

3. Flujo Básico en Programación Reactiva

El flujo básico de datos en una aplicación reactiva sigue los siguientes pasos:

1. Un **Observable** produce datos de manera asíncrona.
2. Los datos pasan por una secuencia de **Operadores** que los transforman, filtran o combinan.
3. Un **Observer** se suscribe al flujo para consumir los datos emitidos.

Este modelo sigue el patrón *publish-subscribe* (pub-sub), donde el flujo se ejecuta solo cuando hay suscriptores.

4. Componentes Clave

4.1. Observable

Es el productor del flujo de datos. Puede emitir uno o varios valores en el tiempo. En lenguajes como JavaScript se usa 'RxJS', mientras que en Java se usa 'Flux' y 'Mono' en 'Project Reactor'.

4.2. Observer

Es el consumidor del flujo. Define cómo reaccionar ante nuevos valores ('next'), errores ('error') y finalización ('complete'). Su rol es esencial para ejecutar las acciones en respuesta a los datos emitidos.

4.3. Operator

Permiten modificar los flujos mediante funciones como 'map', 'filter', 'merge', entre otros. Son fundamentales para crear flujos declarativos y legibles.

4.4. Subject

Es un componente híbrido que actúa como ‘Observable’ y ‘Observer’. Permite enviar datos manualmente y que múltiples observadores reciban la misma información.

4.5. Scheduler

Define el contexto de ejecución: cuándo y en qué hilo o cola se ejecutarán las operaciones. Es útil para controlar concurrencia, evitar bloqueos y distribuir cargas.

5. Ejemplos Prácticos

Ejemplo en RxJS (JavaScript)

```
fromEvent(input, 'input')
  .pipe(
    debounceTime(300),
    map(event => event.target.value)
  )
  .subscribe(value => {
    console.log('Búsqueda:', value);
  });
```

Ejemplo en Spring WebFlux (Java)

```
@GetMapping("/usuarios")
public Flux<Usuario> listarUsuarios() {
    return usuarioRepository.findAll();
}
```

6. Conclusiones

La programación reactiva representa un cambio significativo frente a los modelos tradicionales basados en peticiones bloqueantes. Permite desarrollar sistemas altamente escalables, tolerantes a fallos y con mejor desempeño. Comprender los componentes como ‘Observable’, ‘Observer’, ‘Operator’ y ‘Subject’, así como los fundamentos del Reactive Manifesto, es esencial para diseñar soluciones modernas y robustas.

7. Referencias Bibliográficas

- Reactive Manifesto. (2014). *The Reactive Manifesto*. Recuperado de: <https://www.reactivemanifesto.org/>
- Webber, J. (2020). *Reactive Systems Explained*. O'Reilly Media.
- Baeldung. (2023). *Introduction to Project Reactor*. Recuperado de: <https://www.baeldung.com/reactor-core>
- Oracle. (2022). *Reactive Programming in Java*. Recuperado de: <https://www.oracle.com/technical-resources/articles/java/reactive-programming.html>
- RxJS Documentation. (2023). *Reactive Extensions for JavaScript*. Recuperado de: <https://rxjs.dev>