

# Programação Reativa

**A** programação reativa é um paradigma que se concentra em como os sistemas reagem a eventos, utilizando fluxos de dados assíncronos. Diferente da programação sequencial convencional, a programação reativa é fundamentada em quatro características principais: responsividade, elasticidade, resiliência e ser guiada por mensagem. Este paradigma permite que aplicações sejam altamente adaptáveis e eficientes, reagindo rapidamente a eventos e mantendo a estabilidade mesmo diante de falhas. Nesta aula, exploraremos os fundamentos da programação reativa e como implementá-la em suas aplicações, utilizando frameworks como Spring WebFlux para facilitar o desenvolvimento.

## Fundamentos da Programação Reativa

A programação reativa é um paradigma que se concentra em como os sistemas reagem a eventos. Ao contrário da programação convencional, que segue uma abordagem sequencial, a programação reativa é baseada em fluxos de dados assíncronos e eventos. Esse paradigma se fundamenta em quatro principais características: responsividade, elasticidade, resiliência e ser guiado por mensagem.

**Responsividade** refere-se à capacidade do sistema de se adaptar a demandas específicas e responder rapidamente a eventos e solicitações.

**Elasticidade** envolve a capacidade do sistema de lidar com variações de carga e demanda, expandindo ou contraindo conforme necessário.

**Resiliência** é a capacidade do sistema de se recuperar de falhas e continuar funcionando, tratando erros de maneira a não comprometer a execução.

Por fim, ser **guiado por mensagem** significa que a comunicação assíncrona entre componentes é garantida, proporcionando flexibilidade e eficiência.

Esses fundamentos são essenciais para identificar e implementar a programação reativa em suas aplicações. Por exemplo, um sistema de notificações em tempo real, onde eventos como novos e-mails ou mensagens são tratados de forma assíncrona, exemplifica bem esses conceitos. No próximo tópico, exploraremos os desafios comuns na programação reativa.

## **Desafios Comuns na Programação Reativa**

A programação reativa, sendo orientada a eventos e fluxos de dados, apresenta desafios específicos. Os três principais desafios são a maior complexidade, a curva de aprendizado e o gerenciamento de estado.

**Maior complexidade** exige um conhecimento técnico avançado dos desenvolvedores. Implementações assíncronas, como métodos `await` em JavaScript ou multi-threading em Java, necessitam de um entendimento profundo para serem bem-sucedidas.

A **curva de aprendizado** refere-se ao tempo e esforço necessários para dominar as especificidades da programação reativa, que difere significativamente dos paradigmas imperativos e declarativos.

Por fim, o **gerenciamento de estado** envolve lidar com eventos assíncronos e garantir que cada estado do sistema seja corretamente gerenciado e atualizado conforme necessário.

Para superar esses desafios, é crucial estudar e entender a documentação e exemplos práticos de código. Por exemplo, ao trabalhar com JavaScript,

compreender como as Promises e o async/await funcionam pode facilitar a implementação de lógica assíncrona em uma aplicação web. A engenharia reversa de código existente também pode ajudar a entender como eventos disparam ações específicas no sistema.

## **Frameworks de Programação Reativa**

Frameworks reativos facilitam a implementação de chamadas assíncronas e a construção de pipelines de consumo de eventos. Eles proporcionam uma fundação simples para o processamento assíncrono, permitindo que os desenvolvedores se concentrem mais na lógica da aplicação do que nas complexidades técnicas.

O **Spring WebFlux** é um exemplo de framework reativo amplamente utilizado. Ele resolve problemas comuns, como a escalabilidade de aplicações, tempo de latência e solicitações concorrentes, oferecendo atividades assíncronas e não bloqueantes, fluxo de dados orientado a eventos e programação funcional. Frameworks como **React**, **Vue.js**, **Flutter** e **Wix** também são exemplos de frameworks reativos, cada um com suas especificidades e vantagens.

Para utilizar esses frameworks, é essencial consultar a documentação oficial e participar de comunidades de desenvolvedores para trocar experiências e soluções. A aplicação de frameworks reativos pode ser exemplificada em uma aplicação de chat em tempo real, onde mensagens são enviadas e recebidas de maneira assíncrona e reativa.

## **Aplicações Reativas com Spring**

O Spring Framework 5 suporta programação reativa via Reactive Streams, facilitando a implementação de streams de dados que reagem a eventos. O **Reactive Stream** é uma especificação que padroniza o uso de streams reativas dentro da JVM, permitindo um fluxo contínuo de dados com tratamento de eventos.

Os componentes principais são **Flux** e **Mono**. **Flux** lida com zero ou N elementos, enquanto **Mono** lida com zero ou um elemento. Ambos permitem a implementação de operações como filtragem, transformação e combinação de dados. Por exemplo, uma aplicação de monitoramento de temperatura que reage a mudanças no ambiente em tempo real pode ser implementada usando Flux para processar múltiplos eventos de temperatura e Mono para tratar eventos únicos, como alarmes de temperatura crítica.

Visualizar esses conceitos em diagramas ajuda a entender melhor como os eventos fluem e são processados em uma aplicação reativa. Na prática, é importante desenhar essas visualizações para debugar e otimizar o código reativo. No próximo módulo, iniciaremos a implementação prática de um projeto usando Spring WebFlux, aplicando todos os conceitos aprendidos até agora.

## **GitHub da Disciplina**

Você pode acessar o repositório da disciplina no GitHub a partir do seguinte link:

<https://github.com/FaculdadeDescomplica/Framework>. Esse espaço é o seu portal para mergulhar fundo no universo da aprendizagem interativa. Nele, você encontrará todos os códigos, além dos links para os arquivos e dados.

## **Conteúdo Bônus**

A documentação oficial do Spring WebFlux é uma fonte rica de informações sobre a utilização deste framework para construir aplicações reativas. Inclui

guias, tutoriais, e exemplos práticos que são essenciais para entender e implementar programação reativa em projetos Java.

**Título:** Spring Framework Documentation

**Plataforma:** [Docs.spring.io](https://docs.spring.io)

## Referência Bibliográfica

CARDOSO, L. da C. **Design de aplicativos**. Intersaberes: 2022

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7.ed. Pearson: 2018.

JOÃO, B. do N. **Usabilidade e interface homem-máquina**. Pearson: 2017

LEAL, G. C. L. **Linguagem, programação e banco de dados**: guia prático de aprendizagem. Intersaberes: 2015.

MEDEIROS, L. F. de. **Banco de dados**: princípios e prática. Intersaberes: 2013.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados**: implementação em SQL, PL/SQL e Oracle 11g. Pearson: 2013.

SETZER, V. W.; SILVA, F. S. C. **Bancos de dados**. Blucher: 2005.

VICCI, C. (Org.). **Banco de dados**. Pearson: 2014.

**Ir para exercício**