



Kresil - Kotlin Multi-Platform Library for Fault-Tolerance

Francisco Engenheiro, n.º 49428, e-mail: a49428@alunos.isel.pt, tel.: 928051992

Orientadores: Pedro Félix, e-mail: pedro.felix@isel.pt

Março de 2024

1 Introdução

1.1 Necessidade de Desenho de Software Resiliente

Os sistemas distribuídos representam um conjunto de computadores independentes e interligados em rede, que se apresentam aos utilizadores como um sistema único e coerente [?].

Dado a constante necessidade destes sistemas estarem disponíveis, aliados à sua complexidade de funcionamento, é natural que estejam suscetíveis a falhas de comunicação, de hardware, de software, entre outras. Por esse motivo, existe a necessidade de garantir que os serviços que disponibilizam sejam resilientes, e mais concretamente, tolerantes a falhas.

Um serviço tolerante a falhas, é um serviço que é capaz de manter a sua funcionalidade total ou parcial, ou apresentar uma alternativa, quando um ou mais componentes que lhes estão associados falham. De forma a alcançar este objetivo, foram desenhados mecanismos de resiliência. Alguns exemplos:

- **Retry:** Tenta novamente uma operação que falhou, aumentando a sua probabilidade de sucesso;
- **Rate Limiter:** Limita a taxa de requisições que um determinado serviço pode receber;
- **Circuit Breaker:** Interrompe, temporariamente, a comunicação com um serviço que está a falhar, de forma a evitar que o mesmo sobrecarregue o sistema. Semelhante a um disjuntor elétrico;
- **Fallback:** Fornece um valor ou executa uma ação alternativa caso uma operação falhe.

1.2 Mecanismos de Resiliência

Existem bibliotecas que fornecem mecanismos de resiliência (Tabela ??). Estes atuam em tempo de execução e implementam uma determinada estratégia. A configuração de um mecanismo de resiliência é feita através de uma política que define o seu comportamento.

Tabela 1: Exemplos de bibliotecas que fornecem mecanismos de resiliência.

Biblioteca	Linguagem	Plataforma
Netflix's Hystrix [?]]	Java	JVM
Resilience4j [?]]	Java/Kotlin	JVM
Polly [?]]	C#	.NET

A biblioteca *Polly* [?]] divide os mecanismos de resiliência em duas categorias:

- **Resiliência Reativa:** Reage a falhas e mitiga o seu impacto (e.g., *Retry*, *Circuit Breaker*);
- **Resiliência Proativa:** Previne que as falhas aconteçam (e.g., *Rate Limiter*, *Timeout*).