



Scala Fundamentals

Framework Akka

Michael Silva

O que é?

É um framework para criação de aplicações que são executadas na JVM e que necessitam lidar com alta concorrência, distribuição de processamento e tolerância a falhas.

Modelo de Atores

- Akka é baseado no modelo de atores
- Em um sistema de atores, tudo é ator, assim como os objetos no paradigma de orientação a objetos
- Os atores interagem e compartilham dados um com os outros sem, necessariamente, uma sequência definida
- Os atores se comunicam via *mensagens*

Ator

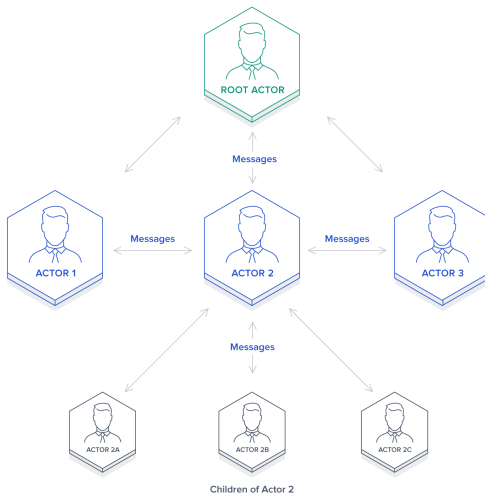
- Um ator nada mais é do que um objeto que recebe mensagens e executa ações a depender da mensagem
- A ação de um ator pode ser:
 - Executar um cálculo, persistir dados, realizar uma chamada externa e outros
 - Encaminhar a mensagem para outro ator
 - Criar um novo ator
 - Ignorar

Implementação de Atores

```
import akka.actor.Actor
import akka.actor.Props
import akka.event.Logging

// A classe deve estender a trait Actor
class MyActor extends Actor {
  // Metodo obrigatorio, as mensagens chegam por ele
  def receive = {
    // Mensagens que esse ator podera receber
    case value: String => doSomething(value)
    case _ => println("received unknown message")
  }
}
```

Sistema de atores



Comunicação entre os atores

- Ao criar um ator referente a uma classe (que estende ator), é gerado um *ActorRef*
- Um *ActorRef* é a referência para um ator
- Os atores tem *buffers* de entrada. Ex: FIFO
- Uma mensagem é enviada para um *ActorRef* através dos seguintes métodos:
 - '!' (tell) - Envia uma mensagem sem esperar a resposta
 - '?' (ask) - Envia uma mensagem e retorna um *Future*

Akka

Exemplo

Bibliografia

- [1] Castorina, Diego. *Concurrency and Fault Tolerance Made Easy*. Disponível em <https://www.toptal.com>