

MicroProfile Benefits for Monolithic Applications

Víctor Orozco

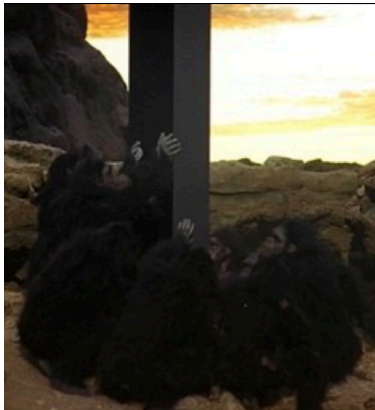
29 de Agosto de 2020

@tuxtor

Modular monoliths

—

Todo mundo odeia os monólitos . . . so que não



- Java vem criando sistemas "legados" desde 1995
- O mundo ainda tenta sair do Java 8
- Banca, governo, telco
- Nem todo mundo é Netflix

“Como consultor independente é difícil aceitar que muito do trabalho real é garantir a continuidade para sistemas legados”

Eu, bem triste mas com dinheiro na bolsa

June 18, 2020

What Is a Modular Monolith?

JAVA APPLICATION DEVELOPMENT | ENTERPRISE DEVELOPMENT

A modular monolith is a software design approach in which a monolith is designed with an emphasis on interchangeable (and potentially reusable) modules.

Figura 1: JRebel

2020 o ano da gourmetização do monólito

ARCHITECTURE & DESIGN

Upcoming Webinar: [How to pay down technical debt in JavaScript applications \(SEPT 22\)](#):

Modular Monolithic Architecture, Microservices and Architectural Drivers

LIKE

3



JAN 29, 2020 • 4 MIN READ

by

Jan Stenberg

FOLLOW

From observing what is currently happening in the IT community, [Kamil Grzybek](#) concluded in a recent article that most new projects are implemented using the microservices architecture. He believed that the IT industry at large is making a mistake adopting microservices just because they believe this will solve all of the problems with monolithic applications. Instead, Grzybek recommended we focus on [architectural drivers](#), and he emphasized that every architecture has its pros and cons – it will solve some problems but also create new problems. In a series of articles, he has started to describe the [basic concepts and properties of a modular monolith](#) and the [drivers](#)

RELATED CONTENT

**Operating Microservices & Se
InfoQ Live, 25th August**

AUG 18, 2020

**Building a Banking Infrastruct
Microservices: QCon London C**

AUG 13, 2020

Figura 2: InfoQ

Microserviços

Vantagens: Elasticidade, tolerância à falhas, responsividade

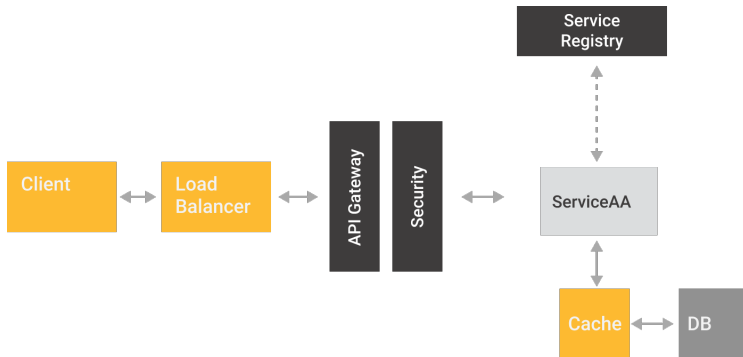


Figura 3: Microserviços

Desvantagens: Complexidade de desenvolvimento, integração é gestão

Modular monolith

O objetivo nunca foi criar um exercito de mini aplicativos. O objetivo sempre foi criar "reactive apps" ou seja aplicativos responsivos e escaláveis.

Modular monolith = Monólito criado de forma modular

1. Módulos intercambiáveis
2. Funcionalidade autônoma para cada modulo
3. Encapsulação garantida por contratos/interfaces

<https://www.kamilgrzybek.com/design/modular-monolith-primer/>

Modular monolith

Modular monolith = SOA modular sem XML



Eclipse MicroProfile

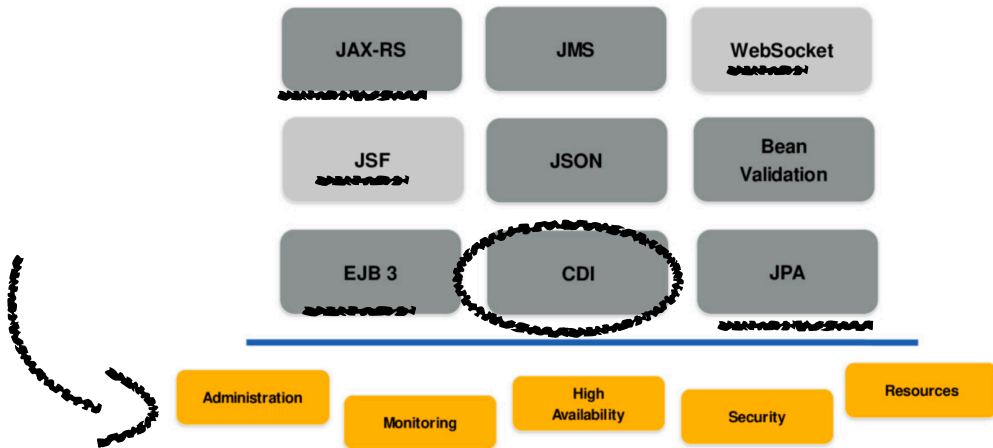
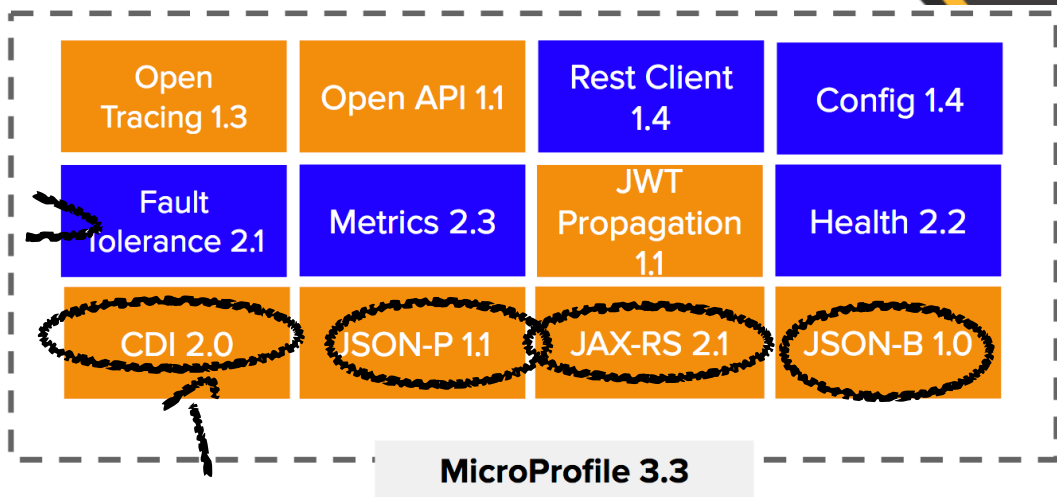




Figura 4: Credito: Reza Rahman



 = New
 = Updated



Bibliotecas

1. SmallRye
2. Apache Geronimo
3. Fujitsu Launcher



JEAS - FatJar, UberJar

1. DropWizard
2. KumuluzEE
3. Helidon (Oracle)
4. WebSphere/Open Liberty (IBM)
5. Quarkus (Red Hat)
6. Payara Micro
7. Apache TomEE



Micro server

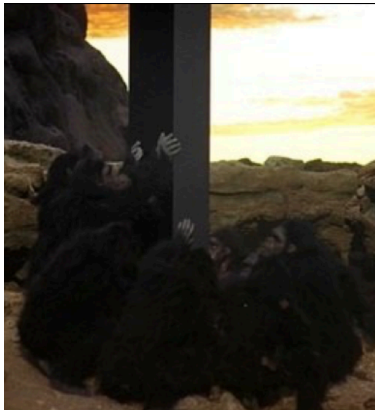
1. Payara Micro
2. Apache TomEE



Full server (Jakarta EE/Java EE)

1. Payara
2. JBoss / Wildfly
3. WebSphere/Open Liberty
4. Apache TomEE

Eclipse MicroProfile para "modular monolith"



- Integração = CDI
- APIs Micro em servers tradicionais
- Aproveitar os princípios de arquitetura micro em entornos tradicionais

Use cases comuns

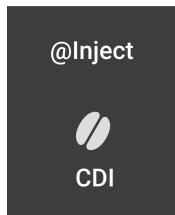
- Externalização da configuração (módulos intercambiáveis)
- Documentação de APIs para integradores e clientes (contratos e interfaces)
- Criação de comunicação Typesafe entre módulos via HTTP-Rest (contratos e interfaces)
- Tolerância à falhas sem complicações (funcionalidade autônoma)
- Gestão de métricas e observabilidade (funcionalidade autônoma)

MP Use cases

Configuração

- MP Config
- Alternativas: Apache Tamaya, DeltaSpike Config
- Bom para: Docker, K8s, Maven Profiles + Filtering
- Monólito: Configuração no deployment, alternativa simples ao JNDI, evita a necessidade de recompilação

Config



Props file

#!/bin/bash
Env var



Value

- MP OpenAPI
- Alternativas: Swagger Java
- Bom para: Documentação REST
- Monólito: Definição de integrações, documentações on-line

- MP TypeSafe REST Client
- Alternativas: Jersey, Hoodie
- Bom para: Integrações via HTTP
- Monólito: Separação de responsabilidade, eventual possibilidade de levar os módulos para microserviços



Tolerância â falhas

- MP Fault Tolerance
- Alternativas: Hystrix, ResilienceJ
- Bom para: Cotas, SLAs, Tolerância à falhas
- Monólito: APIs mais resilientes, independência na cadeia de confiança entre serviços

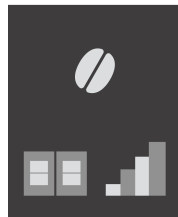
Fault Tolerance

~~a () X~~
aFallback()



- MP Metrics
- Alternativas: Metrics CDI
- Bom para: Docker, K8s
- Monólito: Evitar complexidade do JMX, MBeans, monitoramento via http

Metrics



Performance

Demo time!

- Payara
- Docker
- Oracle cloud

Objetivo: Olá mundo, modular, autônomo e intercambiável

Java 11, JAX-RS, CDI, MicroProfile

<https://github.com/tuxtor/MP-Workshop>



**Oracle
Groundbreakers**



ORACLE®
Certified Professional
Java SE 8 Programmer

ORACLE®
Certified Associate
Java SE 8 Programmer

- me@vorozco.com
- [@tuxtor](https://tuxtor)
- <http://vorozco.com>
- <http://tuxtor.shekalug.org>



This work is licensed under a
Creative Commons
Attribution-ShareAlike 3.0.