

Eclipse MicroProfile para el desarrollador ocupado

Víctor Orozco - Cesar Hernandez

26 de agosto de 2019

@tuxtor - @CesarHgt



¿Microservicios?

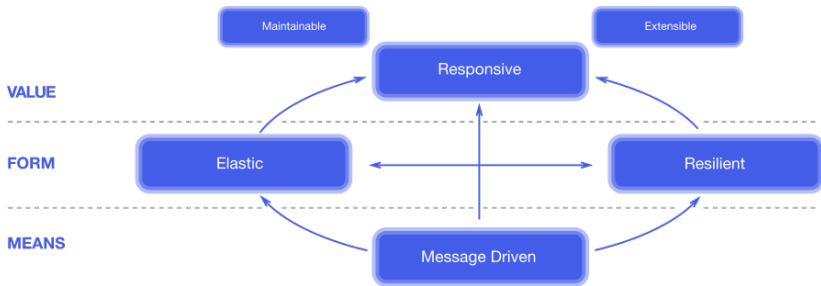
¿Microservicios?

¿Necesito migrar todo a microservicios? ¿Que pasa si no utilizo microservicios? ¿Me estoy volviendo viejo?



Reactive applications

Aplicaciones reactivas



Microservicios son una (de muchas) herramienta para creación de aplicaciones reactivas

Monolito - Escalabilidad

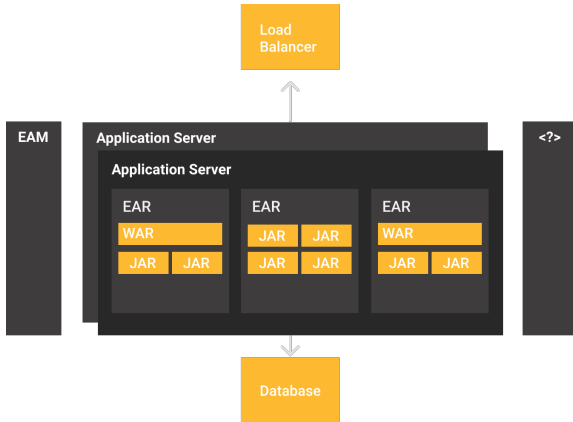


Figura 1: Monólito

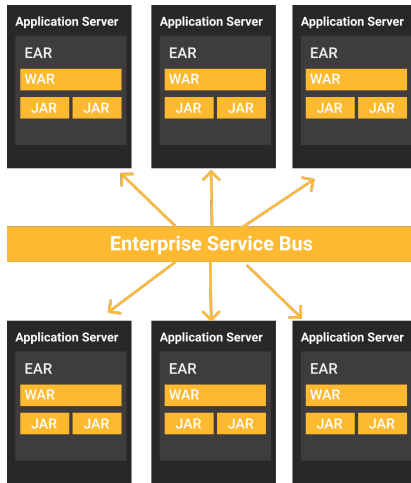


Figura 2: ESB

Microservicios

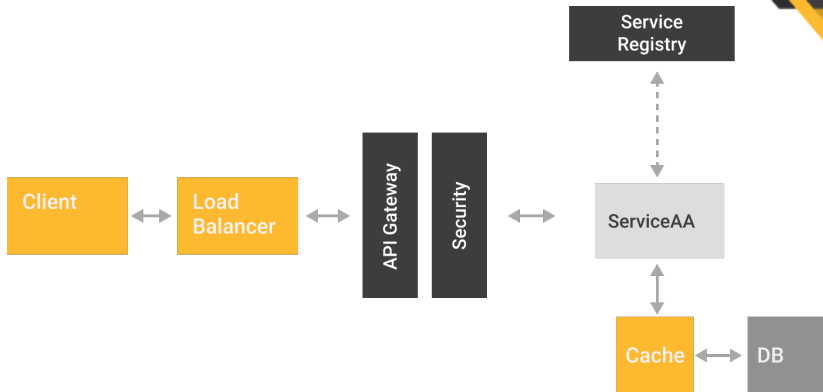


Figura 3: Microservicios

Java EE 8

Java EE 8



Batch	Dependency Injection	JACC	JAXR	JSTL	Management
Bean Validation	Deployment	JASPIC	JMS	JTA	Servlet
CDI	EJB	JAX-RPC	JSF	JPA	Web Services
Common Annotations	EL	JAX-RS	JSON-P	JavaMail	Web Services Metadata
Concurrency EE	Interceptors	JAX-WS	JSP	Managed Beans	WebSocket
Connector	JSP Debugging	JAXB			
JSON-B	Security				

Java EE 8

- Integración JSF-CDI
- Integración JMS-CDI
- HTTP/2
- JSON-B
- Security
- **JAX-RS Reactivo**



JAKARTA EE



JAVA EE
GUARDIANS



MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA



NABENIK

Eclipse MicroProfile

Eclipse MicroProfile

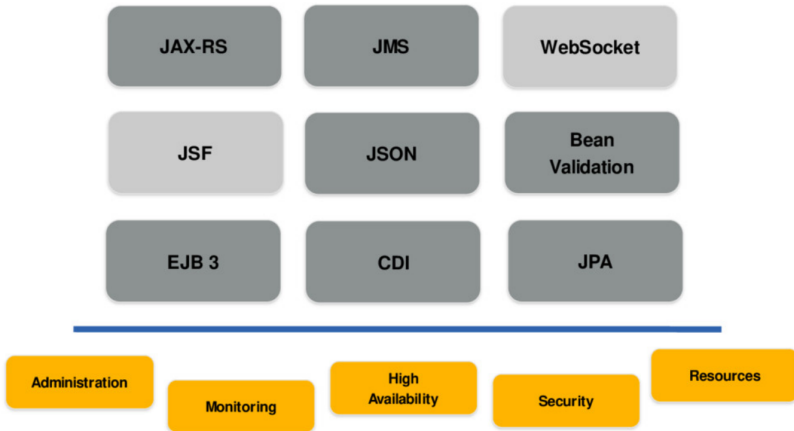
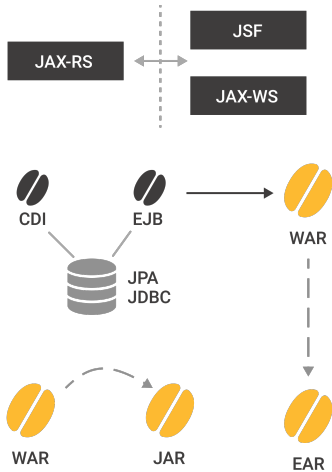


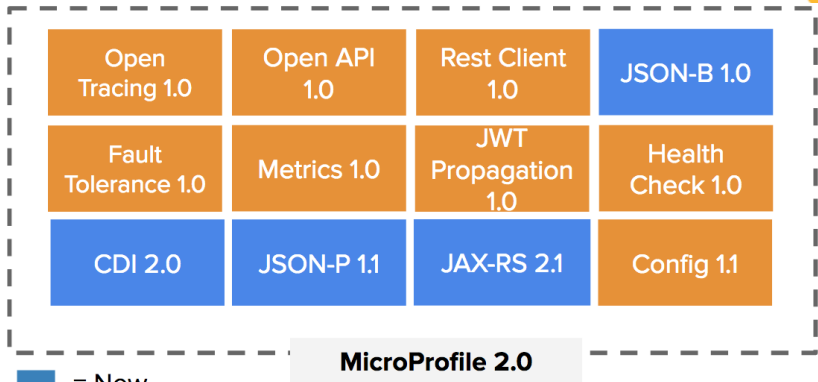
Figura 4: Credito: Reza Rahman



Eclipse MicroProfile



BV + JSON-P
JSON-B + Security

Eclipse MicroProfile



-  = New
-  = No change from last release

Bibliotecas

- SmallRye (Red Hat)
- Hammock
- Apache Geronimo
- Fujitsu Launcher

JEAS - Fat Jar

- Dropwizard
- KumuluzEE
- Helidon (Oracle)
- Open Liberty (IBM)
- Apache Meerowave
- Thorntail (Red Hat)
- Quarkus (Red Hat)

Micro server

- Payara Micro
- TomEE JAX-RS

Full server

- Payara Application Server
- JBoss Application Server / Wildfly Application Server
- WebSphere Liberty (IBM)

<https://wiki.eclipse.org/MicroProfile/Implementation>

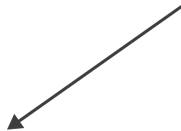
Eclipse MicroProfile on Payara 5

```
<dependency>  
    <groupId>org.eclipse.microprofile</groupId>  
    <artifactId>microprofile</artifactId>  
    <type>pom</type>  
    <version>2.0.1</version>  
    <scope>provided</scope>  
</dependency>
```

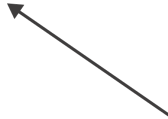
Config



Props file



`#!/bin/bash`
Env var



Value

```
@Inject  
@ConfigProperty(name = ".%mdbservice.url")  
String omdbDaemonServiceUrl;
```

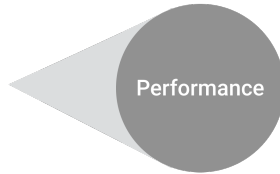
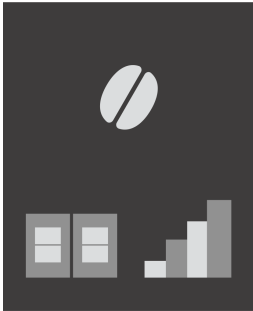
Ext. de la configuración (VM, Docker, Kubernetes)

Fault Tolerance

~~a () X~~
aFallback()

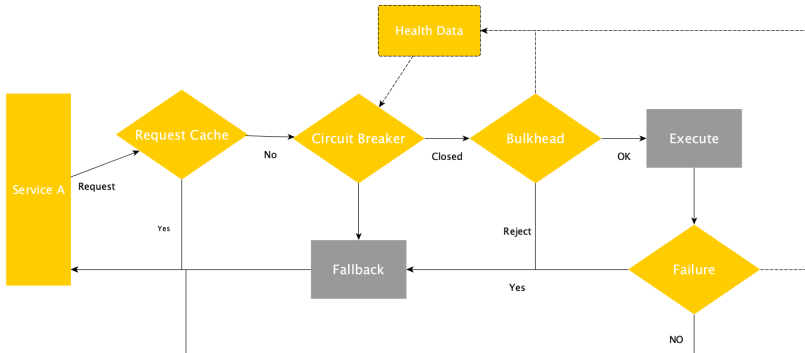


Metrics



Fault Tolerance + Metrics

- *Fault Tolerance* depende de la existencia de metricas, las metricas se exponen mediante *Metrics*



Reglas de evaluación y alternativas

- Circuit Breaker
- Bulkhead
- Retry
- Timeout
- Fallback

Fault tolerance - Fallback, Timeout

```
@GET
@Path("/{id:[a-z]*[0-9][0-9]*}")
@Fallback(fallbackMethod = "findByIdFallBack")
@Timeout(TIMEOUT)
public Response findById(@PathParam("id")
final String imdbId) {
    ...
}

public Response findByIdFallBack(@PathParam("id")
final String imdbId) {
    ...
}
```

- JSON or OpenMetrics (Prometheus)
- Vendor
- Base
- Application

¿Cuales?

- Counted
- Gauge
- Metered
- Timed
- Histogram

Metrics - Counted

```
@Inject
@Metric
Counter failedQueries;

@GET
@Path("/{id:[a-z]*[0-9][0-9]*}")
@Fallback(fallbackMethod = "findByIdFallBack")
@Timeout(TIMEOUT)
public Response findById(@PathParam("id")
final String imdbId) {
    ...
}

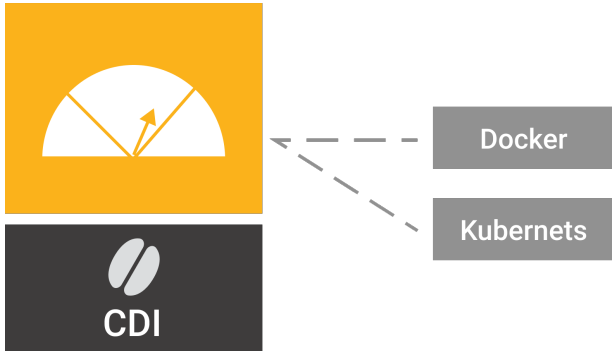
public Response findByIdFallBack(@PathParam("id")
final String imdbId) {
    ...
    failedQueries.inc();
}
```

Inc-dec en tiempo real

```
@Gauge(unit = ".ExternalDatabases",name = "movieDatabases", absolute  
= true)  
public long getDatabases() {  
    return 99; //Any value  
}
```

/metrics/application/movieDatabases

Health check



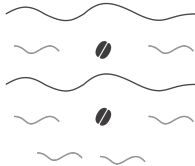
- OK?
- How much ok?

¿Estas vivo?

```
@Override
```

```
public HealthCheckResponse call() {  
    return HealthCheckResponse.named("TaVivoAinda")  
        .withData("key1", "val1")  
        .withData("key2", "val2")  
        .up()  
        .build();  
}
```

JWT



@Inject
Principal _____

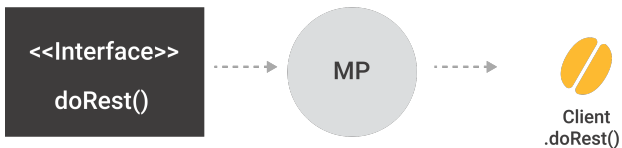
@Inject
Realm _____


```
@LoginConfig(authMethod = "MP-JWT")
public class ApplicationConfig extends Application {
}

@Inject
private JsonWebToken jwtPrincipal;

@Inject
@Claim("email")
private String email;
```

Type Safe



```
@Path("/playlist")
@Consumes("application/json")
public interface MusicPlaylistService {

    @GET
    List<String> getPlaylistNames();

    @PUT
    @Path("/{playlistName}")
    long updatePlayList(@PathParam("playlistName")
        String name,
        List<Song> playlist)
        throws UnknownPlaylistException;
}
```

Demo

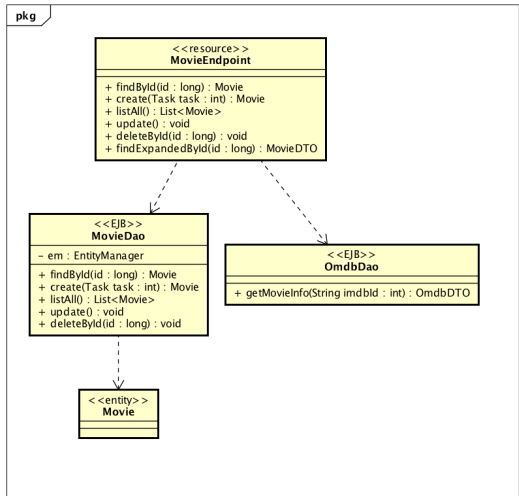
Java 8, JAX-RS, CDI, EJB, MicroProfile

`https://github.com/tuxtor/payara-demo`

`https://github.com/tuxtor/omdb-demo`

Stacks tradicionales

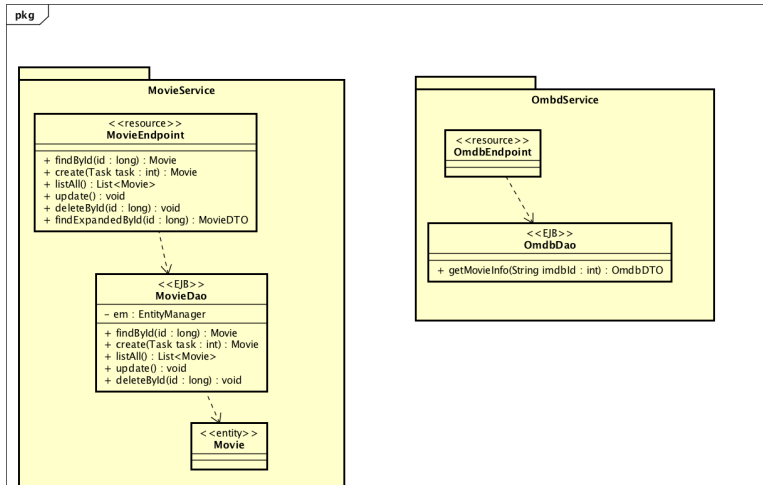
- EJB
- JTA
- JAX-RS
- CDI



MicroProfile: JAX-RS, CDI, Config, Fault Tolerance, Metrics

Payara Micro: EJB, JTA

Fatores externos: Location, Deployment, Orchestration, Balancing, Consistency



12 factores cloud native (Heroku)

Microprofile

- Config
- Backing service
- Disposability

Cloud

- Codebase (Git-Flow)
- Dependencies (Maven)
- Build, Release, Run
- Processes (Pipelines)
- Port binding
- Concurrency (Docker - k8s)
- Dev / Prod parity
- Logs
- Admin process

MENU

ORACLE
Cloud Infrastructure

Containers

Clusters

Registry

Registry

Create Repository

tuxtor

- microprofile (Public)
- microprofile/omdb-demo
- microprofile/payara-demo (Public)

1

microprofile/payara-demo


User: vorozca@nabenik.com

Created: an hour ago

Access: Public


Readme

No readme has been created yet for this repository.

 MENU

ORACLE[™]
Cloud Infrastructure

Compute » Instances » Instance Details



RUNNING

instance-20181206-0243

Create Custom ImageStartStopRebootTerminateApply Tag(s)

Instance InformationTags

Instance Information

Availability Domain: hgWe:US-ASHBURN-AD-1

Fault Domain: FAULT-DOMAIN-2

Region: iad

Shape: VM.Standard2.1

Virtual Cloud Network: [vcn20181206044411](#)

Maintenance Reboot: -

Primary VNIC Information

```
tuxtor@millenium-falcon-2:~$ docker tag omdb-demo iad.ocir.io/tuxtor/microprofile/omdb-demo:latest
```

```
tuxtor@millenium-falcon-2:~$ docker push iad.ocir.io/tuxtor/microprofile/omdb-demo
The push refers to repository [iad.ocir.io/tuxtor/microprofile/omdb-demo]
31d661ce120c: Pushing [=====>] 952.3kB/1.46MB
5dd1fd455c13: Layer already exists
2aa3decaee68: Layer already exists
685fdd7e6770: Layer already exists
c9b26f41504c: Layer already exists
cd7100a72410: Layer already exists
```

Stateful Rules

Source: 0.0.0.0/0	IP Protocol: TCP	Source Port Range: All	Destination Port Range: 22	Allows: TCP traffic for ports: 22 SSH Remote Login Protocol
Source: 0.0.0.0/0	IP Protocol: ICMP	Type and Code: 3, 4		Allows: ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
Source: 10.0.0.0/16	IP Protocol: ICMP	Type and Code: 3		Allows: ICMP traffic for: 3 Destination Unreachable
Source: 0.0.0.0/0	IP Protocol: TCP	Source Port Range: All	Destination Port Range: 8080-8085	Allows: TCP traffic for ports: 8080-8085



Oracle
Groundbreakers



ORACLE
Certified Professional
Java SE 8 Programmer

ORACLE
Certified Associate
Java SE 8 Programmer

- vorozco@nabenik.com
- @tuxtor
- <http://www.nabenik.com>



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