Red Hat Application Development II: Implementing Microservice Architectures (DO283)

- **Formato do curso:** Presencial
- **Localidade:** Lisboa
- **Data:** 02 Dez. 2020 a 07 Dez. 2020
- **Preço:** 2180€
- **Horário:** Laboral - das 09h00 às 17h00
- **Duração:** 27 horas

Develop microservice-based applications in Java EE with MicroProfile and OpenShift

Building on [Red Hat Application Development I: Programming in Java EE (JB183)](#), the introductory course for Java EE application development, Red Hat Application Development II: Implementing Microservice Architectures (JB283) emphasizes learning architectural principles and implementing microservices in Java EE, primarily based on MicroProfile with WildFly Swarm and OpenShift.

You will build on Java EE application development fundamentals and focus on how to develop, monitor, test, and deploy modern microservices applications. Many enterprises are looking for a way to take advantage of cloud-native architectures, but many do not know the best way to go about it. These enterprises have monolithic applications written in Java Enterprise Edition (JEE).

This course is based on Red Hat® Enterprise Application Platform 7.4 and Wildfly Swarm 2018.3.3.

Learn to

- Deploy and monitor microservice-based applications.
- Implement a microservice with MicroProfile.
- Implement unit and integration tests for microservices.
- Use the config specification to inject data into a microservice.
- Create a health check for a microservice.
- Implement fault tolerance in a microservice.
- Secure a microservice using the JSON Web Token (JWT) specification.

Diagnóstico de Competências

Teste previamente os seus conhecimentos, ou os da sua equipa, em:

- Red Hat Satellite
- Ansible
- RH JBoss Enterprise Application Platform
- RH Gluster Storage
- RH OpenShift
- RH OpenStack Platform
- RH Enterprise Linux 7
- RH Fuse
- RH Camel
- RH AMQ
- RH Ceph Storage
- RH Identify Management
- RH Enterprise Linux 8

[Aceda aqui ao diagnóstico!](#)

Destinatários

- This course is designed for Java developers.

Pré-requisitos

- Attend [Introduction to OpenShift Applications (DO101)](#) or demonstrate equivalent experience
- Attend [Red Hat Application Development I: Programming in Java EE (JB183)](#) or demonstrate equivalent experience
- Be proficient in using an integrated development environment such as Red Hat®Developer Studio or Eclipse
- Experience with Maven is recommended, but not required

Objectivos

Impact on the organization

Many organizations are struggling with how to make the move from monolithic applications to applications based on microservices, as well as how to reorganize their development paradigm to reap the benefits of microservice development in a DevOps economy. In particular, many organizations are invested in Java programming frameworks and Red Hat® OpenShift Container Platform. This course exposes you to the Wildfly Swarm runtime for streamlined deployment on OpenShift clusters.

Impact on the individual

As a result of attending this course, you will understand how to develop, monitor, test, and deploy microservice-based Java EE applications using Wildfly Swarm and Red Hat OpenShift.

You should be able to demonstrate these skills:

- Design a microservices-based architecture for an enterprise application.
- Implement fault tolerance and health checks for microservices.
- Secure microservices to prevent unauthorized access.

Programa

Describe microservice architectures

- Describe components and patterns of microservice-based application architectures.

Deploy microservice-based applications

- Deploy portions of the course case study applications on an OpenShift cluster.

Implement a microservice with MicroProfile

- Describe the specifications in MicroProfile, implement a microservice with some of the specifications, and deploy it to an OpenShift cluster.

Test microservices

- Implement unit and integration tests for microservices.

Inject configuration data into a microservice

- Inject configuration data from an external source into a microservice.

Create application health checks

- Create a health check for a microservice.

Implement fault tolerance

- Implement fault tolerance in a microservice architecture.

Develop an API gateway

- Describe the API gateway pattern and develop an API gateway for a series of microservices.

Secure microservices with JWT

- Secure a microservice using the JSON Web Token specification.

Monitor microservices

- Monitor the operation of a microservice using metrics, distributed tracing, and log aggregation.

Note: Course outline is subject to change with technology advances and as the nature of the underlying job

evolves.