
# Facultad de ingeniería

## Técnico en Ingeniería en Computación

## MicroProfile: Desarrollo de Microservicios en Java

**Estudiantes:**

- Elías Daniel Franco Rodríguez RF230727
- Marlon Osmin Ortiz Cárcamo OC232936
- Mauricio Enrique Herrera Rico HR230334

**Docente:**

- Ing. Mauricio Saca

## Introducción

MicroProfile es un conjunto de herramientas de código abierto creado para mejorar el desarrollo de microservicios en Java. Nació en 2016 como una respuesta a la necesidad de tener una plataforma más ligera y rápida para construir aplicaciones modernas, especialmente aquellas basadas en microservicios, que no se estaban cubriendo completamente con Java EE (ahora conocido como Jakarta EE).

El objetivo de MicroProfile es hacer que el desarrollo de aplicaciones empresariales y servicios web sea más fácil, rápido y escalable. Para ello, ofrece soluciones como la configuración externa, la tolerancia a fallos, el monitoreo de rendimiento, la generación de documentación de APIs y la seguridad en las comunicaciones entre servicios. Además, es compatible con herramientas y plataformas modernas como Docker y Kubernetes, lo que lo hace ideal para aplicaciones que se ejecutan en la nube.

## Historia y origen de MicroProfile

MicroProfile nació en 2016 como una respuesta a un problema que muchos desarrolladores de Java estaban enfrentando: Java EE (lo que hoy conocemos como Jakarta EE) estaba avanzando muy lentamente en términos de innovación, especialmente cuando se trataba de arquitecturas modernas como los microservicios.

Varios miembros de la comunidad se juntaron y dijeron: "Oye, necesitamos algo más ligero y rápido para construir microservicios con Java", y así nació MicroProfile. Su primera versión se basó en tecnologías que ya existían en Java EE, como CDI (para la inyección de dependencias), JAX-RS (para crear APIs REST) y JSON-P (para manejar datos en JSON).

Poco después, MicroProfile se unió a la Eclipse Foundation, lo que le dio más estructura y apoyo. Desde entonces, ha seguido evolucionando y agregando nuevas especificaciones como configuración dinámica, métricas, OpenAPI y rastreo de llamadas (tracing), todo enfocado en hacer que el desarrollo de microservicios en Java sea más ágil y eficiente.

Hoy en día, MicroProfile es una opción sólida para quienes quieren construir aplicaciones empresariales basadas en microservicios sin depender de soluciones pesadas.

## Arquitectura fundamental de MicroProfile

MicroProfile fue diseñada con el propósito de facilitar el desarrollo de microservicios en Java, principalmente en Componentes de Arquitectura, algunos de ellos son:

- **CDI (para la inyección de dependencias):** Facilita la gestión del ciclo de vida de objetos dentro de la aplicación. Permite crear aplicaciones modulares y desacopladas.
- **JAX-RS (para crear APIs REST):** Facilita la creación de servicios web RESTful, permitiendo exponer APIs de manera sencilla y estándar.
- **JSON-P (para manejar datos en JSON):** JSON-P permite manejar datos en formato JSON de forma manual (parseo, construcción de objetos JSON).
- **JSON-B** proporciona una forma automática y sencilla de convertir objetos Java en JSON y viceversa.

## ¿Cuáles son las características clave de MicroProfile?

- **Configuración externa (MicroProfile Config):** Permite importar configuraciones desde fuera del contenedor, facilitando la adaptación de la aplicación a diferentes entornos sin necesidad de recompilar.
- **Tolerancia a fallos (MicroProfile Fault Tolerance):** Ofrece mecanismos para manejar la indisponibilidad de servicios, implementando patrones como reintentos, circuit breakers y fallbacks para aumentar la resiliencia de las aplicaciones.

- **Documentación de APIs (MicroProfile OpenAPI):** Permite generar documentación interactiva y detallada de las APIs RESTful, facilitando su comprensión y uso por parte de desarrolladores y sistemas integradores.
- **Autenticación JWT (MicroProfile JWT Auth):** Facilita la propagación de información de identidad y seguridad entre microservicios utilizando tokens web JSON (JWT), asegurando una comunicación segura y estandarizada.
- **Monitoreo y métricas (MicroProfile Metrics):** Proporciona herramientas para recopilar y exponer métricas de rendimiento y estado de la aplicación, esenciales para el monitoreo en tiempo real y la toma de decisiones informadas
- **Chequeo de salud (MicroProfile Health):** Ofrece endpoints para verificar el estado de salud de los microservicios, facilitando la detección temprana de problemas y la implementación de estrategias de recuperación

## ¿Cuáles son las ventajas y desventajas de MicroProfile?

### Ventajas de utilizar MicroProfile en proyectos de desarrollo:

- **Estandarización:** Al ser una especificación respaldada por organizaciones como IBM, Red Hat y Apache, garantiza consistencia y facilita la colaboración entre equipos.

- **Flexibilidad:** No está ligado a un servidor de aplicaciones específico, permitiendo ejecutar aplicaciones como JARs independientes o desplegarlas en servidores compatibles.
- **Integración con herramientas modernas:** Compatible con tecnologías como Docker y Kubernetes, facilitando la implementación y orquestación de microservicios en entornos de contenedores.
- **Comunidad activa:** Cuenta con el respaldo de una comunidad vibrante y en crecimiento, lo que asegura mejoras continuas y soporte actualizado.

**Desventajas de utilizar MicroProfile:**

- **Madurez relativa:** Aunque las APIs subyacentes tienen una trayectoria sólida, MicroProfile como especificación es relativamente reciente, lo que puede implicar menos recursos y casos de estudio disponibles.
- **Documentación en español limitada:** La mayoría de la documentación y recursos están en inglés, lo que puede representar una barrera para desarrolladores hispanohablantes.
- **Curva de aprendizaje:** Requiere familiarizarse con patrones y conceptos como inyección de dependencias (DI), programación orientada a aspectos (AOP) y servicios RESTful, lo que puede ser desafiante para quienes no tienen experiencia previa en estos temas.

## ¿Existe un lenguaje de programación específico asociado con el Frameworks MicroProfile? En caso afirmativo, ¿Cuál es y cómo se relaciona con el Frameworks MicroProfile?

Sí, el lenguaje de programación asociado con el framework MicroProfile es Java, MicroProfile está ligado a Java, pero su diseño modular permite usarlo en otros lenguajes JVM. Sin embargo, Java sigue siendo el principal debido a su integración con Jakarta EE y las APIs estándar.

Relación entre Java y MicroProfile:

MicroProfile está diseñado para aplicaciones Java empresariales: Se creó como una extensión ligera de Jakarta EE (antes Java EE) para el desarrollo de microservicios en Java.

Utiliza APIs estándar de Java EE/Jakarta EE: Como JAX-RS (para servicios REST), CDI (Inyección de dependencias) y JSON-P/B (para manipulación de JSON).

Soporte para Microservicios en Java: Proporciona herramientas y especificaciones diseñadas para facilitar la creación de aplicaciones Java en arquitecturas de microservicios.

Implementaciones en servidores de aplicaciones Java: Servidores como Quarkus, Open Liberty, Payara Micro y Helidon son compatibles con MicroProfile y están escritos en Java.

MicroProfile está completamente basado en Java y extendido a partir de Jakarta EE para facilitar el desarrollo de microservicios modernos en este lenguaje.

## Proyectos que utilizan este framework de MicroProfile

Sí, existen varios proyectos y aplicaciones conocidos que utilizan el framework MicroProfile para desarrollar microservicios en Java. A continuación, se presentan algunos ejemplos:

**Helidon:** Un conjunto de bibliotecas en Java para escribir microservicios. Helidon es una implementación de MicroProfile que facilita la creación de aplicaciones basadas en microservicios.

**Apache TomEE:** Un servidor de aplicaciones que combina Apache Tomcat con implementaciones de varias tecnologías Java EE, compatible con MicroProfile para facilitar el desarrollo de microservicios.

**IBM Cloud & Open Liberty:** IBM adopta MicroProfile en su runtime Open Liberty para aplicaciones empresariales en la nube. Servicios bancarios que requieren tolerancia a fallos.

## Casos de uso comunes de MicroProfile

- **Desarrollo de microservicios Java** en entornos empresariales.
- **Aplicaciones nativas para la nube**, integradas con Docker y Kubernetes.
- **Sistemas distribuidos** que requieren resiliencia, monitoreo y escalabilidad.
- **Servicios RESTful** que necesitan seguridad, documentación y configuración externa.
- **Aplicaciones que migran desde Java EE/Jakarta EE** hacia arquitecturas más ligeras.

## Recomendaciones para desarrolladores al usar MicroProfile

- **Conocer Jakarta EE:** Facilita el aprendizaje, ya que muchas APIs provienen de allí.
- **Comenzar con una implementación ligera:** Como Quarkus, Helidon u Open Liberty.
- **Aprovechar la comunidad:** Explorar foros, documentación oficial y ejemplos reales.
- **Integrar herramientas modernas:** Usar Docker, Kubernetes y CI/CD desde el inicio.
- **Adoptar buenas prácticas:** Como la inyección de dependencias, tolerancia a fallos y chequeo de salud.

## Bibliografía


 Eclipse Foundation. (2020, diciembre 23). *MicroProfile 4.0 is now available!* MicroProfile. https://microprofile.io/2020/12/23/microprofile-4-0-is-now-available/

 Oracle. (s.f.). *Definition of the Java Platform*. Oracle Java Tutorials. https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

 Krill, P. (2018, febrero 14). *El perfil de microservicios de Java obtiene capacidades de tolerancia a fallos*. CIO. https://www.cio.com/article/2081086/el-perfil-de-microservicios-de-java-obtiene-capacidades-de-tolerancia-a-fallos.html

- Gamboa, A. (2022). *MicroProfile: introducción al perfil de microservicios en Java*. Blog de Adam Gamboa. https://blog.adamgamboa.dev/es/microprofile-overview-2

- Open Liberty. (s.f.). *Using MicroProfile with Open Liberty*. https://openliberty.io/docs/latest/microprofile.html

- IBM. (s.f.). *Developing microservices with MicroProfile*. IBM Documentation. https://www.ibm.com/docs/es/cics-ts/6.x?topic=server-developing-microservices-microprofile