

Soluciones JavaEE, arquitectura de referencia

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
**Consejería de Sanidad, Presidencia
y Emergencias**

Contenido

- [Dirección General de Sistemas de Información y Comunicaciones](#)
 - [Áreas de Gobierno Tecnológico de SSII y del Dato](#)
- [1. Thin Server](#)
 - [1.1 - Tecnología de referencia](#)
 - [1.2 - Catálogo de verificaciones:](#)
 - [1.3 - Documentación de referencia:](#)
- [2. Fat Server](#)
 - [2.2 - Especificaciones de Weblogic](#)

Resumen

• Versión: v04r01 • Fecha publicac ión: 14 nov 2022 • Entrada en vigor desde: 28 feb 2023	Arquitectura tecnológica de referencia Java EE PRE-RELEASE • Versión: v04r01 • Estado: Pre- Release • Fecha publicación: 14 nov 2022 • Entrada en vigor desde: -
--	---

Cumplimiento normativo



Las normas expuestas son de obligado cumplimiento. La STIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la STIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la STIC.

La STIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y /o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la STIC para proceder a su validación técnica.

Contacto Arquitectura: l-arquitectura.stic@juntadeandalucia.es

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabezas de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	Pre-release	Adopción	Activa	Retiro	Alcance
v04r01	14/11 /2022	-	-	-	• Normativa microservicios en Java • JDK 11.x.x • Microprofile 3.x.x • OpenLiberty 20.x.x.x
v03r01	30 sept 2019	30 sept 2019	30 sept 2019		• Nueva versión de la arquitectura de referencia
v02r01	1 jun 2017	1 jun 2017	1 jun 2017	30 sept 2019	• Actualización de la normativa de selección del servidor de aplicaciones JEE7 a Weblogic Server 12.2.1 • Cambio de JDK7 a JDK8
v01r01	3 oct 2016	3 oct 2016	3 oct 2016	1 jun 2017	• Normativa inicial de uso de selección del servidor de aplicaciones JEE6 Weblogic Server 12.1.3

1. Thin Server

La arquitectura microservicios es una aproximación a un modelo de sistemas distribuidos con un estilo que adopta el enfoque de división en componentes. Concretamente es una división que permite la construcción de servicios, [entendido s estos como pequeñas unidades de negocio independientes](#), que pueden funcionar a su vez en conjunto para ofrecer servicios de más alto nivel, comunicándose entre sí a través de peticiones, como por ejemplo HTTP/HTTPS o Websocket , e incluso pueden estar desarrollados en tecnologías diferentes. Este tipo de servicios nos permite contar con infraestructuras de TI adaptables y flexibles, ya que para modificar un único servicio no es necesario alterar el resto de la infraestructura.

Este modelo de arquitectura toma como enfoque la delimitación clara de las unidades de negocio y al independizarlas mantiene las premisas de los principios de responsabilidad asegurando un bajo acoplamiento mejorando la cohesión del código.

Los microservicios suelen presentar las siguientes características:

- **Autonomía:** Son piezas software independientes que suelen ser desplegadas como servicios aislados, habitualmente sobre plataformas PaaS bajo uso de contenedores como Docker.
- **Heterogeneidad:** La pila tecnológica de referencia puede ser diferente entre los microservicios ya que en caso de tener que interactuar entre ellos se realizará mediante mecanismos de petición respuesta haciendo uso de una API que cada uno de ellos expone.
- **Resiliencia:** Este es un factor clave que lo distingue de la arquitectura monolítica, y es que cuando un microservicio falla, este no debe afectar al funcionamiento de los demás. Con esto se consigue que en estos casos se podría tener una degradación de funcionalidades pero no fallas del sistema completo. Esto se consigue con mecanismos de tolerancia a fallos.
- **Escalabilidad:** Debido a que son piezas software de menor tamaño permiten hacer un escalado mucho más óptimo y utilizar de forma eficiente los recursos de infraestructura permitiendo escalar solo los microservicios necesarios y evitando el desperdicio habitual que puede generar el escalado de las arquitecturas monolíticas.
- **Facilidad de despliegue:** Al ser de un tamaño reducido se minimiza el riesgo asociado al cambio, ya que el nº de pruebas a realizar es menor y se puede realizar una mayor cobertura de estas. Ante fallos el problema suele aislarse fácilmente, y se pueden liberar rápidamente nuevas versiones. Suelen estar muy vinculados al concepto de "Despliegue Continuo" automatizando tanto un conjunto de pruebas de regresión como el despliegue en los entornos.
- **Reutilización de servicios:** Facilita enormemente la reutilización, permitiendo por ejemplo que un microservicio pueda ser consumido por una aplicación web, dispositivo móvil, o cualquier otro dispositivo.
- **Cohesión:** Son piezas de software que resuelven problemas muy concretos (cohesionados).

Para profundizar más sobre los conceptos asociados a los microservicios se puede tomar como referencia los siguientes enlaces:

- <https://12factor.net/es/>

- <https://martinfowler.com/articles/microservices.html>
- <https://microservices.io/>
- <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/>
- <https://www.cncf.io/>
- <https://microprofile.io/>

1.1 - Tecnología de referencia

La arquitectura de referencia para el desarrollo de microservicios en el SAS está basada en Java. Tradicionalmente el desarrollo en Java ha estado marcado por el uso de estándares, y en el caso de aplicaciones empresariales estos se aglutinaron en forma de plataforma bajo la denominación Java EE. Hace algún tiempo Oracle ha cedido el proyecto Java a la comunidad Eclipse y este finalmente ha pasado a denominarse Jakarta. La última versión de esta plataforma es [Jakarta EE 8](#) la cual pone el foco en el desarrollo cloud-native.

En relación a los microservicios la comunidad Eclipse ha elaborado [Microprofile](#) el cual permite a los desarrolladores de Java EE aprovechar su conjunto de habilidades existente mientras cambian su enfoque de las aplicaciones tradicionales a microservicios. Este se compone de un subconjunto de estándares Java EE que son extendidos para abordar patrones comunes en el uso de microservicios.

El beneficio de adoptar [MicroProfile](#) es que sus especificaciones son impulsados por la comunidad, gratuitos y abiertos, y de las cuales existen múltiples implementaciones eliminando el riesgo del denominado "vendor lock-in" y mantiene la libertad de elección de los desarrolladores. Este continúa evolucionando entregando aproximadamente tres lanzamientos anuales.

La arquitectura de referencia para el desarrollo de microservicios en el SAS estará basada en [Microprofile](#) y en el estándar [Jakarta EE 8](#) para aquellos requerimientos técnicos necesarios y no cubiertos actualmente por [Microprofile](#).

Especificación de referencia:	Microprofile
Implementación de referencia:	Cualquier implementación verificada por Eclipse para la versión de Microprofile (https://wiki.eclipse.org/MicroProfile/Implementation).
Versión JDK:	Open JDK 11.x (LTS) con la última versión update existente

Especificaciones a utilizar en la versión de Microprofile:	• MicroProfile Config • MicroProfile Fault Tolerance • MicroProfile Health Check • MicroProfile JWT Authentication • MicroProfile Metrics • MicroProfile OpenAPI • MicroProfile OpenTracing • MicroProfile Rest Client • CDI • Common Annotations • JAX-RS • JSON-B • JSON-P
Conectividad a BBDD:	Se utilizará exclusivamente JPA 2.2 con alguna de las siguientes implementaciones: • Eclipselink 2.7.x • Hibernate 5.5.x

1.2 - Catálogo de verificaciones:

Identificador	Descripción	Carácter
ARQ.REF.MSC.01	Uso de versión Microprofile.	OBLIGATORIO
ARQ.REF.MSC.02	Implementación de conexión a BBDD.	OBLIGATORIO

1.3 - Documentación de referencia:

- <https://12factor.net/es/>
- <https://martinfowler.com/articles/microservices.html>
- <https://microservices.io/>
- <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/>
- <https://www.cncf.io/>
- <https://microprofile.io/>
- Ebook Microprofile
- Documentación Microprofile 3.x

2. Fat Server

Java Enterprise Edition o Java EE es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación, similar a otras especificaciones del Java Community Process (JCP), Java EE es un estándar debido a que los proveedores deben cumplir ciertos requisitos para declarar que sus productos son conformes al mismo.

Java EE tiene varias especificaciones de API, tales como JDBC, EJB, CDI, JMS, JAX-WS, JAX-RS, etc. y define cómo coordinarlos. Ello permite al desarrollador crear una aplicación de empresa portable entre plataformas y escalable, a la vez que integrable con otras tecnologías. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de desarrollo y mantenimiento de bajo nivel.

En los siguientes apartados se detalla las especificaciones técnicas de los servidores existentes en la STIC, siendo por tanto la norma que debe seguirse de forma estricta.

2.1 - Plataforma de ejecución del servidor

Arquitectura de ejecución	x86-64bits (AMD64/EM64T)
Sistema Operativo	Oracle Linux Release 7 Update 3 x86 64 bits
Java	Oracle JDK 1.8
Servidor de aplicaciones	Oracle Weblogic Suite 12cR2 12.2.1.3

2.2 - Especificaciones de Weblogic

Partiendo de que el servidor de aplicaciones corporativo actualmente es Weblogic Server 12.2.1.3, es necesario tener en cuenta que esta versión cumple la especificación de Java EE7 al completo. Por otro lado, hay que considerar el último PSU disponible para la citada versión de WLS (12.2.1.3) indicada en la siguiente URL de Oracle:

- <https://blogs.oracle.com/fusionmiddlewaresupport/july-2019-oracle-weblogic-server-patch-set-update-have-been-released#one>
 - Patch: 29814665
 - Unique Patch ID: 22937216
 - Patch description: "WLS PATCH SET UPDATE 12.2.1.3.0(ID:20190522.070630)"

En el siguiente listado se muestra el conjunto de versiones concretas disponibles en esta versión de weblogic:

Especificación	Weblogic 12.2.1.3
Batch Application Processing (JSR 352)	1.0
Contexts and Dependency Injection for Java EE	1.1
Dependency Injection for Java EE	1.0

Concurrent Managed Objects (JSR 236)	1.0
Expression Language (EL)	3.0, 2.2, 2.1, 2.0 Only JSP 2.0 and greater supports Expression Language 2.x.
Java API for JSON Processing (JSR-353)	1.0
Java API for XML-Based Web Services (JAX-WS)	2.2, 2.1, 2.0
Java API for RESTful Web Services (JAX-RS)	2.0
Java API for WebSocket	1.1
JavaBeans Activation Framework	1.1
Java EE	7.0
Java EE Application Deployment	1.2
Java EE Bean Validation	1.1
Java EE Common Annotations	1.2
Java EE Connector Architecture	1.7
Java EE EJB	3.2, 3.1, 3.0, 2.1, 2.0, and 1.1
Java EE Enterprise Web Services	1.3, 1.2, 1.1
Java EE Interceptors	1.2
Java EE JDBC	4.0, 3.0
Java EE JMS	2.0, 1.1, 1.0.2b
Java EE JNDI	1.2
Java EE JSF	2.2, 2.1.*, 2.0, 1.2, 1.1
Java EE JSP	2.3, 2.2, 2.1, 2.0, 1.2, and 1.1 JSP 1.2 and 1.1 include Expression Language (EL), but do not support EL 2.x or greater.
Java EE Managed Beans	1.0
Java EE Servlet	3.1, 3.0, 2.5, 2.4, 2.3, and 2.2
Java RMI	1.0
JavaMail	1.5
Java Transaction API	1.2
JAX-B	2.2, 2.1, 2.0
JAX-P	1.3, 1.2, 1.1
JAX-R	1.0
JAX-RPC	1.1
JDKs	8.0 (8.0 and 7.0 for clients) See JDK 8 Certification for details.
JMX	2.0

JPA	2.1, 2.0., 1.0
JSR 77: Java EE Management	1.1
JSTL	1.2
Managed Beans	1.0
OTS/JTA	OTS 1.2 and JTA 1.2
RMI/IIOP	1.0
SOAP Attachments for Java (SAAJ)	1.3, 1.2
Streaming API for XML (StAX)	1.0
Web Services Metadata for the Java Platform	2.0, 1.1

Para mas detalle sobre el entorno Weblogic que finalmente se monta podeis consultar la documentacion de CTI aqui: <http://bit.ly/2McFWXD>

La arquitectura de referencia establecida por norma para el desarrollo de aplicaciones con tecnología Java EE está basada en el principio de Separación de Problemas (**SoC, Separation of Concerns**). Este principio permite el estudio de cada uno de los componentes de un sistema de forma aislada. Es un principio fundamental en el diseño de sistemas complejos empresariales y de la programación Orientada a Objetos.

Este principio aplicado junto a el patron **BCE** para la especificacion de Java EE resulta en la separación lógica indicada en la [Arquitectura de Referencia Común](#).