



# Descripción general técnica de Sun Java Enterprise System 5



Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Referencia: 820-0888  
Marzo de 2007

Sun Microsystems, Inc. posee derechos de propiedad intelectual en relación con la tecnología incluida en el producto descrito en este documento. De forma específica y, sin limitación, entre estos derechos de propiedad intelectual se incluyen una o varias patentes en los EE.UU. o aplicaciones pendientes de patente en los EE.UU. y otros países.

Derechos del gobierno de Estados Unidos: software comercial. Los usuarios gubernamentales están sujetos al acuerdo de licencia estándar de Sun Microsystems, Inc. y a las disposiciones aplicables de la regulación FAR y sus suplementos.

Esta distribución puede incluir materiales desarrollados por terceras partes.

Algunas partes del producto pueden proceder de los sistemas Berkeley BSD, con licencia de la Universidad de California. UNIX es una marca comercial registrada en EE.UU. y en otros países, cuya licencia se otorga exclusivamente a través de X/Open Company, Ltd.

Sun, Sun Microsystems, el logotipo de Sun, el logotipo de Solaris, el logotipo de la taza de café de Java, docs.sun.com, Java y Solaris son marcas comerciales o marcas comerciales registradas de Sun Microsystems, Inc. en los EE.UU. y en otros países. Todas las marcas comerciales de SPARC se utilizan bajo licencia y son marcas comerciales o marcas registradas de SPARC International, Inc. en EE.UU. y en otros países. Los productos con marcas registradas de SPARC se basan en una arquitectura desarrollada por Sun Microsystems, Inc.

La interfaz gráfica de usuario de OPEN LOOK y Sun<sup>TM</sup> fue desarrollada por Sun Microsystems, Inc. para sus usuarios y licenciatarios. Sun reconoce los esfuerzos pioneros de Xerox en la investigación y desarrollo del concepto de las interfaces gráficas o visuales de usuario para el sector de la informática. Sun dispone de una licencia no exclusiva de Xerox para la interfaz gráfica de usuario de Xerox, que también cubre a los licenciatarios de Sun que implementen las GUI de OPEN LOOK y que, por otra parte, cumplan con los acuerdos de licencia por escrito de Sun.

Los productos descritos y abordados en esta publicación están sometidos a la legislación de control de exportaciones de los EE.UU. y pueden estar sujetos a leyes de importación o exportación de otros países. Queda terminantemente prohibido el uso final (directo o indirecto) de esta documentación para el desarrollo de armas nucleares, químicas, biológicas, de uso marítimo nuclear o misiles. Se prohíbe estrictamente la exportación o reexportación a países bajo el embargo de los EE.UU. o a entidades incluidas en la lista de exclusión de exportación de los EE.UU., incluidas, pero no limitándose a, las personas rechazadas y a las listas nacionales designadas específicamente.

ESTA DOCUMENTACIÓN SE PROPORCIONA "TAL COMO ESTÁ" Y NO SE ASUMIRÁ LA RESPONSABILIDAD DE NINGUNA CONDICIÓN EXPRESA O IMPLÍCITA, NI DE REPRESENTACIONES NI GARANTÍAS, INCLUIDA CUALQUIER GARANTÍA IMPLÍCITA DE APROVECHAMIENTO, IDONEIDAD PARA UN FIN PARTICULAR O NO INFRACCIÓN, EXCEPTO EN EL CASO EN QUE TALES RENUNCIAS DE RESPONSABILIDAD NO SE CONSIDEREN LEGALMENTE VÁLIDAS.

# Contenido

---

<b>Prefacio</b> .....	11
<b>1 Introducción a Java Enterprise System</b> .....	15
¿Por qué necesita Java ES? .....	15
Componentes de Java ES .....	17
Componentes de servicios del sistema .....	18
Componentes de calidad del servicio .....	20
Componentes compartidos .....	21
Componentes de los conjuntos de Sun Java .....	22
Cómo trabajar con Java ES .....	24
Ciclo de vida de las soluciones de Java ES .....	24
Escenarios de adopción de Java ES .....	27
Términos clave de este capítulo .....	29
<b>2 Arquitecturas de soluciones de Java ES</b> .....	31
Estructura arquitectónica de Java ES .....	31
Dimensión 1: dependencias de los servicios de infraestructuras .....	33
Dimensión 2: capas lógicas .....	38
Dimensión 3: calidad del servicio .....	41
Síntesis de las tres dimensiones arquitectónicas .....	45
Arquitectura de una solución de Java ES de ejemplo .....	45
Escenario de comunicaciones de las empresas .....	46
Arquitectura lógica para el escenario de ejemplo .....	46
Arquitectura de implementación para el escenario de ejemplo .....	48
Términos clave de este capítulo .....	48

<b>3</b>	<b>Funciones de integración de Java ES</b> .....	51
	El programa de instalación integrado de Java ES .....	51
	Comprobación del software previo .....	52
	Comprobación de dependencias .....	52
	Configuración inicial .....	52
	Desinstalación .....	53
	Servicios de supervisión del sistema .....	53
	Identidad integrada y servicios de seguridad .....	54
	Identidad única .....	54
	Autenticación e inicio de sesión único .....	55
	Autorización .....	57
	Términos clave de este capítulo .....	57
<b>4</b>	<b>Ciclo de vida de las soluciones de Java ES</b> .....	59
	Tareas del ciclo de vida de las soluciones .....	59
	Etapa previa a la implementación .....	61
	Implementación .....	62
	Diseño de la implementación .....	62
	Implementación de despliegue .....	64
	Etapa posterior a la implementación .....	67
	Términos clave de este capítulo .....	67
<b>A</b>	<b>Componentes de Java ES</b> .....	69
	Componentes de servicios del sistema .....	69
	Access Manager 7.1 .....	70
	Application Server Enterprise Edition 8.2 .....	71
	Directory Server Enterprise Edition 6.0 .....	71
	Java DB 10.1 .....	72
	Message Queue 3.7 UR 1 .....	72
	Portal Server 7.1 .....	72
	Service Registry 3.1 .....	73
	Web Server 7.0 .....	73
	Componentes de calidad del servicio .....	73
	Componentes de disponibilidad .....	74
	Componentes de acceso .....	76

Componentes de supervisión ..... 77

Componentes compartidos ..... 77

**Índice** .....79



# Lista de tablas

---

TABLA 1-1	Componentes de los servicios del sistema de Java ES .....	19
TABLA 1-2	Componentes de disponibilidad de Java ES .....	20
TABLA 1-3	Componentes de acceso de Java ES .....	21
TABLA 1-4	Componentes de los conjuntos de Sun Java .....	22
TABLA 1-5	Categorías de usuarios para las tareas del ciclo de vida de Java ES .....	26
TABLA 1-6	Consideraciones sobre los escenarios de adopción de Java ES .....	28
TABLA 2-1	Relaciones entre los componentes de los servicios del sistema de Java ES .....	37
TABLA 2-2	Calidades de servicio que afectan a la arquitectura de solución .....	42
TABLA 2-3	Componentes de calidad de servicio y calidades de sistema afectadas .....	43
TABLA 2-4	Resumen de requisitos de negocio: escenario de comunicaciones .....	46



# Lista de figuras

---

FIGURA 1-1	Compatibilidad necesaria para aplicaciones de empresa distribuidas .....	16
FIGURA 1-2	Categorías de los componentes de Java ES .....	18
FIGURA 1-3	Etapas del ciclo de vida de las soluciones y categorías de usuarios .....	25
FIGURA 2-1	Dimensiones de la arquitectura de una solución de Java ES .....	32
FIGURA 2-2	Dimensión 1: niveles de los servicios de infraestructuras .....	34
FIGURA 2-3	Componentes de los servicios del sistema de Java ES .....	36
FIGURA 2-4	Dimensión 2: capas lógicas para aplicaciones de empresa distribuidas .....	38
FIGURA 2-5	Messaging Server : ejemplo de arquitectura en capas .....	41
FIGURA 2-6	Arquitectura lógica para el escenario de comunicaciones de la empresa .....	47
FIGURA 3-1	La entrada única de usuario es compatible con varios servicios .....	55
FIGURA 3-2	Secuencia de autenticación .....	56
FIGURA 3-3	Secuencia de autorización .....	57
FIGURA 4-1	Tareas del ciclo de vida de las soluciones .....	60
FIGURA 4-2	Especificación de un escenario de implementación .....	61
FIGURA 4-3	Conversión de un escenario de implementación en una arquitectura de implementación .....	63



# Prefacio

---

La *Descripción general técnica de Sun Java Enterprise System 5* presenta los aspectos básicos técnicos y conceptuales de Java™ Enterprise System (Java ES). También describe los componentes, la arquitectura, los procesos y las funciones de Java ES.

En este documento, se describen la terminología y los conceptos técnicos utilizados en el conjunto de documentación de Java ES. Los términos técnicos clave se describen en la última sección de cada capítulo.

## Usuarios a los que está destinada esta guía

Esta guía está destinada a individuos que diseñan, implementan o mantienen soluciones de software basadas en Java ES, entre los que se incluyen analistas de negocios, arquitectos de sistemas, ingenieros técnicos y administradores de sistemas.

Los usuarios que utilicen esta guía deben estar familiarizados con las siguientes tecnologías:

- Conceptos generales de redes
- Aspectos básicos de seguridad relativos a la autenticación y a la autorización
- El lenguaje de programación de Java
- Los componentes de Java 2 Platform, Standard Edition (plataforma J2SE™) y de Java 2 Platform, Enterprise Edition (plataforma J2EE™).

## Organización de esta guía

La guía se organiza en los capítulos siguientes:

- El [Capítulo 1](#) proporciona una introducción a Java ES y presenta las tareas relacionadas con el uso del sistema.
- El [Capítulo 2](#) describe una estructura para el diseño de arquitecturas de soluciones de Java ES y proporciona una arquitectura de ejemplo basada en esa estructura.
- El [Capítulo 3](#) proporciona información sobre las funciones que desempeñan un papel fundamental en la integración de los componentes de Java ES en un único sistema de software.

- El **Capítulo 4** describe la terminología y los conceptos relacionados con cada una de las etapas del ciclo de vida de las soluciones de Java ES.
- El **Apéndice A** proporciona una lista de los componentes de Java ES.

## Conjunto de documentación de Java Enterprise System

El conjunto de documentación del sistema Java ES describe la planificación de la implementación y la instalación del sistema. La dirección URL para la documentación del sistema es <http://docs.sun.com/coll/1286.2>. Para ver una introducción a Java ES, consulte los manuales en el orden en que figuran en la siguiente tabla. También puede encontrar información y recursos de Java ES en <http://www.sun.com/bigadmin/hubs/javaes/>.

**TABLA P-1** Documentación de Java Enterprise System

Título del documento	Contenido
<i>Notas de la versión Sun Java Enterprise System 5 para UNIX</i>	Contiene la información más actualizada acerca de Java ES, incluidos los problemas conocidos. Además, los componentes tienen sus propias notas de la versión que se muestran en la colección de notas de la versión ( <a href="http://docs.sun.com/coll/1315.2">http://docs.sun.com/coll/1315.2</a> ).
<i>Sun Java Enterprise System 5 Release Notes for Microsoft Windows</i>	
<i>Descripción general técnica de Sun Java Enterprise System 5</i>	Presenta los fundamentos técnicos y conceptuales de Java ES. Describe los componentes, la arquitectura, los procesos y las funciones.
<i>Sun Java Enterprise System Deployment Planning Guide</i>	Proporciona una introducción a la planificación y el diseño de soluciones de implementación empresarial basadas en Java ES. Presenta conceptos básicos y principios relacionados con el diseño y la planificación de implementaciones, muestra el ciclo de vida de la solución y proporciona ejemplos generales y estrategias que se pueden usar a la hora de planificar soluciones basadas en Java ES.
<i>Guía de planificación de la instalación de Sun Java Enterprise System 5</i>	Le ayuda a desarrollar las especificaciones de implementación para el hardware, el sistema operativo y la red de su implementación de Java ES. Describe problemas a los que deberá hacer frente al planificar la configuración y la instalación como, por ejemplo, las dependencias existentes entre los componentes.
<i>Guía de instalación de Sun Java Enterprise System 5 para UNIX</i>	Le guía en el proceso de instalación de Java ES. También muestra cómo se configuran los componentes después de instalarlos y cómo se comprueba si funcionan correctamente.
<i>Sun Java Enterprise System 5 Installation Guide for Microsoft Windows</i>	

**TABLA P-1** Documentación de Java Enterprise System (Continuación)

Título del documento	Contenido
<i>Guía de referencia de instalación de Sun Java Enterprise System para UNIX</i>	Ofrece información complementaria sobre los parámetros de configuración, proporciona hojas de cálculo para utilizarlas en la planificación de la configuración e incluye material de referencia de, por ejemplo, los directorios y los números de puerto predeterminados del sistema operativo Solaris™ (SO Solaris) y del entorno operativo Linux.
<i>Guía de actualización de Sun Java Enterprise System 5 para UNIX</i>	Proporciona instrucciones para realizar actualizaciones de versiones instaladas anteriormente a Java ES 5.
<i>Sun Java Enterprise System 5 Upgrade Guide for Microsoft Windows</i>	
<i>Guía de supervisión de Sun Java Enterprise System 5</i>	Proporciona instrucciones para configurar la estructura de supervisión para cada uno de los componentes de productos y utilizar Monitoring Console para ver datos en tiempo real y crear reglas de supervisión.
<i>Sun Java Enterprise System Glossary</i>	Define los términos usados en la documentación de Java ES.

## Convenciones tipográficas

La tabla siguiente describe las convenciones tipográficas utilizadas en este manual.

**TABLA P-2** Convenciones tipográficas

Tipos de letra	Significado	Ejemplo
AaBbCc123	Los nombres de los comandos, los archivos, los directorios y los resultados que el equipo muestra en pantalla.	Edite el archivo <code>.login</code> . Utilice <code>ls -a</code> para mostrar todos los archivos. <code>nombre_equipo% tiene correo.</code>
<b>AaBbCc123</b>	Lo que escribe el usuario, en contraste con lo que el equipo muestra en pantalla.	<code>nombre_equipo% su</code> Contraseña:
<i>AaBbCc123</i>	Un marcador de posición que se debe sustituir por un valor o un nombre real.	El comando para eliminar un archivo es <code>rm filename</code> .
<i>AaBbCc123</i>	Títulos de manuales, términos nuevos, y términos que se deben destacar (tenga en cuenta que algunos elementos destacados aparecerán en negrita en línea)	Consulte el capítulo 6 de la <i>Guía del usuario</i> . Un elemento <i>almacenado en caché</i> es una copia que se almacena localmente. <i>No</i> guarde el archivo.

## Documentación, asistencia y formación

El sitio web de Sun proporciona información acerca de los siguientes recursos adicionales:

- Documentación (<http://www.sun.com/documentation/>)
- Asistencia técnica (<http://www.sun.com/support/>)
- Formación (<http://www.sun.com/training/>)

## Búsqueda de documentación de productos de Sun

Además de buscar documentación de productos de Sun en el sitio web [docs.sun.com](http://docs.sun.com)<sup>SM</sup>, puede utilizar un motor de búsqueda escribiendo la siguiente sintaxis en el campo de búsqueda:

```
search-term site:docs.sun.com
```

Por ejemplo, para buscar “broker,” escriba lo siguiente:

```
broker site:docs.sun.com
```

Para incluir otros sitios web de Sun en la búsqueda (por ejemplo, [java.sun.com](http://java.sun.com), [www.sun.com](http://www.sun.com) y [developers.sun.com](http://developers.sun.com)), utilice `sun.com` en lugar de `docs.sun.com` en el campo de búsqueda.

## Referencias a sitios web de terceros relacionados

Se hace referencia a las direcciones URL de terceras partes para proporcionar información adicional relacionada.

---

**Nota** – Sun no se hace responsable de la disponibilidad de los sitios Web de terceras partes que se mencionan en este documento. Sun no garantiza ni se hace responsable de ningún contenido, anuncio, producto ni de cualquier otro material disponible en dichos sitios o recursos. Sun no se responsabiliza de ningún daño, real o supuesto, ni de posibles pérdidas que se pudieran derivar del uso de los contenidos, bienes o servicios que estén disponibles en dichos sitios o recursos.

---

## Sun valora sus comentarios

Sun tiene interés en mejorar su documentación y valora sus comentarios y sugerencias. Si desea compartir algún comentario, vaya a <http://docs.sun.com> y haga clic en Enviar comentarios (Send Comments). Se mostrará un formulario en línea en el que deberá indicar el título completo del documento y el número de referencia. El número de referencia consta de 7 o 9 dígitos y se puede encontrar en la página de título del manual o en la dirección URL del documento. Por ejemplo, el número de referencia de este manual es 820-0167-10.

# Introducción a Java Enterprise System

---

Sun Java™ Enterprise System (Java ES) es un conjunto de componentes de software que proporciona los servicios necesarios para poder utilizar potentes aplicaciones de empresa distribuidas en un entorno de red o de Internet. Estas aplicaciones reciben el nombre de aplicaciones de empresas distribuidas. Esta guía se centra en los componentes de software de Java ES y los servicios que proporcionan.

En este capítulo se ofrece una introducción a Java Enterprise System y se describen las tareas que conlleva la utilización del sistema. Este capítulo contiene las siguientes secciones:

- “¿Por qué necesita Java ES?” en la página 15
- “Componentes de Java ES” en la página 17
- “Componentes de los conjuntos de Sun Java” en la página 22
- “Cómo trabajar con Java ES” en la página 24
- “Términos clave de este capítulo” en la página 29

## ¿Por qué necesita Java ES?

En la actualidad, los requisitos empresariales demandan soluciones de software distribuidas en la red o en entornos de Internet y que tengan altos niveles de rendimiento, disponibilidad, seguridad, escalabilidad y facilidad de mantenimiento.

Java ES proporciona los servicios de infraestructura necesarios para estas [aplicaciones de empresa distribuidas](#) que, por lo general, presentan las siguientes características:

- **Distribuida.** La aplicación consiste en [componentes](#) de software que interactúan implementados en un entorno de red que puede incluir sitios remotos geográficamente. Estos componentes distribuidos que se ejecutan en varios equipos del entorno funcionan juntos para proporcionar funciones de negocio específicas para los [usuarios finales](#) y otras aplicaciones de negocio.

- **Potencia empresarial.** El ámbito de la aplicación y la escala deben satisfacer las necesidades de un entorno de producción o de un proveedor de servicios de Internet. La aplicación normalmente abarca toda la empresa e integra varios departamentos, operaciones y procesos en un único sistema de software. La aplicación debe cumplir requisitos de alta calidad de servicio relativos al rendimiento, la disponibilidad, la seguridad, la escalabilidad y la facilidad de mantenimiento.

Las aplicaciones de empresa distribuidas requieren un conjunto subyacente de [servicios](#) de infraestructura que permita a los componentes distribuidos comunicarse entre ellos, coordinar el trabajo, implementar un acceso seguro, etc. Estos servicios de infraestructura son, a su vez, compatibles con un entorno de hardware de equipos y vínculos de red. Este entorno de hardware incluye arquitecturas de hardware SPARC® y x86 (Intel y AMD).

En la siguiente figura se muestra el esquema general. En su mayor parte, Java ES proporciona la capa de servicios de infraestructura distribuidos que se muestra en la figura.

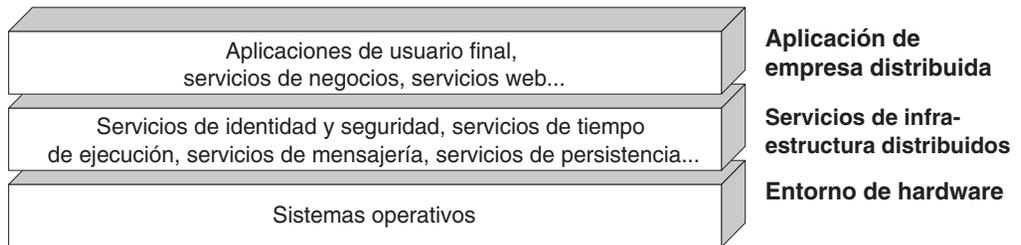


FIGURA 1-1 Compatibilidad necesaria para aplicaciones de empresa distribuidas

Entre los servicios que proporciona Java ES, se incluyen los siguientes:

- **Servicios de portal.** Estos servicios permiten a los empleados, los teletrabajadores, las personas que trabajan con la información, los socios empresariales, los proveedores y los clientes acceder a recursos corporativos desde cualquier lugar fuera de la red corporativa. Estos servicios proporcionan funciones de acceso en cualquier lugar y en cualquier momento a las comunidades de usuarios, al tiempo que aportan características de integración, adición, seguridad, acceso móvil y búsqueda personalizadas.
- **Servicios de comunicación y colaboración.** Estos servicios permiten el intercambio seguro de información entre varias comunidades de usuarios. Entre las funciones específicas, se incluyen las siguientes: mensajería, colaboración en tiempo real (como mensajería y conferencia) y programación de calendario en el contexto del entorno de negocios del usuario.

---

**Nota** – En esta guía, se hace referencia a los componentes de Sun Java Communications Suite que dependen de los componentes de Java ES y se utilizan en las arquitecturas de implementación de Java ES. Los componentes de comunicación y colaboración no se incluyen en Java ES.

---

- **Identidad de red y servicios de seguridad.** Estos servicios mejoran la seguridad y la protección de los activos clave de información corporativa, garantizando la aplicación de directivas de control de acceso adecuadas en todas las comunidades, aplicaciones y servicios de forma global. Estos servicios funcionan con un depósito para almacenar y administrar perfiles de identidad, privilegios de acceso e información de recursos de aplicaciones y de redes.
- **Servicios de aplicación y contenedor web.** Estos servicios permiten que los componentes distribuidos se comuniquen entre sí en el tiempo de ejecución y que sean compatibles con el desarrollo, la implementación y la administración de aplicaciones para una amplia gama de servidores, clientes y dispositivos. Estos servicios se basan en la tecnología de J2EE.

Java ES también proporciona servicios que mejoran la disponibilidad, la escalabilidad, la facilidad de mantenimiento y otras cualidades de las aplicaciones o del sistema. Entre las funciones de calidad del servicio proporcionadas por Java ES, se incluyen las siguientes:

- **Servicios de disponibilidad.** Estos servicios proporcionan una disponibilidad prácticamente continua, también para los componentes de aplicación y para los componentes de infraestructura que los respaldan.
- **Servicios de acceso.** Estos servicios proporcionan acceso por Internet o basado en navegador a los servicios de Java ES.
- **Servicios de supervisión.** Estos servicios proporcionan información en tiempo real acerca de los componentes de Java ES.

Puede implementar uno o varios servicios de Java ES; cada uno de ellos puede incluir varios componentes de Java ES.

## Componentes de Java ES

Java ES es una integración de distintos productos de software y componentes en un único sistema de software. Una serie de funciones en el nivel del sistema facilitan esta integración, entre las que se incluyen:

- Todos los componentes están sincronizados en un conjunto común de bibliotecas compartidas.
- Todos los componentes de Java ES se instalan usando un único instalador.
- Todos los componentes de Java ES pueden compartir una identidad de usuario integrada y un sistema de administración de la seguridad.

- Todos los componentes de Java ES tienen una estructura de supervisión común.

Estas funciones se describen en los siguientes capítulos de esta guía. El objetivo de esta sección es presentar los componentes integrados en Java ES. Estos [componentes del sistema](#) se pueden agrupar en tres categorías principales, tal y como se muestra en la siguiente figura:

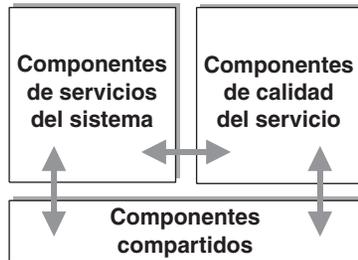


FIGURA 1-2 Categorías de los componentes de Java ES

Los componentes proporcionan los siguientes servicios, como se indica a continuación:

- **Componentes de servicios del sistema.** Estos componentes proporcionan los servicios de infraestructura principales de Java ES que admiten el uso de aplicaciones de empresa distribuidas.
- **Componentes de calidad del servicio.** Estas calidades mejoran la disponibilidad, la seguridad, la escalabilidad y la facilidad de mantenimiento de los componentes de los servicios del sistema y los componentes de aplicaciones distribuidas.
- **Componentes compartidos.** Estos componentes proporcionan el entorno en el que se ejecutan los componentes de los servicios del sistema y de calidad del servicio.

Para obtener una lista de los componentes de Java ES, consulte el [Apéndice A](#).

## Componentes de servicios del sistema

Una serie de componentes de Java ES proporcionan los servicios principales que dan respaldo a las soluciones de software distribuidas. Entre estos [servicios del sistema](#) se incluyen los servicios de portal, identidad y seguridad, contenedor web, aplicaciones de J2EE y persistencia.

Los [componentes de servicios del sistema](#) que proporcionan estos servicios distribuidos y los servicios que éstos proporcionan se muestran en orden alfabético y se describen brevemente en la siguiente tabla. Cada componente de servicios del sistema es un proceso de servidor con varios subprocesos que admite el uso de una gran cantidad de clientes. Para obtener más información sobre cada componente, consulte “[Componentes de servicios del sistema](#)” en la [página 69](#).

TABLA 1-1 Componentes de los servicios del sistema de Java ES

Componente	Servicios del sistema proporcionados
Sun Java System Access Manager	Incluye servicios de administración de acceso y de administración de identidades digitales. Los servicios de administración de acceso incluyen la autenticación (también el inicio de sesión único) y la autorización basada en funciones para acceder a las aplicaciones y los servicios. Estos servicios incluyen la administración centralizada de cuentas de usuario individuales, roles, grupos y directivas.
Sun Java System Application Server	Incluye servicios de contenedor de J2EE para componentes de Enterprise JavaBeans™ (EJB), como beans de sesión, beans de entidad y beans controlados por mensajes. El contenedor incluye servicios de infraestructura necesarios para que interactúen los componentes distribuidos y bien acoplados, lo que convierte a Application Server en una plataforma para el desarrollo y la ejecución de aplicaciones de comercio electrónico y servicios web. Application Server también proporciona servicios de contenedor web.
Sun Java System Directory Server	Incluye un depósito central para almacenar y administrar información de intranet e Internet como, por ejemplo, perfiles de identidad (empleados, clientes, proveedores, etc.), credenciales de usuario (certificados de clave pública, contraseñas y números de identificación personal), privilegios de acceso, información de recursos de aplicaciones e información de recursos de red.
Java DB <sup>1</sup>	Proporciona una base de datos ligera para el desarrollo de aplicaciones de Java. Java DB es una distribución compatible de Sun de la base de de datos de código abierto Apache Derby, implementada por competo mediante tecnología de Java.
Sun Java System Message Queue	Proporciona funciones de mensajería asíncrona y fiable entre aplicaciones y componentes distribuidos que no estén firmemente acoplados. Message Queue implementa la especificación de la API de Java Message Service (JMS) y agrega funciones empresariales, tales como la seguridad, la escalabilidad y la administración remota.
Sun Java System Portal Server	Proporciona servicios de portal claves, como por ejemplo, personalización y agregación para clientes basados en navegadores que acceden a aplicaciones o servicios de negocios. Portal Server también proporciona un motor de búsquedas configurable.
Sun Java System Service Registry	Proporciona un registro y un depósito para que se puedan usar aplicaciones web con arquitecturas orientadas a servicios (SOA). Service Registry implementa estándares del mercado para registrar y detectar servicios web, así como para administrar los datos y artefactos de información asociados como, por ejemplo, el esquema XML, las reglas de los procesos empresariales, el control de acceso, el control de versión, etc.

<sup>1</sup> Java ES 5 es la primera versión que incluye Java DB como componente de producto. Java DB se introdujo por primera vez como componente compartido en Java ES 2005Q4, con el nombre de Derby Database.

TABLA 1-1 Componentes de los servicios del sistema de Java ES (Continuación)

Componente	Servicios del sistema proporcionados
Sun Java System Web Server	Proporciona servicios de contenedor web de J2EE™ para componentes web de Java como, por ejemplo, los componentes Java Servlet y JavaServer Pages™ (JSP™). Web Server también admite otras tecnologías de aplicación web para proporcionar contenido web dinámico y estático, como, por ejemplo, las secuencias de comandos CGI y Sun Java System Active Server Pages.

## Componentes de calidad del servicio

Además de los componentes de servicios del sistema que se muestran en la [Tabla 1-1](#), Java ES incluye una serie de componentes que se usan para mejorar la calidad de los servicios proporcionados por los componentes de servicios del sistema. Los componentes de calidad del servicio también pueden mejorar los servicios de aplicaciones desarrollados de forma personalizada. Los [componentes de servicio de calidad](#) se pueden clasificar en las siguientes categorías:

- Componentes de disponibilidad
- Componentes de acceso
- Componentes de supervisión

### Componentes de disponibilidad

Los componentes de disponibilidad ofrecen un tiempo de actividad prácticamente continuo para los componentes de los servicios del sistema y los servicios de aplicaciones personalizadas. Los componentes de disponibilidad incluidos en Java ES y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más información sobre cada componente, consulte [“Componentes de disponibilidad” en la página 74](#).

TABLA 1-2 Componentes de disponibilidad de Java ES

Componente	Servicios de disponibilidad proporcionados
High Availability Session Store	Proporciona un almacén de datos que hace que los datos de la aplicación, especialmente los datos del estado de la sesión, estén disponibles incluso en caso de fallo.
Sun Cluster	Proporciona servicios de escalabilidad y de alta disponibilidad para Java ES, las aplicaciones que se ejecutan en la parte superior de la infraestructura de Java ES y el entorno de hardware en el que se implementan.
Sun Cluster Geographic Edition <sup>1</sup>	Protege las aplicaciones de interrupciones inesperadas mediante el uso de varios clústeres separados geográficamente y una infraestructura redundante que repite los datos entre estos clústeres. El software Sun Cluster Geographic Edition es una extensión por capas del software de Sun Cluster.

<sup>1</sup> Java ES 5 es la primera versión que incluye Sun Cluster Geographic Edition como componente de producto de Java ES.

## Componentes de acceso

Los componentes de acceso proporcionan acceso cliente a los servicios del sistema, a menudo un acceso seguro desde ubicaciones de Internet fuera de un servidor de seguridad de la empresa. Además de dicho acceso, también ofrecen una función de enrutamiento y almacenamiento en la caché. Los componentes de acceso incluidos en Java ES y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más información sobre cada componente, consulte [“Componentes de acceso” en la página 76](#).

TABLA 1-3 Componentes de acceso de Java ES

Componente	Servicios de acceso proporcionados
Sun Java System Portal Server (incluye Secure Remote Access)	Proporciona un acceso de Internet seguro desde fuera de un servidor de seguridad corporativo al contenido de Portal Server y los servicios, incluidos los portales internos.
Sun Java System Web Proxy Server	Proporciona funciones de almacenamiento en la caché, filtrado y distribución de contenido web para las solicitudes de Internet entrantes y salientes.

## Componentes de supervisión

Java ES incluye una función de supervisión nueva que proporciona el estado del sistema en tiempo real y realiza trabajos de supervisión que se pueden personalizar. El [componente de producto](#) Sun Java System Monitoring Console, compatible con el [componente compartido](#) Sun Java System Monitoring Framework, implementa la supervisión. Para obtener más información, consulte [“Componentes de supervisión” en la página 77](#).

## Componentes compartidos

Java ES incluye varias bibliotecas compartidas instaladas localmente de las que dependen muchos componentes de servicios del sistema y muchos componentes de calidad del servicio. Los [componentes compartidos](#) de Java ES proporcionan servicios locales para los [componentes de productos](#) de Java ES que se ejecuten en el mismo equipo host.

Los componentes compartidos se utilizan a menudo para proporcionar portabilidad entre los distintos sistemas operativos. Entre los ejemplos de componentes compartidos de Java ES, se incluyen: Java 2 Platform, Standard Edition (J2SE), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Java Security Services for Java (JSS), etc. Para ver la lista completa, consulte [“Componentes compartidos” en la página 77](#).

Los componentes compartidos se instalan automáticamente con el instalador de Java ES, en función de los componentes de los servicios del sistema y de los componentes de calidad del servicio que se instalen.

## Componentes de los conjuntos de Sun Java

Java ES está disponible como una única distribución de software de infraestructura punto a punto y como distribuciones de conjuntos individuales destinadas a satisfacer las necesidades empresariales vitales. Java ES incluye todos los componentes de Java ES, mientras que los conjuntos de Sun Java System incluyen subgrupos de componentes seleccionados para satisfacer determinadas necesidades empresariales. Los programas de instalación y desinstalación de Java ES se incluyen en las distribuciones de conjuntos, pero se han reducido para que sólo administren los componentes del conjunto. Todos los componentes compartidos se incluyen en todas las distribuciones de conjuntos.

El contenido de los conjuntos individuales y de los requisitos empresariales que deben satisfacer se muestra en la siguiente tabla.

**TABLA 1-4** Componentes de los conjuntos de Sun Java

Conjunto	Requisito empresarial	Contenido
Sun Java Application Platform Suite	Desarrollo, implementación y administración de arquitecturas orientadas a servicios (SOA) de nueva generación.	Access Manager Application Server Directory Server HADB Java DB Message Queue Monitoring Console Portal Server (incluye Secure Remote Access y Mobile Access) Service Registry Web Proxy Server Web Server
Sun Java Availability Suite	Recuperación frente a desastres y alta disponibilidad para aplicaciones decisivas en procesos empresariales	Software de Sun Cluster Sun Cluster Agents Sun Cluster Geographic Edition

TABLA 1-4 Componentes de los conjuntos de Sun Java (Continuación)

Conjunto	Requisito empresarial	Contenido
Sun Java Communications Suite <sup>1</sup>	Servicios de mensajería y colaboración fiables y seguros	Access Manager Application Server Calendar Server* Communications Express* Delegated Administrator* Directory Server HADB Instant Messaging* Java DB Message Queue Messaging Server* Monitoring Console Web Proxy Server Web Server
Sun Java Identity Management Suite	Administración de identidades de usuario en infraestructuras informáticas y entornos de aplicación	Access Manager Application Server Directory Server HADB Java DB Message Queue Monitoring Console Web Server

<sup>1</sup> Los componentes con un asterisco (\*) son componentes de comunicaciones que ya no se incluyen en Java ES ni se instalan mediante el programa de instalación de Java ES. Estos componentes forman parte de Sun Java Communications Suite.

TABLA 1-4 Componentes de los conjuntos de Sun Java (Continuación)

Conjunto	Requisito empresarial	Contenido
Sun Java Web Infrastructure Suite	Servicios y aplicaciones web para pequeñas y medianas empresas	Access Manager Application Server Directory Server HADB Java DB Message Queue Monitoring Console Service Registry Web Proxy Server Web Server

## Cómo trabajar con Java ES

La creación de soluciones empresariales basadas en el software de Java ES requiere una serie de tareas estándar. Estas tareas varían en su ámbito y dificultad en función del punto de partida que se use para la adopción de Java ES y de la naturaleza de la solución que desee crear e implementar.

En esta sección, se describen dos aspectos del trabajo con Java ES: el ciclo de vida de las soluciones de Java ES y los distintos escenarios de adopción generalmente implicados.

### Ciclo de vida de las soluciones de Java ES

Las tareas necesarias para crear soluciones empresariales basadas en el software de Java ES se pueden dividir en varias fases, tal y como se muestra en la siguiente figura. La ilustración también muestra la categoría del usuario de Java ES que normalmente realiza las distintas tareas.

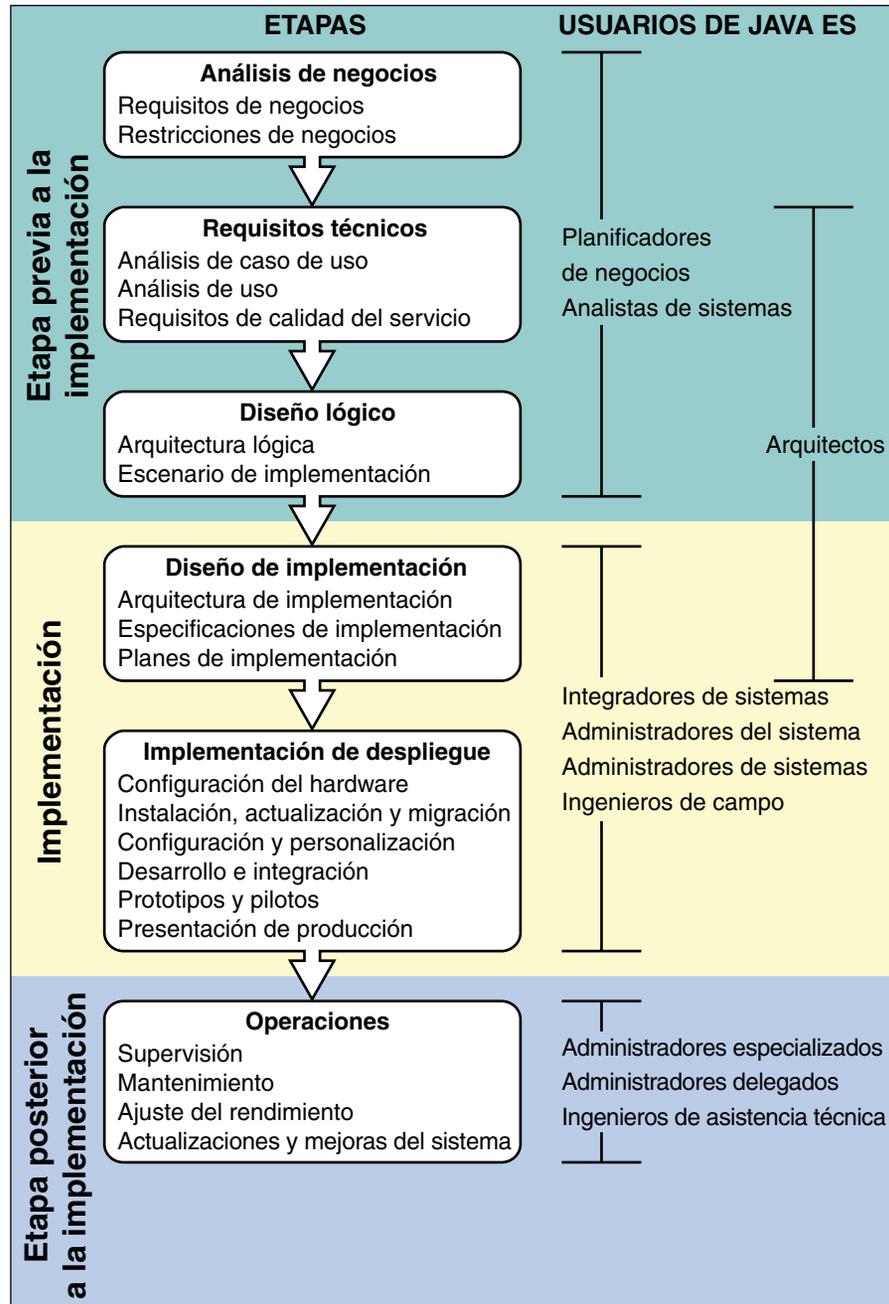


FIGURA 1-3 Etapas del ciclo de vida de las soluciones y categorías de usuarios

Las etapas del ciclo de vida que se muestran en la figura anterior se pueden dividir en los siguientes grupos generales:

- **Etapas previas a la implementación.** En estas etapas, la necesidad empresarial se convierte en un escenario de implementación, es decir, una arquitectura lógica y un conjunto de requisitos de calidad del servicio. El escenario de implementación sirve como especificación para el diseño de la arquitectura de implementación.
- **Implementación.** En estas etapas, el escenario de implementación se convierte en una arquitectura de implementación. Esta arquitectura se puede utilizar como la base para la aprobación y la elaboración del presupuesto del proyecto. La arquitectura de implementación también es la base de la especificación de implementación que proporciona los detalles necesarios para implementar (crear, probar y desplegar) una solución de software en un entorno de producción.
- **Etapas posteriores a la implementación.** En esta fase de operaciones, una solución implementada se ejecuta en condiciones de producción y se supervisa y optimiza para obtener el mejor rendimiento. La solución implementada también se actualiza para que incluya nuevas funciones, según sea necesario.

Las tareas de esta etapa del ciclo de vida se muestran en la [Figura 1-3](#) y se describen detalladamente en el [Capítulo 4](#).

La [Figura 1-3](#) muestra los usuarios de Java ES que realizan normalmente las tareas de las distintas etapas del ciclo de vida. La siguiente tabla describe las capacidades e información de cada categoría de usuario.

**TABLA 1-5** Categorías de usuarios para las tareas del ciclo de vida de Java ES

Usuario	Capacidades y formación	Etapas
Planificador de negocios }Analista de sistemas	Conocimiento general, en lugar de un conocimiento técnico en profundidad Comprende la dirección estratégica de la empresa. Conoce los procesos, objetivos y requisitos empresariales.	Análisis empresarial Requisitos técnicos Diseño lógico
Arquitecto	Conocimiento altamente técnico. Tiene conocimientos amplios sobre arquitecturas de implementación. Familiarizado con las últimas tecnologías. Comprende las limitaciones y los requisitos empresariales.	Requisitos técnicos Diseño lógico Diseño de la implementación

TABLA 1-5 Categorías de usuarios para las tareas del ciclo de vida de Java ES (Continuación)

Usuario	Capacidades y formación	Etapas
Integrador de sistemas	Conocimiento altamente técnico.	Diseño de la implementación
Ingeniero de campo	Está íntimamente familiarizado con los entornos de las tecnologías de la información.	Implementación de despliegue
Administrador de sistemas	Tiene experiencia en la implementación de soluciones de software distribuidas.	
Administrador del sistema	Conoce la arquitectura, los protocolos, los dispositivos y la seguridad de redes. Conoce los lenguajes de secuencias de comandos y de programación.	
Administrador de sistemas especializado	Conocimiento técnico especializado o del producto.	Operaciones
Administrador delegado	Familiarizado con hardware, plataformas, directorios y bases de datos.	
Ingeniero de soporte	Capacidad de supervisión, solución de problemas y actualización de software. Conoce la administración del sistema para las plataformas del sistema operativo.	

## Escenarios de adopción de Java ES

Las necesidades empresariales que impulsan la adopción de Java ES varían significativamente. Sin embargo, los objetivos generales para prácticamente todas las implementaciones de Java ES se ajustan a uno de los siguientes [escenarios de adopción](#):

- **Sistema nuevo.** No existe ningún sistema de software y el usuario implementa Java ES para que se pueda usar una nueva solución empresarial.
- **Mejora.** Se parte de una infraestructura de TI existente en la que se reemplazan una, varias o todas las partes del sistema por el software Java ES. Normalmente se sustituyen los sistemas o subsistemas porque son demasiado complicados, presentan demasiadas limitaciones o son demasiado costosos de mantener. Por ejemplo, puede requerir una mayor seguridad, una mayor disponibilidad, una mayor escalabilidad, más flexibilidad, menor complejidad, capacidades adicionales (como el inicio de sesión único) o un mejor uso de los recursos de TI.
- **Ampliación.** Se parte de una infraestructura de TI existente en la que se implementa el software de Java ES que hasta ahora no formaba parte del sistema. Normalmente, se amplía un sistema de software de esta manera porque es necesario satisfacer nuevas necesidades empresariales. Es posible que se necesiten nuevas capacidades funcionales, como una adición personalizada de los servicios existentes mediante un portal de Java ES o funciones de autenticación y autorización de Java para los servicios existentes.

- **Actualización.** Se parte de una infraestructura de TI que cuenta con una versión anterior de Java ES o con productos de Sun de versiones anteriores a las de Java ES. El usuario actualiza a la versión más actual de los componentes de Java ES.

Cada escenario de adopción cuenta con sus propios retos y consideraciones. Sin embargo, en función del escenario de adopción, es posible que varíen los problemas que deberá solucionar y los recursos en los que necesitará invertir en las distintas etapas del ciclo de vida mostradas en la [Figura 1-3](#).

Las siguientes consideraciones se aplican en diversos grados a los escenarios de adopción:

- **Migración.** La mejora o la actualización de una infraestructura existente con un software nuevo requiere la migración de los datos desde el sistema existente hasta el sistema nuevo. Los datos pueden ser información de configuración, de usuario o de aplicación. También tendrá que migrar la lógica empresarial o de presentación debido a las interfaces de programación nuevas.
- **Integración.** La adición de nuevo software a un sistema existente o la sustitución de subsistemas de software requieren a menudo la integración de los nuevos componentes de software con los subsistemas restantes. La integración puede implicar el desarrollo de nuevas capas de interfaces utilizando conectores J2EE o adaptadores de recursos, la reconfiguración de los componentes existentes y la implementación de esquemas de transformación de datos.
- **Formación.** Casi todos los cambios que se realicen en la infraestructura implican cambios en los procedimientos de TI y en los conjuntos de habilidades. El departamento de TI debe tener el tiempo suficiente para adquirir los nuevos conocimientos o transferir los antiguos para poder utilizar las nuevas tecnologías de Java ES.
- **Hardware.** Al sustituir o mejorar un sistema o subsistema existentes, las restricciones del negocio pueden exigir que se reutilice hardware existente. En función del escenario de adopción, los recursos de hardware pueden convertirse en un factor importante.

La siguiente tabla resume la naturaleza de las consideraciones que se aplican a cada uno de los escenarios de adopción de Java ES.

TABLA 1-6 Consideraciones sobre los escenarios de adopción de Java ES

Escenario de adopción	Migración	Integración	Formación	Hardware
Sistema nuevo	No hay consideraciones	Es relativamente fácil integrar nuevos componentes	Puede ser una preocupación importante	Concesiones entre costes de equipos y de mano de obra <sup>1</sup>
Mejora	Puede ser una preocupación grave	Es necesario integrar nuevos componentes con el sistema existente	Puede ser una preocupación importante	Puede implicar limitaciones importantes debido al equipo existente

<sup>1</sup> La utilización de unos pocos equipos potentes generalmente aumenta los costes de los equipos a la vez que requiere menos recursos de TI. La utilización de varios equipos más pequeños reduce los costes de equipos a la vez que se necesitan más recursos de TI.

TABLA 1-6 Consideraciones sobre los escenarios de adopción de Java ES (Continuación)

Escenario de adopción	Migración	Integración	Formación	Hardware
Ampliación	Normalmente no es una preocupación	Es posible que sea necesario integrar nuevos componentes con el sistema existente	Puede ser una preocupación importante	Generalmente requiere nuevo hardware con las mismas concesiones que con un nuevo sistema
Actualización	Puede ser una preocupación importante	Es relativamente fácil integrar componentes actualizados	Una preocupación menor	Una preocupación menor

## Términos clave de este capítulo

En esta sección, se explican los términos técnicos clave empleados en este capítulo, con especial atención al uso de estos términos en el contexto de Java ES.

- escenario de adopción** Un motivo general para implementar el software de Java ES, que describe el sistema de software desde el que se parte y el objetivo que se desea conseguir. Existen cuatro escenarios de adopción básicos de Java ES: sistema nuevo, sustitución, ampliación y actualización.
- componente** Unidades lógicas de software a partir de las cuales se crean las aplicaciones distribuidas. Un componente puede ser uno de los [componentes del sistema](#) incluidos en Java ES o también un [componente de aplicación](#) que se desarrolle de forma personalizada. Normalmente, un componente de aplicación se ajusta a un modelo de componente distribuido (como CORBA o la plataforma J2EE) y desempeña una función informática específica. Estos componentes, ya sea de forma individual o combinados, proporcionan [servicios de negocio](#) y se pueden encapsular como [servicios web](#).
- aplicación de empresa distribuida** Una aplicación cuya lógica abarca varios entornos de red o de Internet (el aspecto distribuido), y cuyo ámbito y escala responden a las necesidades de un entorno de producción o de un proveedor de servicios (el aspecto empresarial).
- usuario final** Una persona que usa una aplicación distribuida, a menudo, a través de una interfaz gráfica de usuario como, por ejemplo, un navegador de Internet o una GUI de un dispositivo móvil. El número de usuarios finales simultáneos que admite una aplicación es un factor importante de la [arquitectura de implementación](#) de la aplicación.
- servicio** Una función de software realizada para uno o varios [clientes](#). Esta función podría ser de muy bajo nivel, como la administración de memoria, o de alto nivel, como el [servicio de negocios](#) de comprobación de crédito. Un servicio de alto nivel puede estar formado por una familia de servicios individuales. Los servicios pueden ser locales (disponibles para clientes locales) o distribuidos (disponibles para clientes remotos).

<b>componente de producto</b>	Los <b>componentes de servicios del sistema</b> de Java ES, que proporcionan los principales servicios de infraestructura de Java ES y los <b>componentes de calidad del servicio</b> de Java ES, que mejoran los servicios del sistema. Los componentes de productos se pueden seleccionar en el programa de instalación de Java ES.
<b>componente de calidad del servicio</b>	Un tipo de <b>componente del sistema</b> incluido en Java ES. Estos componentes mejoran la disponibilidad, seguridad, escalabilidad y facilidad de mantenimiento de los componentes de servicios del sistema y los componentes de aplicaciones distribuidas.
<b>componente compartido</b>	Un tipo de <b>componente del sistema</b> incluido en Java ES. Los componentes compartidos, normalmente bibliotecas, proporcionan servicios locales a otros componentes del sistema.
<b>componente del sistema</b>	Cualquier paquete de software o conjunto de paquetes incluidos en Java ES e instalados mediante el programa de instalación de Java ES. Existen varios tipos de componentes del sistema: los <b>componentes de productos</b> que proporcionan los servicios de infraestructura de Java ES y los <b>componentes compartidos</b> que proporcionan servicios locales a los demás componentes del sistema.
<b>servicio del sistema</b>	Uno o varios <b>servicios</b> distribuidos que definen la función exclusiva que proporciona Java ES. Los servicios del sistema normalmente requieren la compatibilidad de una serie de <b>componentes de calidad del servicio</b> , de <b>componentes compartidos</b> o puede que requieran ambos.
<b>componente de servicios del sistema</b>	Un tipo de <b>componente del sistema</b> incluido en Java ES. Los componentes de servicios del sistema proporcionan los servicios de infraestructura de Java ES: servicios de portal, identidad y seguridad, web y de aplicación, y de disponibilidad.

## Arquitecturas de soluciones de Java ES

---

En este capítulo, se proporciona una descripción general de los conceptos arquitectónicos en los que se basan las soluciones de Java ES. En este capítulo, se muestra cómo se utilizan los componentes de servicios del sistema y los componentes de calidad del servicio para poder permitir el uso de soluciones de empresa distribuidas.

Las [arquitecturas](#) de las soluciones de Java ES tienen dos aspectos: una [arquitectura lógica](#) y una [arquitectura de implementación](#). La arquitectura lógica describe las interacciones entre los bloques de construcción lógica (los componentes de software) de una solución. La arquitectura de implementación establece la asignación existente entre la arquitectura lógica y un entorno informático físico. Los componentes de Java ES desempeñan papeles importantes tanto en la arquitectura lógica como en la de implementación.

También se describe una estructura arquitectónica para diseñar arquitecturas de soluciones de Java ES y se muestra una arquitectura de ejemplo basada en dicha estructura. Este capítulo contiene las siguientes secciones:

- “Estructura arquitectónica de Java ES” en la página 31
- “Arquitectura de una solución de Java ES de ejemplo” en la página 45
- “Términos clave de este capítulo” en la página 48

### Estructura arquitectónica de Java ES

Los componentes de Java ES admiten la implementación de soluciones de software distribuidas. Para obtener la funcionalidad solicitada en los niveles de rendimiento, disponibilidad, seguridad, escalabilidad y la facilidad de mantenimiento establecidos por los requisitos de la empresa, estas soluciones de software se deben diseñar de forma adecuada.

Hay una serie de dimensiones arquitectónicas relacionadas con el diseño de potentes soluciones empresariales. Estas dimensiones representan perspectivas distintas desde las que se ven las

interacciones de los distintos componentes de software utilizados para crear dichos sistemas. En concreto, el diseño de los sistemas distribuidos implica las siguientes tres dimensiones arquitectónicas:

- **Dependencias de servicio de infraestructura.** Esta dimensión se centra en la función que desempeñan los componentes de servicios del sistema en el uso de soluciones distribuidas (consulte “[Componentes de servicios del sistema](#)” en la página 18).
- **Capas lógicas.** Esta dimensión se centra en la independencia física y lógica de los componentes de soluciones con el fin de implementarlos en una red o en un entorno de Internet.
- **Calidad del servicio.** Esta dimensión se centra en cómo se satisfacen los requisitos de calidad del servicio como, por ejemplo, la disponibilidad, la seguridad, la escalabilidad y la facilidad de mantenimiento, incluida la función que desempeñan los componentes de calidad del servicio (consulte “[Componentes de calidad del servicio](#)” en la página 20).

Estas tres dimensiones de la arquitectura de la solución se muestran en la siguiente figura.

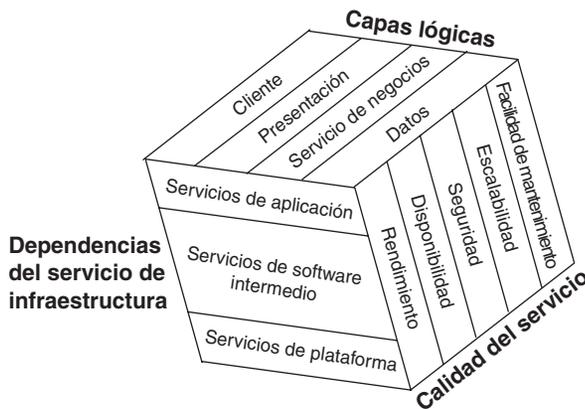


FIGURA 2-1 Dimensiones de la arquitectura de una solución de Java ES

Juntas, estas tres dimensiones representan un único marco que incorpora las relaciones entre los siguientes elementos de software: [componentes de aplicación](#) y componentes de infraestructura; todos ellos son necesarios para obtener las funciones de servicio y la calidad de servicio que se exigen a una solución de software.

Las siguientes secciones describen las tres dimensiones individualmente y, a continuación, figura una síntesis de las tres dimensiones en un marco unificado.

## Dimensión 1: dependencias de los servicios de infraestructuras

Los componentes del software de interacción de las aplicaciones de empresa distribuidas requieren servicios de infraestructura subyacentes que permitan a los componentes distribuidos comunicarse entre sí, coordinar su trabajo, implementar un acceso seguro, etc. Esta sección explica la función principal desarrollada por una serie de componentes de Java ES al proporcionar estos servicios de infraestructura.

### Niveles de servicio de infraestructura

Al diseñar un sistema de software distribuido, con independencia de que esté formado principalmente por los componentes desarrollados de forma personalizada o por los componentes "de fábrica" de Java ES, éste debe incluir una serie de servicios de infraestructura. Estos servicios operan en varios niveles.

La dimensión de las dependencias de los servicios de infraestructura se muestra en la [Figura 2-2](#). Los niveles que se muestran en esta figura son una vista ampliada de la capa de los servicios de infraestructura de la [Figura 1-1](#). La jerarquía de los servicios de la [Figura 2-2](#) y las dependencias existentes entre ellos constituyen una importante dimensión de la arquitectura lógica de la solución. Estos servicios de infraestructura proporcionan los principales motivos para los componentes de servicios del sistema de Java ES (consulte "[Componentes de servicios del sistema](#)" en la [página 18](#)).

En general, los servicios mostrados en la siguiente figura se dividen en tres amplios grupos: servicios de plataforma de nivel inferior, servicios de aplicación de nivel superior y un grupo de servicios de nivel intermedio. Los respectivos nombres proceden de su ubicación entre los otros dos grupos.

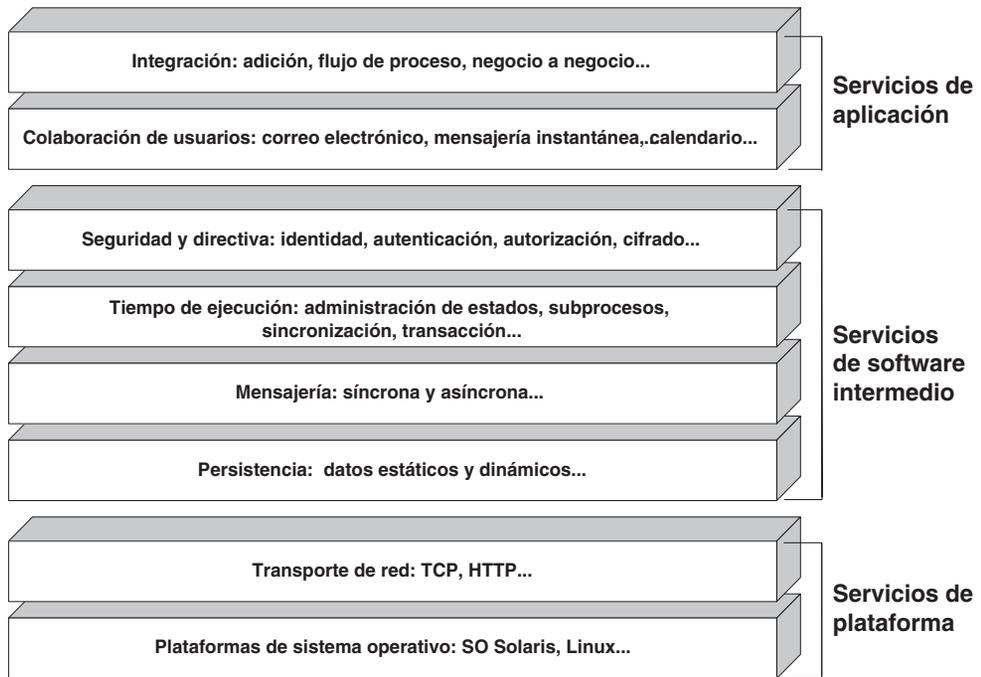


FIGURA 2-2 Dimensión 1: niveles de los servicios de infraestructuras

Las siguientes descripciones de los diferentes niveles de servicio de infraestructura hacen referencia a los artefactos del lenguaje de programación de Java, según sea pertinente. La [Figura 2-2](#) muestra los niveles en orden ascendente, de menor a mayor:

- **Plataforma de sistema operativo.** Proporciona la compatibilidad básica para cualquier proceso que se ejecute en un equipo. El sistema operativo administra los dispositivos físicos, así como la memoria, los subprocesos y otros recursos necesarios para poder utilizar la Máquina virtual de Java (JVM™).
- **Transporte de red.** Proporciona la compatibilidad de red necesaria para las comunicaciones entre los componentes de aplicación distribuidos que se ejecutan en distintos equipos. Estos servicios son compatibles con protocolos como TCP y HTTP. Otros protocolos de comunicación de nivel superior (consulte el nivel de mensajería) dependen de estos servicios de transporte básicos.
- **Persistencia.** Proporciona la compatibilidad necesaria para almacenar datos estáticos (información sobre el usuario, el directorio o la configuración) y datos de aplicación dinámica (información que se actualiza con frecuencia). También permite acceder a estos dos tipos de datos.
- **Mensajería.** Proporciona compatibilidad con las comunicaciones tanto síncronas como asíncronas entre los componentes de la aplicación. La mensajería síncrona es el envío y la recepción de mensajes en tiempo real; incluye llamadas a métodos remotos (RMI) entre

componentes de J2EE e interacciones de SOAP con servicios web. La mensajería asíncrona es la comunicación por la cual el envío de un mensaje no depende de la disponibilidad del consumidor para recibirlo inmediatamente. Las especificaciones de mensajería asíncrona, por ejemplo, Java Message Service (JMS) y ebXML, admiten la fiabilidad garantizada y otras semánticas de mensajería.

- **Tiempo de ejecución.** Proporciona la compatibilidad necesaria para cualquier modelo de componente distribuido, como los modelos J2EE o CORBA. Además de la invocación de métodos remotos necesaria para componentes distribuidos y bien acoplados, los servicios de tiempo de ejecución incluyen administración de estado de componentes (ciclo de vida), administración de grupos de subprocesos, sincronización (bloqueo mutuo), servicios de persistencia, supervisión de transacciones distribuidas y gestión de excepciones distribuidas. En un entorno de J2EE, estos servicios de tiempo de ejecución los ofrecen los contenedores web, de EJB y beans controlados por mensajes en un servidor de aplicaciones o en un servidor web.
- **Seguridad y directiva.** Proporciona compatibilidad con el acceso seguro a los recursos de la aplicación. Estos servicios incluyen compatibilidad con las directivas que controlan el acceso basado en roles o en grupos a los recursos distribuidos, así como las funciones de [inicio de sesión único](#). El inicio de sesión único permite que la autenticación de un usuario en un servicio en un sistema distribuido se aplique automáticamente a otros servicios (componentes de J2EE, servicios de negocios y servicios web) del sistema.
- **Colaboración de usuarios.** Proporciona servicios que desempeñan un papel importante al permitir la comunicación directa entre usuarios y la colaboración entre usuarios en entornos de Internet y empresariales. Son servicios de negocios de nivel de aplicación proporcionados normalmente por servidores independientes (como un servidor de correo electrónico o un servidor de calendario).
- **Integración.** Proporciona los servicios que agregan servicios de negocios existentes. Ofrece una interfaz común para acceder a los servicios, como en un portal, o integrando los servicios mediante un motor de procesos que los coordina en el flujo de trabajo de producción. La integración también puede producirse como interacciones de negocio a negocio entre varias empresas.

Los niveles de servicio que se muestran en la [Figura 2-2](#) reflejan una dependencia entre los servicios de infraestructura, desde los servicios del sistema operativo de nivel inferior hasta los servicios de aplicaciones de nivel superior y los servicios de integración. Cada servicio depende normalmente de servicios de niveles inferiores y respalda el funcionamiento de servicios superiores. La [Figura 2-2](#), sin embargo, no representa una distribución en capas estricta de los servicios de infraestructura. Los servicios de nivel superior pueden interactuar directamente con servicios de nivel inferior sin depender de niveles intermedios. Por ejemplo, algunos servicios de tiempo de ejecución pueden depender directamente de servicios de plataforma sin necesitar ninguno de los niveles de servicio intermedios. Además, otros niveles de servicio, como los de supervisión o administración, también podrían incluirse en esta ilustración conceptual.

## Componentes de los servicios de infraestructura de Java ES

Los componentes de Java ES despliegan los niveles de los servicios de infraestructura distribuidos que se muestran en la [Figura 2-2](#). La posición de los componentes de servicios del sistema en los diferentes niveles se muestra en la siguiente figura.

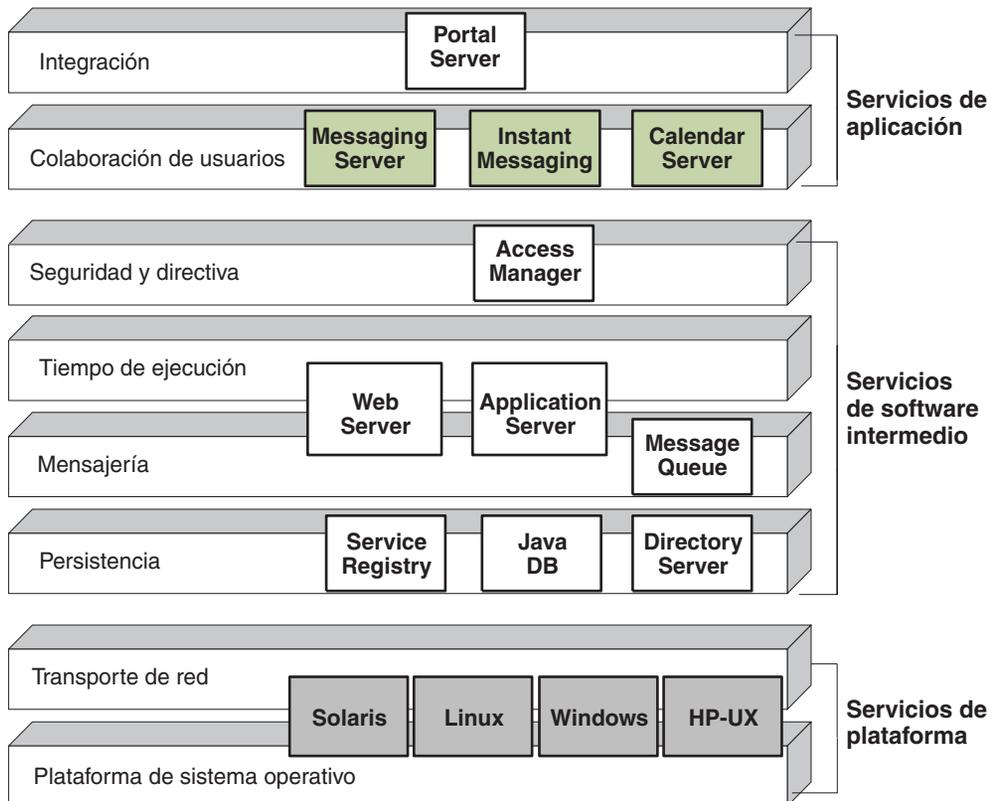


FIGURA 2-3 Componentes de los servicios del sistema de Java ES

**Nota** – Los cuadros sombreados de la figura hacen referencia a componentes no incluidos en Java ES. Los componentes de colaboración de usuarios no forman parte de Java ES, aunque se implementen a menudo junto con los componentes de Java ES y se utilicen en las arquitecturas de Java ES. Estos componentes forman parte de Sun Java Communications Suite y se hace referencia a ellos en este documento sólo con fines ilustrativos. Además, las plataformas de sistemas operativos no forman parte formalmente de Java ES, pero se incluyen para mostrar las plataformas en las que se pueden utilizar los componentes de Java ES.

## Dependencias de los servicios de infraestructura de Java ES

En general, los componentes de los servicios del sistema de Java ES que se muestran en la [Figura 2-3](#) dependen de los componentes situados debajo de ellos en la infraestructura, a la vez que proporcionan respaldo a los componentes que están situados encima de ellos. Estas relaciones de dependencia y compatibilidad son un factor clave para diseñar las arquitecturas lógicas.

En la siguiente tabla, se muestran las relaciones específicas entre los componentes de servicios del sistema de Java ES, enumerados desde el nivel superior al inferior, tal y como se muestra en la [Figura 2-3](#).

**TABLA 2-1** Relaciones entre los componentes de los servicios del sistema de Java ES

Componente	Depende de	Proporciona compatibilidad a
Portal Server	Application Server o Web Server Access Manager Directory Server Si se configuran para usar los canales correspondientes: Calendar Server, Messaging Server , e Instant Messaging <sup>1</sup>	Ninguno
Access Manager	Application Server o Web Server Directory Server	Portal Server Si se configura para el inicio de sesión único: Calendar Server, Messaging Server , y Instant Messaging
Application Server	Message Queue Directory Server (para los objetos administrados)	Portal Server Access Manager
Message Queue	Directory Server (para los objetos administrados)	Application Server
Web Server	Access Manager (para el control de acceso)	Portal Server Access Manager
Directory Server	Ninguno	Portal Server Access Manager Calendar Server Messaging Server Instant Messaging

<sup>1</sup> Los componentes de Calendar Server, Messaging Server e Instant Messaging están disponibles como parte de Sun Java Communications Suite.

TABLA 2-1 Relaciones entre los componentes de los servicios del sistema de Java ES (Continuación)

Componente	Depende de	Proporciona compatibilidad a
Service Registry	Java DB	Componentes basados en Application Server
Java DB	Ninguno	Service Registry

## Dimensión 2: capas lógicas

Los componentes de software de las aplicaciones de empresa distribuidas que interactúan, se pueden visualizar como residentes en un número de capas lógicas. Estas capas representan la independencia física y lógica de los componentes de software en función de la naturaleza de los servicios que proporcionan.

La dimensión de las capas lógicas de la arquitectura de la solución se muestra en la siguiente figura.

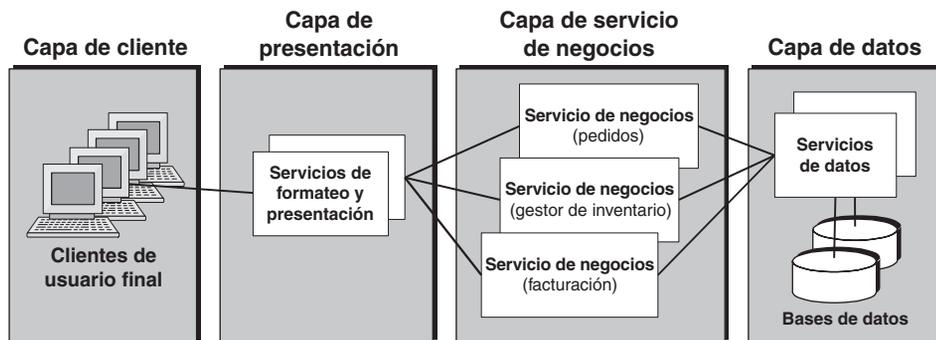


FIGURA 2-4 Dimensión 2: capas lógicas para aplicaciones de empresa distribuidas

En su mayor parte, las arquitecturas de capas lógicas representan la capa de aplicación empresarial distribuida de la [Figura 1-1](#). Los componentes de los servicios del sistema de Java ES se describen en “[Niveles de servicio de infraestructura](#)” en la [página 33](#) para proporcionar asistencia a los componentes de aplicaciones de todas las capas lógicas que se muestran en la [Figura 2-4](#). Aunque los conceptos de las capas lógicas se aplican principalmente a las aplicaciones de empresa personalizadas, también se aplican a los servicios de colaboración proporcionados por los componentes de Sun Java Communications Suite y algunos servicios de portal.

## Descripción de capas lógicas

Esta sección proporciona breves descripciones de las cuatro capas lógicas que se muestran en la [Figura 2-4](#). Las descripciones hacen referencia a los componentes de la aplicación

implementados utilizando el modelo de componente de plataforma de J2EE. No obstante, otros modelos de componente distribuidos, como CORBA, también son compatibles con esta arquitectura.

- **Capa de cliente.** La capa de cliente está formada por la lógica de la aplicación a la que el usuario final accede directamente mediante una interfaz de usuario. La lógica de la capa de cliente podría incluir clientes basados en navegadores, componentes de Java que se ejecuten en un equipo de escritorio o clientes móviles de Java™ Platform, Micro Edition (plataforma J2ME™) que se ejecuten en un dispositivo portátil.
- **Capa de presentación.** La capa de presentación está formada por la lógica de aplicación, que prepara datos para su envío a la capa de cliente y procesa solicitudes desde la capa de cliente para su envío a la lógica de negocios del servidor. La lógica en la capa de presentación está formada normalmente por componentes de J2EE como, por ejemplo, Java Servlet o los componentes de JSP que preparan los datos para enviarlos en formato HTML o XML, o que reciben solicitudes para procesarlas. Esta capa también puede incluir un servicio de portal que proporcione acceso personalizado y seguro a los [servicios de negocios](#) en la capa de servicios de negocio.
- **Capa de servicios de negocios.** La capa de servicios de negocio consiste en la lógica que realiza las funciones principales de la aplicación: procesamiento de datos, implementación de funciones de negocios, coordinación de varios usuarios y administración de recursos externos como, por ejemplo, bases de datos o sistemas heredados. Esta capa suele estar formada por componentes firmemente acoplados que se ajustan al modelo de componentes distribuidos de J2EE como, por ejemplo, los objetos Java, los componentes EJB o los beans conducidos mediante mensajes. Pueden montarse componentes de J2EE individuales para ofrecer servicios de negocios complejos, como, por ejemplo, un servicio de inventario o uno de cálculo de impuestos. Los componentes individuales y los ensamblados de servicios se pueden encapsular como [servicios web](#) que no estén firmemente acoplados en un modelo de arquitectura orientada a servicios, que se ajuste a los estándares de la interfaz SOAP (Simple Object Access Protocol). Los servicios de negocios también se pueden crear como [servidores](#) independientes como, por ejemplo, un servidor de mensajería o un servidor de calendario empresarial.
- **Capa de datos.** La capa de datos está formada por los servicios que proporcionan los datos persistentes utilizados por la lógica de negocios. Los datos pueden ser datos de aplicaciones almacenados en un sistema de administración de bases de datos o pueden incluir información de recursos y directorios almacenada en un almacén de datos de protocolo ligero de acceso a directorios (LDAP). Los servicios de datos también pueden incluir alimentación de datos de orígenes externos o datos a los que se puede obtener acceso desde sistemas informáticos heredados.

## Independencia lógica y física

La dimensión arquitectónica que se ilustra en la [Figura 2–4](#) destaca la independencia lógica y física de los componentes, representada mediante 4 capas separadas. Estas capas representan la partición de la lógica de la aplicación en varios equipos en un entorno de red:

- **Independencia lógica.** Las cuatro capas del modelo arquitectónico representan independencia lógica: puede modificar la lógica de la aplicación en una capa (por ejemplo, en la capa de servicio de negocios) independientemente de la lógica de las otras capas. Puede cambiar la implementación de la lógica de negocios sin tener que cambiar o actualizar la lógica de la capa de presentación o la de cliente. Esta independencia significa, por ejemplo, que puede introducir nuevos tipos de componentes de clientes sin tener que modificar los componentes de los servicios de negocios.
- **Independencia física.** Las cuatro capas también representan independencia física: es posible implementar la lógica en capas distintas en varias plataformas de hardware (es decir, varias configuraciones de procesador, conjuntos de chips y sistemas operativos). Esta independencia permite ejecutar componentes de aplicación distribuida en los equipos que mejor se adapten a las necesidades informáticas individuales y a maximizar el ancho de banda de red.

La forma de asignar componentes de aplicación o componentes de infraestructura a un entorno de hardware (es decir, la arquitectura de implementación) depende de muchos factores, en función de la escala y la complejidad de la solución de software. Para implementaciones muy pequeñas, una arquitectura de implementación puede implicar sólo unos pocos equipos. Para las implementaciones a gran escala, la asignación de los componentes en un entorno de hardware puede tener en cuenta factores como la velocidad y potencia de los distintos equipos, la velocidad y el ancho de banda de los enlaces de la red, las consideraciones de seguridad y de servidores de seguridad y las estrategias de duplicación de componentes para obtener escalabilidad y una alta disponibilidad.

## Arquitectura en capas aplicada a los componentes del sistema

Tal y como se muestra en la [Figura 2-3](#), los componentes de los servicios de infraestructura de Java ES proporcionan la infraestructura subyacente que permite utilizar soluciones de software distribuidas. Algunas de estas soluciones proporcionan servicios de nivel de aplicación proporcionados por los componentes de Sun Java Communications Suite y algunos componentes de Java ES. Estas soluciones utilizan enfoques de diseño de capas lógicas.

Por ejemplo, los servicios de comunicación mediante correo electrónico proporcionados por Messaging Server se despliegan usando una serie de configuraciones diferenciadas desde el punto de vista lógico de Messaging Server. Cada una de estas diferentes configuraciones proporciona un conjunto de servicios distinto. Al diseñar soluciones de mensajería, estas distintas configuraciones se representan como componentes independientes ubicados en diferentes capas lógicas, como se muestra en la siguiente figura en la que las líneas que conectan los componentes representan las interacciones.

---

**Nota** – La siguiente figura no se ha diseñado para que represente una arquitectura lógica completa. Se han omitido una serie de componentes de Java ES para simplificar la ilustración.

---

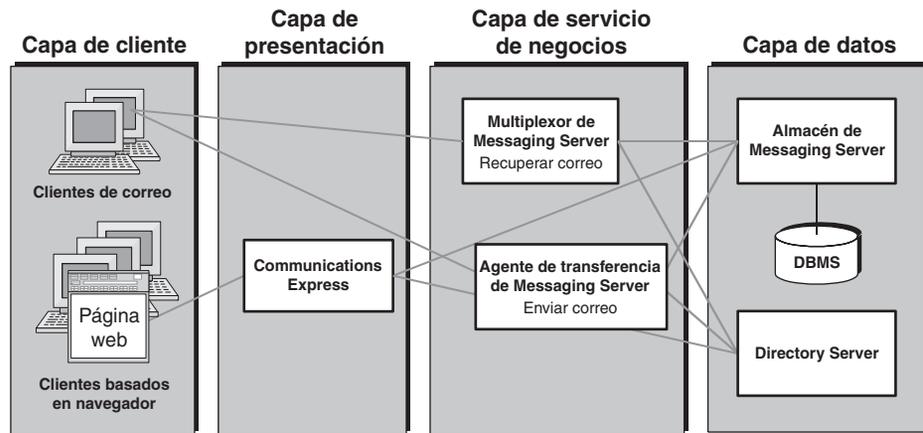


FIGURA 2-5 Messaging Server : ejemplo de arquitectura en capas

**Nota** – Los componentes de comunicaciones no forman parte de Java ES, aunque se implementen a menudo con los componentes de Java y se utilicen en las arquitecturas de Java ES. Estos componentes de comunicaciones forman parte de Sun Java Communications Suite y se hace referencia a ellos en este documento sólo con fines ilustrativos.

La separación lógica de las funciones de Messaging Server en distintas capas permite implementar las configuraciones lógicamente distintas de Messaging Server en varios equipos en un entorno físico. La separación física aporta flexibilidad a la hora de satisfacer los requisitos de calidad del servicio (consulte “[Dimensión 3: calidad del servicio](#)” en la página 41). Por ejemplo, proporciona diversas soluciones de disponibilidad para distintas instancias y distintas implementaciones de seguridad para funciones de Messaging Server diferentes.

## Dimensión 3: calidad del servicio

Las dos dimensiones arquitectónicas anteriores (dependencias de servicio de infraestructura y capas lógicas) hacen referencia en buena parte a los aspectos lógicos de la arquitectura, es decir, qué componentes son necesarios para interactuar de cierto modo con objeto de ofrecer los servicios a los usuarios finales. Otra dimensión igualmente importante de cualquier solución implementada es la capacidad de ésta para cumplir los requisitos de calidad del servicio.

La dimensión de calidad del servicio de una arquitectura de solución destaca las funciones desempeñadas por los componentes de calidad de servicio de Java ES.

## Calidades del servicio

A medida que los servicios de Internet y de comercio electrónico se han hecho más importantes para las operaciones de negocios, el rendimiento, la disponibilidad, la seguridad, la escalabilidad y la facilidad de mantenimiento de estos servicios se han convertido en un requisito fundamental de calidad del servicio para las arquitecturas de implementación de alto rendimiento y de gran escala.

Para diseñar una solución de software satisfactoria, debe determinar los requisitos de calidad del servicio pertinentes y diseñar una arquitectura que satisfaga dichos requisitos. Se utiliza una serie de importantes calidades del servicio para especificar los requisitos de calidad del servicio. Estas calidades de servicios se resumen en la siguiente tabla.

TABLA 2-2 Calidades de servicio que afectan a la arquitectura de solución

Calidades de servicio del sistema	Descripción
Rendimiento	La medición de la latencia y del tiempo de respuesta con relación a las condiciones de carga de usuarios.
Disponibilidad	Medida de la frecuencia con que los usuarios finales acceden a los servicios y recursos de un sistema (el tiempo de <i>actividad de un sistema</i> ).
Seguridad	Combinación compleja de factores que describe la integridad de un sistema y sus usuarios. La seguridad incluye la seguridad física de los sistemas, seguridad de red, seguridad de datos y aplicaciones (autenticación y autorización de usuarios), así como el transporte seguro de la información.
Escalabilidad	La capacidad de agregar a lo largo del tiempo funciones a un sistema implementado. La escalabilidad normalmente implica agregar recursos al sistema, pero no debería requerir cambios en la arquitectura de implementación.
Capacidad latente	La capacidad de un sistema para gestionar el uso de carga máxima inusual sin recursos adicionales.
Facilidad de mantenimiento	La facilidad con que un sistema implementado puede mantenerse, incluidas tareas tales como la supervisión del sistema, la reparación de los problemas que surjan y la actualización de los componentes de hardware y software.

La dimensión de calidad del servicio influye en gran medida en la arquitectura de implementación de una solución: cómo se implementan en el entorno físico los componentes de la aplicación y componentes de infraestructura.

Las calidades del servicio que afectan a la arquitectura de implementación están estrechamente interrelacionadas. Los requisitos para una calidad de sistema afectan a menudo al diseño de otras calidades de servicio. Por ejemplo, unos mayores niveles de seguridad podrían afectar al rendimiento, que a su vez podría afectar a la disponibilidad. La integración de equipos adicionales para solucionar problemas de disponibilidad mediante la redundancia a menudo afecta a los costes de mantenimiento (facilidad de mantenimiento).

Al diseñar arquitecturas de implementación que satisfagan las necesidades y las limitaciones de negocios, es importante conocer el modo de interrelación de las calidades del servicio y las concesiones que se deben realizar.

## Componentes de calidad del servicio de Java ES

Varios componentes de Java ES se utilizan principalmente para mejorar la calidad de los servicios proporcionados por los componentes de los servicios del sistema o por los componentes de aplicaciones distribuidas. A menudo, estos componentes de software se utilizan junto con los componentes de hardware como, por ejemplo, los equilibradores de carga y los servidores de seguridad.

Los componentes de calidad del servicio de Java ES, de los que se realiza una introducción en “Componentes de calidad del servicio” en la página 20, se resumen a continuación:

- **Componentes de disponibilidad.** Proporcionan un tiempo de actividad casi continuo para una solución implementada.
- **Componentes de acceso.** Proporcionan un acceso seguro a los servicios del sistema a través de Internet y, a menudo, también una función de enrutamiento.
- **Componentes de supervisión.** Proporcionan información en tiempo real acerca de los componentes de Java ES.

La siguiente tabla muestra los componentes de calidad del servicio de Java ES más importantes desde una perspectiva arquitectónica, junto con las calidades del sistema a las que más afectan.

TABLA 2-3 Componentes de calidad de servicio y calidades de sistema afectadas

Componente	Calidades de sistema afectadas
almacén de sesión de alta disponibilidad	Disponibilidad
Monitoring Console	Facilidad de mantenimiento
Portal Server Secure Remote Access	Seguridad Escalabilidad
Sun Cluster	Disponibilidad Escalabilidad
Sun Cluster Geographic Edition	Disponibilidad Escalabilidad
Web Proxy Server	Seguridad Rendimiento Escalabilidad

## Software de Sun Cluster

El software de Sun Cluster proporciona servicios de alta disponibilidad y escalabilidad para los componentes de Java ES y para las aplicaciones a las que proporciona compatibilidad la infraestructura de Java ES. Un clúster es un conjunto de equipos que no están firmemente acoplados y que, en conjunto, ofrecen una vista de cliente única de los servicios, los recursos del sistema y los datos. Internamente, el clúster utiliza equipos redundantes, interconexiones, almacenamiento de datos e interfaces de red para ofrecer alta disponibilidad en datos y servicios basados en clúster.

El software de Sun Cluster supervisa continuamente el estado de los nodos miembros y de otros recursos del clúster. En caso de fallo, el software de Sun Cluster interviene para iniciar la conmutación por error de los recursos que supervisa y utiliza la redundancia interna para proporcionar un acceso prácticamente continuo a estos recursos.

Los paquetes de servicios de datos de Sun Cluster (a veces denominados "agentes de Sun Cluster") están disponibles para todos los componentes de los servicios del sistema Java ES. También puede escribir agentes para componentes de aplicaciones personalizados.

Dado el control que permite el software de Sun Cluster, también puede ofrecer servicios escalables. Aprovechando el sistema de archivos global del clúster y la capacidad de que haya varios nodos en un clúster para ejecutar servicios de aplicaciones e infraestructura, la creciente demanda de estos servicios puede equilibrarse en varias instancias simultáneas de los servicios. Por lo tanto, si se configura correctamente, el software de Sun Cluster puede proporcionar alta disponibilidad y escalabilidad en una aplicación de empresa distribuida.

Debido a la redundancia necesaria para poder usar los entornos de Sun Cluster, la inclusión de Sun Cluster en una solución aumenta sustancialmente el número de equipos y vínculos de red necesarios en el entorno físico.

A diferencia de los servicios proporcionados por otros componentes de Java ES, los servicios de disponibilidad de Sun Cluster son servicios "de igual a igual" distribuidos. Por lo tanto, el software de Sun Cluster debe instalarse en cada equipo de un clúster.

Sun Cluster Geographic Edition proporciona una extensión del software de Sun Cluster. Esta aplicación protege las aplicaciones frente a interrupciones inesperadas mediante el uso de varios clústeres separados geográficamente y una infraestructura que repite los datos entre los clústeres.

---

**Nota** – Sun Cluster y Sun Cluster Geographic Edition sólo se admiten en el sistema operativo Solaris™ (SO Solaris).

---

## Síntesis de las tres dimensiones arquitectónicas

Cuando se visualizan juntas, las tres dimensiones arquitectónicas que se muestran en la [Figura 2-1](#) y que se describen en las secciones anteriores proporcionan una estructura para el diseño de soluciones de software distribuidas. Las tres dimensiones (dependencias de servicios de infraestructura, capas lógicas y calidad del servicio) destacan la función desempeñada por los componentes de Java ES en las arquitecturas de soluciones.

Cada dimensión representa una perspectiva arquitectónica diferente. Cada una de las arquitecturas de soluciones deberá tomar todas ellas en cuenta. Por ejemplo, los componentes distribuidos en cada capa lógica de una arquitectura de soluciones (dimensión 2) deberán estar apoyados por los componentes de infraestructura adecuados (dimensión 1) y los componentes de calidad de servicio adecuados (dimensión 3).

Igualmente, cualquier componente de una arquitectura de soluciones desempeña distintas funciones con respecto a las distintas dimensiones arquitectónicas. Por ejemplo, Directory Server se puede considerar como un componente de servidores en la capa de datos (dimensión 2) y como un proveedor de servicios de persistencia (dimensión 1). Debido al carácter central de Directory Server con respecto a estas dos dimensiones, los problemas de calidad del servicio (dimensión 3) son vitales para este componente de Java ES. Un fallo en Directory Server afectaría de forma significativa a un sistema empresarial, por lo que el diseño de alta disponibilidad de este componente es muy importante. Como Directory Server se utiliza para almacenar información de usuario y configuración confidencial, es también muy importante el diseño de la seguridad.

La interrelación de las tres dimensiones con respecto a los componentes de Java ES afecta al diseño de arquitecturas lógicas de soluciones y al diseño de arquitecturas de implementación de soluciones.

En esta guía no se describen metodologías de diseño detalladas basadas en la estructura arquitectónica representada por “[Estructura arquitectónica de Java ES](#)” en la [página 31](#). Sin embargo, la estructura arquitectónica tridimensional destaca aspectos de diseño que se deben conocer al implementar soluciones de software basadas en Java Enterprise System.

## Arquitectura de una solución de Java ES de ejemplo

Java ES admite una amplia gama de soluciones de software. Muchas soluciones se pueden diseñar e implementar con los valores de fábrica, sin que sea necesario llevar a cabo ningún desarrollo, utilizando los componentes incluidos en Java ES. Sin embargo, es posible que otras soluciones requieran mayores esfuerzos de desarrollo, por lo que deberá desarrollar componentes de J2EE personalizados que proporcionen nuevos servicios de presentación o de negocios. Puede encapsular estos componentes personalizados como servicios web que cumplan los estándares de la interfaz SOAP. Gran cantidad de soluciones implican una combinación de estos dos planteamientos.

En esta sección, se proporciona un ejemplo que muestra la forma en que Java ES admite el uso de una solución con los valores de fábrica elaborada a partir de los conceptos arquitectónicos de la sección anterior.

## Escenario de comunicaciones de las empresas

Normalmente, las empresas tienen que fomentar la comunicación entre sus empleados, específicamente los servicios de calendario y de correo electrónico. Dichas empresas, encontrarán ventajoso que sus empleados tengan un acceso personalizado a sitios web internos y otros recursos basándose en la autenticación de la empresa y servicios de autorización. Además, estas empresas desean que se pueda realizar un seguimiento de la identidad de los empleados en todos los servicios de la empresa, de forma que un único inicio de sesión web permita acceder a todos los servicios.

Estos requisitos específicos de la empresa, que representan únicamente un conjunto de ejemplo de los requisitos de la empresa, se resumen en la siguiente tabla.

**TABLA 2-4** Resumen de requisitos de negocio: escenario de comunicaciones

Requisito empresarial	Descripción	Servicios necesarios
Inicio de sesión único	Acceso a recursos y servicios de negocios seguros basándose en una única identidad con un inicio de sesión único para el acceso web.	Servicios de identidad
Mensajería Calendario	Mensajes de correo electrónico entre los empleados y con personas que no pertenecen a la empresa. Calendario electrónico del empleado y preparación de reuniones.	Servicios de comunicación y colaboración
Acceso al portal	Puntos de acceso personalizados basados en web para los servicios de comunicación como el correo electrónico y el calendario, así como páginas web internas.	Servicios de portal

Además, una empresa tiene requisitos relativos al rendimiento, la disponibilidad, la seguridad de red y la escalabilidad del sistema de software que proporciona estos servicios.

## Arquitectura lógica para el escenario de ejemplo

En la siguiente figura, se muestra una arquitecta lógica para proporcionar los servicios de identidad, comunicación y portal identificados en la [Tabla 2-4](#) mediante el uso de los componentes de Java ES y de Sun Java Communications Suite (Messaging Server, Calendar

Server e Instant Messaging, entre otros). La arquitectura trata configuraciones distintas lógicamente de Messaging Server como componentes separados debido a los distintos servicios que cada una proporciona.

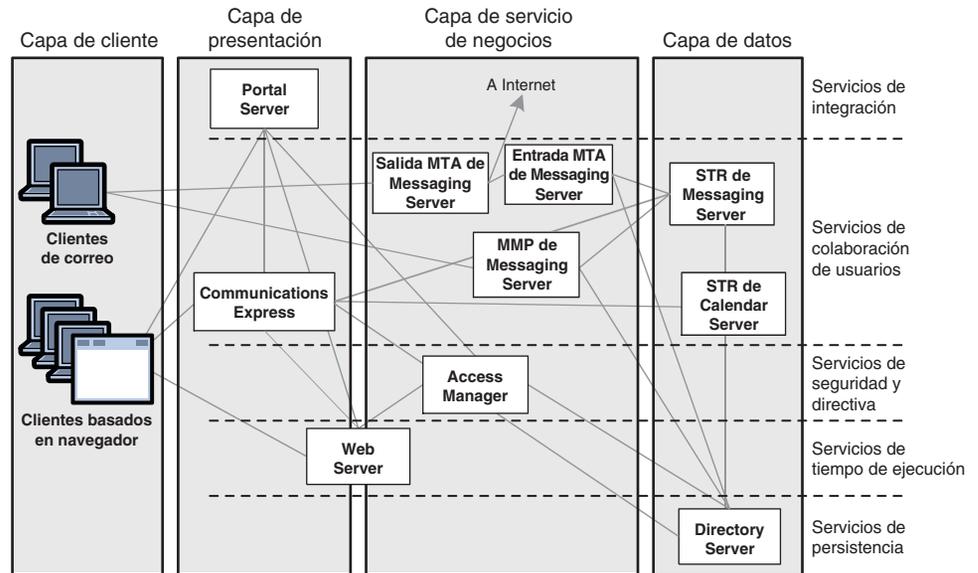


FIGURA 2-6 Arquitectura lógica para el escenario de comunicaciones de la empresa

Los componentes se colocan en una dimensión horizontal que representa las capas lógicas estándar y en una dimensión vertical que representa los niveles de servicio de infraestructura. Las interacciones entre los componentes dependen de sus funciones como servicios de infraestructura distribuidos (interacciones entre niveles de servicio de infraestructura) o de sus funciones en una arquitectura de aplicaciones de capas (interacciones dentro y entre capas lógicas).

En esta arquitectura, Access Manager, que accede a la información de usuario almacenada en Directory Server, actúa como árbitro de los servicios de autorización y autenticación de inicio de sesión único para Portal Server y en otros componentes basados en web en la capa de presentación. Los componentes de Messaging Server incluyen: un almacén de mensajes (Messaging Server -STR) en la capa de datos, que envía y recupera componentes en la capa de servicios de negocio; otro componente de acceso HTTP; y Communications Express en la capa de presentación.

La arquitectura lógica también muestra las dependencias de los servicios de infraestructura entre los distintos componentes. Portal Server, por ejemplo, depende de Communications Express para sus canales de mensajería y calendario. También depende de Access Manager para los servicios de autenticación y autorización. Estos componentes, a su vez, dependen de Directory Server para obtener la información de usuario y los datos de configuración. Varios componentes requieren los servicios de contenedores web proporcionados por Web Server.

Para obtener más información acerca del diseño de soluciones lógicas de Java ES, consulte la *Sun Java Enterprise System Deployment Planning Guide*.

## Arquitectura de implementación para el escenario de ejemplo

Al cambiar de la arquitectura lógica a la arquitectura de implementación, los requisitos de calidad del servicio son vitales. Por ejemplo, las subredes protegidas y los servidores de seguridad se pueden utilizar para crear una barrera de seguridad a los datos de copia de seguridad. Los requisitos de disponibilidad y escalabilidad pueden satisfacerse para varios componentes implementándolos en varios equipos y utilizando equilibradores de carga para distribuir las solicitudes entre los componentes duplicados.

Sin embargo, cuando hay requisitos de disponibilidad más exigentes y se necesitan grandes cantidades de almacenamiento en disco, otras soluciones de disponibilidad son más adecuadas. Por ejemplo, Sun Cluster se puede utilizar para el almacén de Messaging Server y la replicación multimaestro se puede utilizar para Directory Server.

Para obtener más información acerca del diseño de la implementación de soluciones de Java ES, consulte la *Sun Java Enterprise System Deployment Planning Guide*.

## Términos clave de este capítulo

En esta sección, se explican los términos técnicos clave empleados en este capítulo, con especial atención al uso de estos términos en el contexto de Java ES.

### componente de aplicación

Un **componente** de software desarrollado de forma personalizada para alguna función informática específica que proporciona **servicios de negocios** a los **usuarios finales** o a otros componentes de aplicación. Un componente de aplicación se ajusta normalmente a un modelo de componente distribuido (como CORBA o la plataforma J2EE). Estos componentes, juntos o por separado, pueden estar encapsulados como **servicios web**.

### arquitectura

Un diseño que muestra los bloques de construcción físicos y lógicos de una aplicación distribuida (o algún otro sistema de software) y las relaciones entre ellos. En el caso de una **aplicación de empresa distribuida**, el diseño arquitectónico utiliza generalmente la **arquitectura lógica** de la aplicación y la **arquitectura de implementación**.

### servicio de negocios

Un **componente de aplicación** o un ensamblado de componentes que realizan la lógica de negocios en nombre de varios clientes (y es, en consecuencia, un proceso con varios subprocesos). Un servicio de negocios también puede ser un conjunto de componentes distribuidos encapsulados como un **servicio web** o un **servidor** independiente.

---

<b>cliente</b>	Software que solicita <a href="#">servicios</a> de software. Un cliente puede ser un servicio que solicita otro servicio o un componente de la GUI al que accede un usuario final.
<b>arquitectura de implementación</b>	Un diseño general que determina la asignación de una <a href="#">arquitectura lógica</a> a un entorno informático físico. El entorno físico incluye los equipos de un entorno de intranet o Internet, los enlaces de red que se establecen entre ellos y otros dispositivos físicos necesarios para la compatibilidad del software.
<b>arquitectura lógica</b>	Un diseño que representa los bloques de construcción de una aplicación distribuida y las relaciones (o interfaces) existentes entre dichos bloques. La arquitectura lógica incluye los <a href="#">componentes de aplicación</a> distribuidos y los componentes de los servicios de infraestructura necesarios para su compatibilidad.
<b>servidor</b>	Un proceso de software con varios subprocesos (a diferencia de un servidor de hardware) que proporciona un <a href="#">servicio</a> distribuido o un conjunto coherente de servicios para los <a href="#">clientes</a> que acceden al servicio mediante una interfaz externa.
<b>servicio web</b>	Un servicio que responde a los protocolos de Internet estándares para funciones de accesibilidad, encapsulación de servicios y detección. Entre los estándares, se incluyen el protocolo de mensajería SOAP, la definición de la interfaz WSDL (Web Services Description Language) y el estándar de registro UDDI (Universal Description, Discovery and Integration).



## Funciones de integración de Java ES

---

Este capítulo proporciona una base conceptual y técnica para comprender las funciones que desempeñan un papel clave en la integración de componentes de Java ES en un único sistema de software. Estas funciones ayudan a comprender algunas de las ventajas de utilizar Java ES frente a la integración manual de productos de infraestructura dispares.

Este capítulo contiene las siguientes secciones:

- “El programa de instalación integrado de Java ES” en la página 51
- “Servicios de supervisión del sistema” en la página 53
- “Identidad integrada y servicios de seguridad” en la página 54
- “Términos clave de este capítulo” en la página 57

### El programa de instalación integrado de Java ES

Todos los componentes de Java ES se instalan utilizando un único instalador (también llamado "programa de instalación"). El programa de instalación de Java ES transfiere el software de Java ES a un sistema host. Este programa permite seleccionar e instalar cualquier número de componentes de Java ES en un host del entorno informático. También proporciona algunas opciones de configuración en el momento de la instalación, en función de los componentes de Java ES que se estén instalando.

El programa de instalación de Java ES no realiza, por sí mismo, instalaciones distribuidas. Para implementar una solución de software de Java ES distribuida, debe utilizar el programa de instalación de Java ES para instalar los componentes adecuados en cada equipo de su entorno. Debe utilizar una secuencia razonable de sesiones de instalación y procedimiento de configuración, basándose en la arquitectura de implementación y las dependencias de componentes.

El programa de instalación se ejecuta de forma interactiva en modo gráfico y basado en texto; también ofrece un modo de instalación silencioso mediante parámetros. Además del inglés, el programa de instalación admite los siguientes idiomas: francés, alemán, japonés, coreano, español, chino simplificado y chino tradicional.

En esta sección, se describen los diferentes aspectos del programa de instalación integrado de Java ES. Para obtener información más detallada, consulte la *Guía de instalación de Sun Java Enterprise System 5 para UNIX*.

## Comprobación del software previo

El programa de instalación examina el host en el que se esté realizando la instalación e identifica los componentes de Java ES que ya se hayan instalado. El programa de instalación realiza comprobaciones en varios niveles para asegurarse de que todos los componentes instalados previamente tengan un nivel de versión que permita una correcta interoperabilidad. El programa de instalación informa acerca de los componentes de software que son incompatibles y que deben actualizarse o quitarse.

Del mismo modo, el programa de instalación comprueba Java ES en busca de componentes compartidos como, por ejemplo, J2SE o NSS, que ya se hayan instalado y muestra las incompatibilidades (consulte “[Componentes compartidos](#)” en la [página 21](#)). Si continúa con la instalación, el programa de instalación actualiza automáticamente los componentes compartidos con las nuevas versiones.

## Comprobación de dependencias

El instalador realiza una comprobación exhaustiva de los componentes para comprobar que los componentes que elija para instalarlos funcionarán bien juntos. Numerosos componentes guardan relaciones de dependencia con otros componentes. Por este motivo, al seleccionar un componente para instalarlo, el programa de instalación incluye automáticamente los componentes y subcomponentes respecto de los que el componente seleccionado presenta relaciones de dependencia. No puede anular la selección de un componente si otro componente seleccionado depende de éste localmente. No obstante, si la relación de dependencia no es local, aunque reciba una advertencia, podrá continuar, ya que este proceso asume que la relación de dependencia será satisfecha por un componente presente en otro equipo host.

## Configuración inicial

Varios componentes de Java ES requieren una configuración inicial antes de que se puedan iniciar. El programa de instalación de Java ES suele realizar esta configuración inicial para algunos componentes.

Puede elegir que el programa de instalación realice esta configuración inicial (opción "Configurar ahora") u omitirla ("Configurar más tarde"), en cuyo caso tendrá que realizar la configuración de forma explícita para cada componente instalado una vez finalizada la instalación.

Si desea que el programa de instalación realice la configuración inicial, deberá suministrar la información necesaria durante la instalación. En concreto, puede especificar un conjunto de valores de parámetros comunes a todos los productos componentes, como un Id. de administrador y una contraseña.

## Desinstalación

Java ES también proporciona un programa de desinstalación para eliminar los componentes instalados en el equipo local por el programa de instalación de Java ES. El programa de desinstalación comprueba las relaciones de dependencia y emite una advertencia cuando detecta una dependencia. El programa de desinstalación no elimina los componentes compartidos de Java ES. Al igual que el programa de instalación, este programa se puede ejecutar en el modo gráfico, basado en texto o silencioso.

## Servicios de supervisión del sistema

Java ES incluye una nueva función de supervisión del sistema que proporciona supervisión de los servicios del sistema en tiempo real. La supervisión se implementa mediante Sun Java System Monitoring Framework ([componente compartido](#)) y Sun Java System Monitoring Console ([componente de producto](#)). Monitoring Framework se configura y habilita automáticamente para recopilar los datos de cada componente de Java ES instalado, mientras que Monitoring Console es la interfaz gráfica utilizada para visualizar los datos supervisados. Monitoring Console es un componente que puede seleccionarse durante la instalación de Java ES, mientras que Monitoring Framework se instala automáticamente.

La supervisión es el proceso de recopilación y presentación de los datos de tiempo de ejecución, y de cálculo de los criterios de calidad del servicio para que los administradores del sistema puedan evaluar el rendimiento y recibir alarmas. Durante la operación de tiempo de ejecución, los administradores interactúan con Monitoring Console para ver las estadísticas de rendimiento, establecer dinámicamente los umbrales de supervisión, definir los trabajos de supervisión personalizados y confirmar las alarmas.

# Identidad integrada y servicios de seguridad

Una función importante de Java ES es su administración integrada de las identidades de los usuarios, así como su estructura integrada de autorización y autenticación. En esta sección se ofrece información técnica para conocer los servicios de identidad y seguridad integrados proporcionados por Java ES.

## Identidad única

Dentro de un entorno de Java ES, cada usuario final tiene su identidad única integrada. En función de su [identidad única](#), a un usuario se le puede conceder acceso a varios recursos como, por ejemplo, portales, páginas web o servicios como el uso de mensajes y calendarios y la mensajería instantánea.

Esta función integrada de seguridad e identidad está basada en una estrecha colaboración entre Directory Server, Access Manager y otros componentes de Java ES.

El acceso de los usuarios a un servicio o recurso de Java ES se consigue almacenando información específica del usuario en una única entrada de usuario en un depósito o [directorio](#) de usuarios. Esta información normalmente incluye datos como una contraseña y un nombre exclusivos, así como una dirección de correo electrónico, un rol en una organización, preferencias de páginas web, etc. La información de la entrada de usuario puede utilizarse para autenticarlo, autorizar el acceso a recursos específicos u ofrecerle una variedad de servicios.

En el caso de Java ES, las entradas de los usuarios se almacenan en un directorio proporcionado por Directory Server. Cuando un usuario desea solicitar un servicio que ofrece un componente de Java ES, ese servicio utiliza Access Manager para autenticar al usuario y concederle acceso a los recursos específicos. El servicio solicitado comprueba la información de configuración específica del usuario en la entrada de directorio del usuario. El servicio utiliza dicha información para realizar la tarea solicitada por el usuario.

La siguiente figura ilustra el acceso a las entradas de usuario para realizar la autenticación y autorización del usuario y para proporcionarle servicios.

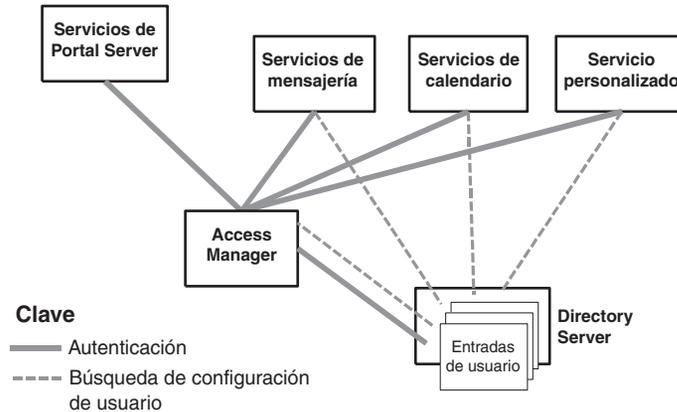


FIGURA 3-1 La entrada única de usuario es compatible con varios servicios

Mediante este sistema, cuando un usuario web inicia sesión en cualquier servicio de Java ES, se autentica automáticamente en otros servicios del sistema. Esta función, conocida como **inicio de sesión único**, es una característica muy eficaz proporcionada por Java ES.

## Autenticación e inicio de sesión único

Access Manager proporciona servicios de autenticación y autorización de Java ES. Access Manager usa la información de Directory Server para actuar como intermediario en la interacción de los usuarios con los servicios web de Java ES u otros servicios basados en web de una empresa.

Access Manager utiliza un componente externo denominado agente de directivas. El agente de directivas se conecta al servidor web que alberga un servicio o recurso asegurado por Access Manager. El agente de directivas intercede en nombre de Access Manager en solicitudes que realizan los usuarios a los recursos asegurados. En algunos componentes de Java ES como, por ejemplo, Portal Server, Access Manager SDK proporciona la función de agente de directivas.

### Autenticación

Access Manager incluye un servicio de autenticación para comprobar las identidades de los usuarios que soliciten acceso (mediante HTTP o HTTPS) a los servicios web dentro de una empresa. Por ejemplo, un empleado de una empresa que necesita buscar el número de teléfono de un compañero utiliza un explorador para ir a la libreta de teléfonos en línea de la empresa. Para iniciar una sesión en el servicio de libreta de teléfonos, el usuario debe proporcionar un Id. de usuario y una contraseña.

La secuencia de autenticación es la que se muestra en la [Figura 3-2](#). Un agente de directivas intercede en la solicitud para acceder a la libreta de teléfonos (1) y envía la solicitud al servicio de autenticación (2). El servicio de autenticación comprueba el Id. de usuario y la contraseña

con la información almacenada en Directory Server (3). Si la solicitud de inicio de sesión es válida, el usuario se autentica (4), (5) y (6), y podrá consultar la libreta de teléfonos de la empresa (7). Si la solicitud de inicio de sesión no es válida, se generará un error y el usuario no podrá autenticarse.

El servicio de autenticación también admite la autenticación basada en certificados sobre HTTPS.

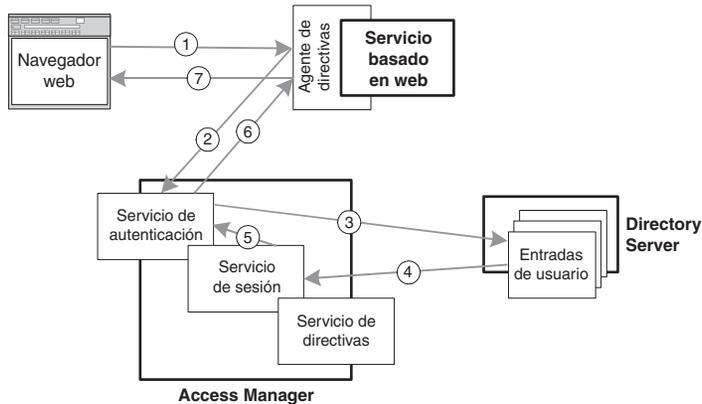


FIGURA 3-2 Secuencia de autenticación

## Inicio de sesión único

El escenario de autenticación descrito en los párrafos anteriores pasa por alto un paso importante. Cuando se comprueba la solicitud de autenticación de un usuario, el servicio de sesiones de Access Manager se activa (4), tal y como se muestra en la [Figura 3-2](#). El servicio de sesiones genera un testigo de sesión, que contiene la información sobre la identidad del usuario y un Id. de testigo (5). El testigo de la sesión se devuelve al agente de directivas (6) que lo envía (como una cookie) al navegador (7) desde el que se ha realizado la solicitud de autenticación.

Cuando el usuario autenticado intenta acceder a otro servicio asegurado, el explorador pasa el testigo de sesión al agente de políticas correspondiente. El agente de directivas comprueba con el servicio de sesión que la autenticación anterior del usuario aún es válida, permitiendo que el usuario acceda al segundo servicio sin necesidad de escribir el Id. de usuario y la contraseña de nuevo.

Por lo tanto, un usuario sólo necesita iniciar una sesión una vez para autenticarse en los distintos servicios basados en web que ofrece Java Enterprise System. La autenticación mediante el inicio de sesión único sigue en efecto hasta que el usuario cierre la sesión explícitamente o cuando ésta caduque.

## Autorización

Access Manager también incluye un servicio de directivas que proporciona control de acceso a los recursos basados en web en un entorno de Java ES. Una **directiva** es una regla que especifica quién tiene autorización para acceder a recursos específicos bajo ciertas condiciones. La secuencia de autorización se muestra en la figura que aparece a continuación.

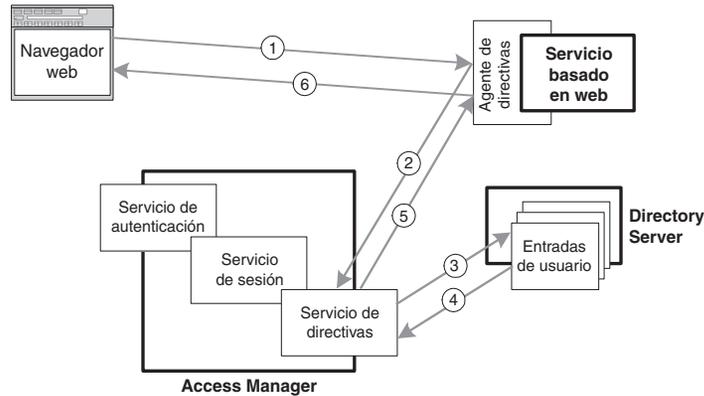


FIGURA 3-3 Secuencia de autorización

Cuando un usuario autenticado realiza una solicitud con respecto a recursos asegurados de Access Manager (1), el agente de directivas lo notifica al servicio de directivas (2), el cual utiliza la información de Directory Server (3) para evaluar la directiva de acceso que controla el recurso para ver si el usuario tiene permiso para acceder al recurso en cuestión (4). Si el usuario cuenta con privilegios de acceso (5), se admite la solicitud del recurso (6).

Access Manager proporciona los medios para definir, modificar, conceder, revocar y eliminar directivas en una empresa. Las directivas se almacenan y se configuran en Directory Server mediante atributos relacionados en entradas de organización. También pueden definirse roles para usuarios e incorporarse a definiciones de políticas.

Los agentes de directivas de Access Manager son los que ponen en práctica las directivas. Cuando el servicio de directivas rechaza una solicitud de acceso, el agente de directivas deniega el acceso a los recursos protegidos al usuario que lo solicita.

## Términos clave de este capítulo

En esta sección, se explican los términos técnicos clave empleados en este capítulo, con especial atención al uso de estos términos en el contexto de Java ES.

### directorio

Un tipo especial de base de datos optimizado para leer datos en lugar de escribirlos. La mayoría de los directorios se basan en LDAP (Protocolo ligero de acceso a directorios, del inglés Lightweight Directory Access Protocol), un protocolo estándar del sector.

<b>directiva</b>	Regla que describe quién está autorizado a acceder a un recurso específico y en qué condiciones concretas. La regla puede basarse en grupos de usuarios o roles de una organización.
<b>identidad única</b>	Identidad que tiene un usuario en virtud de una entrada de usuario única en un directorio de Java ES. En función de esa entrada de usuario única, un usuario puede acceder a varios recursos de Java ES, como el portal, las páginas web y los servicios como, por ejemplo, los servicios de mensajería, calendario y mensajería instantánea.
<b>inicio de sesión único</b>	Una función que hace posible la autenticación de un usuario para acceder a un servicio de un sistema distribuido que se aplica de forma automática a otros servicios del sistema.

## Ciclo de vida de las soluciones de Java ES

---

En este capítulo se describen los conceptos y la terminología relevantes para cada etapa del ciclo de vida de la solución de Java ES. El eje de este capítulo son las tareas de implementación, especialmente las tareas de diseño de implementación y de puesta en práctica.

En este capítulo se describen las tareas incluidas en cada etapa del ciclo de vida. Este capítulo contiene las siguientes secciones:

- “Etapa previa a la implementación” en la página 61
- “Implementación” en la página 62
- “Etapa posterior a la implementación” en la página 67
- “Términos clave de este capítulo” en la página 67

### Tareas del ciclo de vida de las soluciones

Sobre el ciclo de vida de las soluciones se realizó una introducción en el [Capítulo 1](#) como una aproximación estándar al despliegue de soluciones de negocio que usan el software Java ES. El diagrama del ciclo de vida se vuelve a mostrar en este capítulo para consultarlo con mayor facilidad.

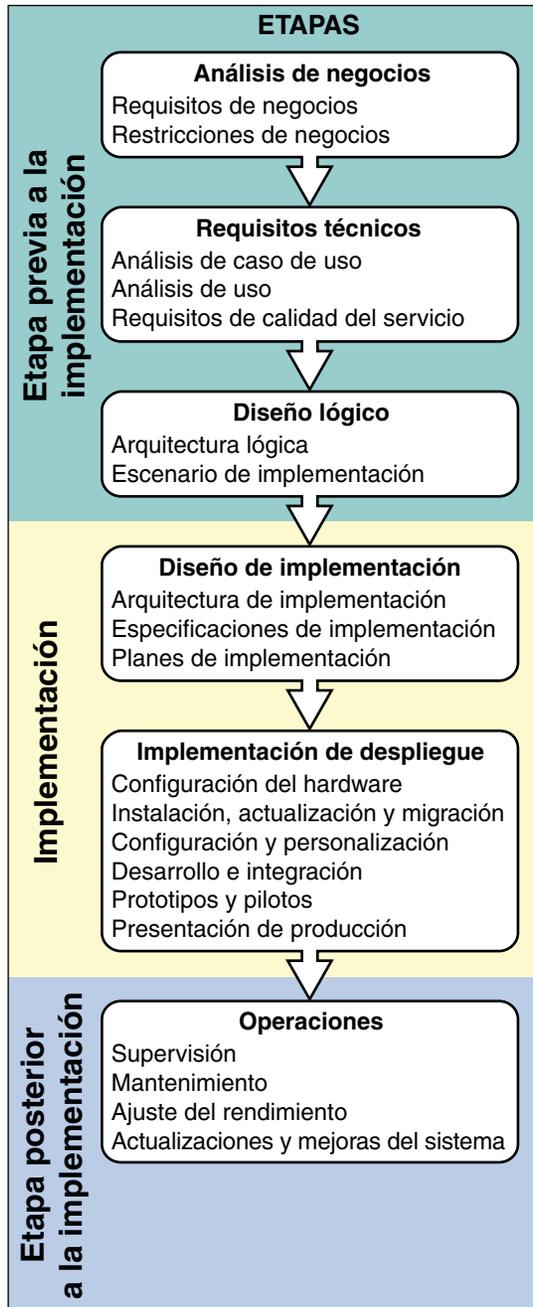


FIGURA 4-1 Tareas del ciclo de vida de las soluciones

## Etapa previa a la implementación

En las [etapa previa a la implementación](#) fases del ciclo de vida, un análisis de las necesidades del negocio se traduce en un [escenario de implementación](#). El escenario de implementación sirve como especificación para el diseño de la implementación.

Las tareas previas a la implementación se agrupan en tres etapas, tal y como se muestra en la [Figura 4-1](#):

- **Análisis de negocios.** Defina los objetivos empresariales de una implementación propuesta y especifique las limitaciones y los requisitos de negocios que se deben cumplir para conseguir ese objetivo.
- **Requisitos técnicos.** Utilice los resultados del análisis de negocios para crear [casos de uso](#) que perfilen la interacción del usuario con un sistema de software previsto. También puede determinar patrones de utilización para esos casos de uso. Sirviéndose tanto del análisis de negocios como del análisis de uso, se deben formular los requisitos de calidad del servicio (consulte la [Tabla 2-2](#)) que debe cumplir la implementación propuesta.
- **Diseño lógico.** Analice los casos de uso desarrollados en la etapa de requisitos técnicos para determinar los componentes de infraestructura de Java ES y los componentes de aplicación personalizados necesarios para proporcionar los servicios al usuario final. Mediante los conceptos que se describen en el [Capítulo 2](#), podrá diseñar una arquitectura lógica. La arquitectura lógica muestra todos los componentes y todas las interacciones entre los componentes que hacen falta para llevar a la práctica los casos de uso de una solución de software concreta.

La arquitectura lógica, junto con el rendimiento, la disponibilidad, la seguridad y otros requisitos de calidad del servicio se encapsulan en un escenario de implementación, tal y como se muestra en la siguiente figura. Para obtener más información acerca de la etapa previa a la implementación del ciclo de vida, consulte la *Sun Java Enterprise System Deployment Planning Guide*.

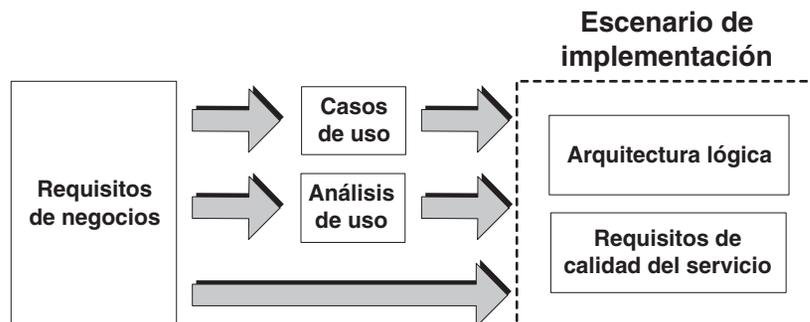


FIGURA 4-2 Especificación de un escenario de implementación

# Implementación

En las etapas de [implementación](#) del ciclo de vida, un escenario de implementación se convierte en un diseño de implementación que, posteriormente, se implementará, se probará y se implantará en un entorno de producción.

El proceso de implementación normalmente engloba componentes de software en todas las capas y en todos los niveles de servicios de infraestructura necesarios para poder utilizar una solución de software. En general, debe implementar tanto los componentes de aplicación desarrollados de forma personalizada (componentes J2EE, servicios web u otros servidores) como los componentes de Java ES necesarios para poder utilizar la solución.

Las tareas de implementación se agrupan en dos etapas, tal y como se muestra en la [Figura 4-1](#):

- “[Diseño de la implementación](#)” en la [página 62](#). El diseño de la implementación depende de la arquitectura lógica de la solución y del rendimiento, la disponibilidad, la seguridad, la escalabilidad, la facilidad de mantenimiento y otros requisitos de calidad del servicio que debe satisfacer la solución. La dimensión de calidad de servicio de la arquitectura de implementación desempeña un papel fundamental en la etapa de diseño de implementación.
- “[Implementación de despliegue](#)” en la [página 64](#). La implementación de un diseño de despliegue es a menudo un proceso repetitivo que implica la configuración del hardware, la instalación y configuración del software, el desarrollo y la integración, así como pruebas y otros aspectos de una presentación de producción.

En las siguientes secciones se examinan estas dos fases del proceso de implementación.

## Diseño de la implementación

En la etapa de diseño de la implementación, se crea una arquitectura de implementación general seguida de las especificaciones de implementación detalladas.

### Arquitectura de implementación

Una arquitectura de implementación se crea asignando los bloques de construcción lógicos de una aplicación (la arquitectura lógica) a un entorno informático físico de modo que se cumplan los requisitos de calidad del servicio especificados en el escenario de implementación. El escenario de implementación se traduce en una arquitectura de implementación, como se muestra en la siguiente figura.

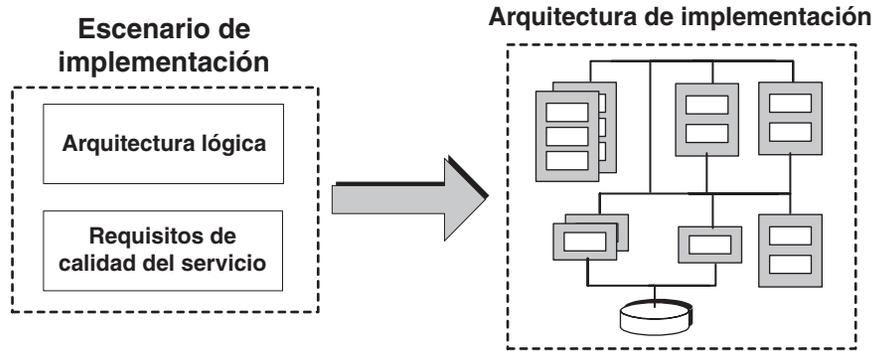


FIGURA 4-3 Conversión de un escenario de implementación en una arquitectura de implementación

Un aspecto de este diseño arquitectónico es el tamaño del entorno físico (que determina el número de equipos y estima los requisitos de potencia del procesador y de memoria RAM) para que satisfaga los requisitos de rendimiento, disponibilidad, seguridad y otros requisitos de calidad del servicio. Una vez completadas las tareas relacionadas con el tamaño, deberá asignar componentes de aplicación y los de Java ES a los distintos equipos en el entorno físico. La arquitectura de implementación resultante debe tener en cuenta las funciones de los diferentes equipos, las características de los servicios de infraestructura del sistema y las restricciones del coste total de propiedad o del coste total de disponibilidad.

Cuanto mayor sea el número de componentes de Java ES incluidos en el escenario de implementación, más exigentes serán los requisitos de calidad del servicio, así como el diseño de los equipos de gran potencia y el ancho de banda de la red. Cuando el hardware es limitado, o excesivamente caro, es posible que deba evaluar las concesiones entre los costes fijos (hardware) y los costes variables (requisitos de recursos humanos) o entre los diferentes requisitos de calidad del servicio. También es posible que tenga que aumentar la sofisticación del diseño.

El diseño de una arquitectura de implementación a menudo se realiza de forma iterativa. Las [arquitecturas de implementación de referencia](#) sirven como punto de partida para el diseño de la implementación de Java ES.

Una arquitectura de referencia se basa en un escenario de implementación específico: una arquitectura lógica con requisitos de calidad del servicio específicos. En la arquitectura de referencia, se implementa una solución de software en un entorno físico específico de una forma tal que satisface los requisitos de calidad del servicio especificados. La prueba de rendimiento en cargas específicas se basa en el mismo conjunto de casos de uso a partir del que se desarrolló el escenario de implementación. La documentación de referencia de la arquitectura está disponible para los clientes de Java ES y está sujeta a un acuerdo de confidencialidad.

Basándose en una arquitectura de implementación de referencia o en una combinación de arquitecturas de referencia, puede diseñar una primera aproximación a una arquitectura de implementación que reúna los requisitos de su propio escenario de implementación. Puede

ajustar las arquitecturas de referencia o utilizarlas como puntos de referencia, teniendo en cuenta la diferencia entre su propio escenario de implementación y aquéllos en los que se basan las arquitecturas de referencia. De este modo, puede valorar el impacto de sus propias necesidades de tamaño del sistema, rendimiento, seguridad, disponibilidad, capacidad y facilidad de mantenimiento.

## Especificaciones de implementación

Las especificaciones de implementación proporcionan los datos necesarios para llevar a la práctica una arquitectura de implementación. Las especificaciones incluyen generalmente la siguiente información:

- Hardware existente, incluidos equipos, dispositivos de almacenamiento, equilibradores de carga y cableado de red
- Sistemas operativos
- Diseño de red, incluidas las subredes y las zonas de seguridad
- Datos de diseño de disponibilidad
- Daños de diseño de seguridad
- Información sobre el diseño del directorio necesaria para la provisión de los usuarios finales

## Planes de implementación

Los planes de implementación describen cómo realizar las planificaciones para llevar a cabo las distintas tareas de la etapa de puesta en práctica de la implementación. Los planes cubren generalmente las siguientes tareas:

- Configuración del hardware
- Instalación, actualización y migración de software
- Configuración y personalización del sistema
- Desarrollo e integración
- Pruebas
- Presentación de producción

## Implementación de despliegue

El despliegue de un diseño de implementación consta de las tareas que se incluyen en la sección anterior y que se muestran en la [Figura 4-1](#). El orden de estas tareas no es rígido ya que el proceso de implementación es repetitivo por naturaleza. En las siguientes subsecciones se describe cada tarea principal de implementación de despliegue en el orden en que se realizan normalmente.

## Configuración del hardware

La especificación de implementación incluye todos los datos del entorno físico: los equipos, el diseño de red, el hardware de red (incluidos el cableado, los conmutadores, los enrutadores y los equilibradores de carga), los dispositivos de almacenamiento, etc. Todas estas necesidades de hardware deben configurarse conforme a la plataforma que admite la solución de Java ES.

## Instalación, actualización y migración de software

La arquitectura de implementación, junto con los detalles adicionales que se proporcionan en las especificaciones de implementación, indica los componentes de aplicación y los componentes de Java ES que se instalarán en cada equipo del entorno físico. Debe utilizar el programa de instalación integrado de Java ES para instalar los componentes adecuados de Java ES en cada equipo de la arquitectura de implementación (consulte [“El programa de instalación integrado de Java ES” en la página 51](#)).

El plan de instalación describe la secuencia y el alcance de las sesiones del programa de instalación. Sin embargo, el planteamiento que utilice para realizar la instalación puede depender de si está realizando una nueva instalación de Java ES, si está actualizando los componentes de Java ES instalados anteriormente o si está sustituyendo componentes de terceros por otros de Java ES. Los dos últimos escenarios de adopción de Java ES a menudo requieren que se migren datos o códigos de aplicaciones para que puedan ser compatibles.

## Configuración y personalización del sistema

Debe completar una serie de tareas de configuración para que los distintos componentes del sistema funcionen como un sistema integrado. La primera de estas tareas es la configuración inicial necesaria para que cada componente individual del sistema se inicie. En segundo lugar, debe configurarse cada componente de Java ES para que pueda comunicarse con los componentes con los que interactúa.

También debe configurar la alta disponibilidad, en función de la solución de disponibilidad para cada componente. Es necesario realizar la provisión de los usuarios para que éstos puedan acceder a los distintos servicios. También hay que configurar los controles y las directivas de autorización y autenticación (consulte [“Identidad integrada y servicios de seguridad” en la página 54](#)).

En la mayoría de los casos, las tareas de configuración incluyen algún grado de personalización de los componentes de Java ES para obtener la función exacta que se necesita. Por ejemplo, debe personalizar normalmente Portal Server para que proporcione canales de portal, Access Manager para que realice las tareas de autorización, etc.

## Desarrollo e integración

La arquitectura lógica especificada en el escenario de implementación determina generalmente el ámbito del trabajo de [desarrollo](#) necesario para desplegar una solución.

Para algunas implementaciones, el proceso de desarrollo puede ser bastante amplio, ya que hay que desarrollar nuevos servicios de presentación y de negocios desde el principio utilizando componentes de J2EE que se ejecutan en un entorno de Application Server o Web Server. En esos casos, es recomendable elaborar un prototipo de la solución y realizar pruebas de concepto antes de embarcarse en un esfuerzo de desarrollo completo.

Para las soluciones que requieren un desarrollo amplio, Sun Java™ Studio proporciona herramientas para programar componentes distribuidos o servicios de negocios. Las herramientas para desarrolladores de Sun Java Studio simplifican la programación y comprobación de las aplicaciones admitidas por la infraestructura de Java ES.

En algunas situaciones, los componentes de Java ES pueden estar integrados con aplicaciones heredadas o de terceros. Estas integraciones pueden implicar los directorios existentes o servicios de datos en la capa de datos o componentes existentes en la capa de servicios empresariales. La integración de componentes de Java ES con estos sistemas puede requerir la migración de datos o de código de aplicaciones.

La plataforma J2EE proporciona una estructura de conector que permite integrar las aplicaciones existentes en el entorno de Application Server, desarrollando adaptadores de recursos J2EE, mientras que Message Queue proporciona una función robusta de mensajería asíncrona para integrar diversas aplicaciones.

## **Pruebas de prototipos y pilotos**

En función del nivel de personalización o la cantidad de trabajo de desarrollo necesario, deberá verificar en algún momento su arquitectura de implementación: debe probar la solución frente a los casos de uso para verificar que se cumplan los requisitos de calidad del servicio.

Si tiene pocos servicios personalizados (una implementación prácticamente estándar), la solución simplemente requerirá la personalización de los componentes de Java ES y una prueba piloto del sistema.

No obstante, si ha desarrollado bastante lógica de aplicación nueva y ha creado servicios personalizados, esta prueba puede ser más amplia, ya que debería probar el prototipo, la integración, etc.

Si esta prueba revela defectos en la arquitectura de implementación, deberá modificar la arquitectura y probarla de nuevo. Este proceso iterativo debe producir al final una arquitectura de implementación y una implementación que esté lista para ponerla en práctica en un entorno de producción.

## **Presentación de producción**

La presentación de producción implica llevar a la práctica la implementación en un entorno de producción. Esta etapa conlleva la instalación, configuración e inicio de las aplicaciones distribuidas y los servicios de infraestructura en un entorno de producción, la provisión de

usuarios finales del sistema de producción, la configuración del inicio de sesión único y las directivas de acceso, etc. Normalmente, se empieza con una implementación limitada y se pasa a una implementación en toda la organización. En este proceso, debe realizar ejecuciones de prueba en las que se apliquen cargas progresivamente mayores para confirmar que se cumplen los requisitos de calidad del servicio.

## Etapa posterior a la implementación

En la [etapa posterior a la implementación](#) del ciclo de vida, se ejecuta una solución implementada en un entorno de producción. La etapa de operaciones del ciclo de vida conlleva las siguientes tareas:

- **Supervisión.** Incluye la supervisión periódica del rendimiento del sistema y de sus funciones.
- **Mantenimiento.** Incluye las funciones administrativas diarias, como agregar usuarios finales nuevos al sistema, cambiar contraseñas, agregar nuevos usuarios administrativos, cambiar los privilegios de acceso, realizar copias de seguridad periódicas, etc.
- **Ajuste del rendimiento.** Incluye la utilización de la información de supervisión periódica para encontrar cuellos de botella en las operaciones del sistema e intentar eliminar dichos cuellos de botella cambiando las propiedades de la configuración, agregando capacidad, etc.
- **Actualizaciones y mejoras del sistema.** Incluye la adición de nuevos componentes de Java ES a un sistema para agregar nuevas funciones o para sustituir componentes que no sean de Java ES. Estos cambios pueden requerir un nuevo diseño del sistema, comenzando por las etapas iniciales del ciclo de vida de la solución. Las tareas de actualización son más limitadas y normalmente se trata de actualizaciones de los componentes de Java ES.

## Términos clave de este capítulo

En esta sección, se explican los términos técnicos clave empleados en este capítulo, con especial atención al uso de estos términos en el contexto de Java ES.

<b>implementación</b>	Etapa del ciclo de vida de una solución de Java ES en la que un escenario de implementación se convierte en un diseño de implementación que luego se desplegará. También se hará un prototipo de éste y, por último, se aplicará en un entorno de producción. El producto final de este proceso también se denomina "implementación" (o solución implementada).
<b>escenario de implementación</b>	Una <a href="#">arquitectura lógica</a> para una solución de Java ES y los requisitos de calidad del servicio que debe satisfacer la solución para ajustarse a las necesidades empresariales. Entre los requisitos de calidad del servicio, se incluyen aquellos relacionados con el rendimiento, la disponibilidad, la seguridad, la facilidad de mantenimiento, la escalabilidad o la capacidad latente. El escenario de implementación es el punto de partida del diseño de implementación.

<b>desarrollo</b>	Una tarea del proceso de implementación de una solución de Java ES mediante la que los componentes personalizados de una <a href="#">arquitectura de implementación</a> se programan y prueban.
<b>etapa previa a la implementación</b>	Una etapa del ciclo de vida de una solución de Java ES en la que las necesidades empresariales se convierten en un <a href="#">escenario de implementación</a> : una <a href="#">arquitectura lógica</a> y un conjunto de requisitos de calidad del servicio que debe satisfacer una solución.
<b>etapa posterior a la implementación</b>	Una etapa del proceso de vida de una solución de Java ES en la que las aplicaciones distribuidas se inician, se supervisan y se ajustan para optimizar su rendimiento, y se actualizan dinámicamente para incluir nuevas funciones.
<b>arquitectura de implementación de referencia</b>	Una <a href="#">arquitectura de implementación</a> que se ha diseñado, desplegado y probado para proporcionar un rendimiento. Las arquitecturas de implementación de referencia se utilizan como puntos de partida para el diseño de arquitecturas de implementación de soluciones personalizadas.
<b>caso de uso</b>	Una tarea específica del usuario final o un conjunto de tareas realizadas por una <a href="#">aplicación de empresa distribuida</a> que se utiliza como base para el diseño, la prueba y la evaluación del rendimiento de una aplicación.

## Componentes de Java ES

---

Java ES está formado por una recopilación de [componentes de productos](#) y [componentes compartidos](#) que permiten de forma conjunta el uso de aplicaciones distribuidas en una red. Durante la instalación, el programa de instalación de Java ES presenta componentes que se pueden seleccionar, muchos de los cuales cuentan a su vez con subcomponentes que permiten su selección. Estos componentes y subcomponentes se muestran en este apéndice.

Este apéndice proporciona una breve descripción de los componentes de Java ES con información de carácter general. Para obtener información detallada sobre componentes específicos, consulte los conjuntos de documentación de los componentes disponibles en <http://docs.sun.com/app/docs/prod/entsys.5>. También puede encontrar una amplia gama de información y recursos de Java ES en <http://www.sun.com/bigadmin/hubs/javaes/>.

Los componentes de Java ES que aparecen en este apéndice se agrupan por categorías y se describen en las siguientes secciones:

- “Componentes de servicios del sistema” en la página 69
- “Componentes de calidad del servicio” en la página 73
- “Componentes compartidos” en la página 77

### Componentes de servicios del sistema

Los componentes de los servicios del sistema de Java ES ofrecen los servicios de infraestructura necesarios para el uso de aplicaciones de empresa distribuidas. Estos servicios se describen en “¿Por qué necesita Java ES?” en la página 15, entre los que se incluyen los servicios de portal, los servicios de identidad y seguridad, los servicios web y de aplicaciones, y los servicios de disponibilidad. Los componentes de servicios del sistema de Java ES se describen en las siguientes secciones:

- “Access Manager 7.1” en la página 70
- “Application Server Enterprise Edition 8.2” en la página 71
- “Directory Server Enterprise Edition 6.0” en la página 71

- “Java DB 10.1” en la página 72
- “Message Queue 3.7 UR 1” en la página 72
- “Portal Server 7.1” en la página 72
- “Service Registry 3.1 ” en la página 73
- “ Web Server 7.0 ” en la página 73

## Access Manager 7.1

Sun Java System Access Manager (Access Manager) integra servicios de autenticación y autorización, agentes de directivas y federación de identidades para proporcionar una completa solución para la protección de los recursos de red. Access Manager impide el acceso no autorizado a las aplicaciones de servicios web y al contenido web, proporcionando una infraestructura que permite a las organizaciones administrar las identidades de los clientes, los empleados y los socios que utilicen los servicios basados en web y las aplicaciones que no sean web. Dado que estos recursos pueden estar distribuidos por una amplia gama de redes internas y externas, los atributos, las directivas y las autorizaciones se definen y se aplican a cada identidad para administrar así el acceso a estas tecnologías.

Access Manager incluye los siguientes subcomponentes:

- **Access Manager Core Services.** Proporciona los medios para crear y administrar identidades de usuarios, así como para definir y evaluar las directivas que proporcionan acceso a los recursos de Java ES basándose en las identidades de los usuarios.
- **Consola de administración de Access Manager.** Consolida servicios de identidad y administración de directivas. También proporciona una única interfaz gráfica para los usuarios con objeto de crear y administrar cuentas de usuario, atributos de servicio y reglas de acceso en Directory Server.
- **Common Domain Services for Federation Management.** Permiten al usuario utilizar una única identidad para acceder a aplicaciones ofrecidas por varios proveedores de servicios afiliados.
- **Access Manager SDK.** Proporciona una interfaz remota para Access Manager. Este subcomponente debe instalarse en cualquier equipo que aloje un componente de Java ES que acceda a Access Manager de forma remota.
- **Interfaz de usuario de la autenticación distribuida de Access Manager.** Proporciona una interfaz de usuario que permite a un agente de directivas o a una aplicación implementada en un área no protegida comunicarse con el servicio de autenticación de Access Manager instalado en un área protegida de la implementación.
- **Access Manager Client SDK.** Permite a los usuarios implementar aplicaciones independientes que puedan acceder al servidor de Access Manager para utilizar los servicios como, por ejemplo, la autenticación, el inicio de sesión único, la autorización, la auditoría, el registro y el lenguaje de marcado de aserción de seguridad (SAML).
- **Cliente de conmutación por error de sesión de Access Manager.** Es necesario para configurar la conmutación por error de sesión de Access Manager.

## Application Server Enterprise Edition 8.2

Sun Java System Application Server (Application Server) proporciona una plataforma compatible con J2EE para desarrollar y proporcionar servicios web y aplicaciones de Java del servidor. Entre sus funciones principales, se incluyen la administración escalable de transacciones, el tiempo de ejecución de persistencia administrado por contenedor, el rendimiento de los servicios web, la función de clúster, el estado de sesión de alta disponibilidad, la seguridad y las funciones de integración.

Application Server incluye los siguientes subcomponentes:

- **Domain Administration Server.** Proporciona funciones administrativas de servidor, como administración y configuración de Application Server e implementación de componentes y aplicaciones J2EE.
- **Agente del nodo de Application Server.** Proceso ligero que se ejecuta en cada equipo que aloja las instancias del servidor y realiza una serie de tareas administrativas, entre las que se incluyen el inicio, la parada y el reinicio de las instancias del servidor.
- **Herramienta de administración de línea de comandos** Proporciona clientes para la administración mediante línea de comandos que permiten administrar y configurar instalaciones de Application Server y aplicaciones alojadas. La herramienta también ayuda a implementar aplicaciones.
- **Complemento de equilibrado de carga** Se utiliza para distribuir la carga de trabajo entre varias instancias de Application Server (ya sean independientes o agrupadas en clústeres), aumentando así el rendimiento general del sistema. También se utiliza para permitir las solicitudes de conmutación por error de una instancia del servidor a otra.
- **Aplicaciones de ejemplo.** Se incluyen con la instalación completa de Application Server.

## Directory Server Enterprise Edition 6.0

Sun Java System Directory Server (Directory Server) es un directorio basado en LDAP que proporciona un servicio de directorios centralizado para la información de la intranet, la red y la extranet. Directory Server se integra con los sistemas existentes y actúa como depósito centralizado para la consolidación de la información de los empleados, los clientes, los proveedores y los socios. Puede ampliar Directory Server para administrar perfiles de usuario y preferencias, así como la autenticación de usuarios de la extranet.

Directory Server incluye los siguientes subcomponentes:

- **Directory Server 6 Core Server.** Proporciona medios flexibles, seguros y escalables para almacenar y administrar datos de identidades.
- **Centro de control del servicio de directorios.** Proporciona una interfaz de administración basada en navegador para configurar los servicios de directorios y los servicios proxy de directorios.

- **Utilidad de línea de comandos de Directory Server.** Permite realizar tareas de administración desde la línea de comandos.
- **Directory Proxy Server 6 Core Server.** Mejora la seguridad ofreciendo funciones de directorio virtual y aumentando la disponibilidad y la escalabilidad del servicio de directorios.

## Java DB 10.1

Java DB proporciona una base de datos ligera para la implementación de aplicaciones de Java. Java DB es una distribución compatible de Sun de la base de datos de código abierto Apache Derby, implementada por competo mediante tecnología de Java. Java ES 5 es la primera versión que incluye Java DB como componente de producto. Java DB se introdujo por primera vez como componente compartido en Java ES 2005Q4, con el nombre de Derby Database.

Java DB incluye los siguientes subcomponentes:

- Java DB Client
- Java DB Server

## Message Queue 3.7 UR 1

Sun Java System Message Queue (Message Queue) es una solución basada en estándares para el problema de la comunicación interna entre las aplicaciones que permite enviar mensajes con fiabilidad. Message Queue: es un sistema de mensajería empresarial que implementa el estándar abierto Java Message Service (JMS).

Además de ser proveedor de JMS, Message Queue cuenta con funciones que superan los requisitos mínimos de la especificación JMS. Con el software de Message Queue, los procesos que se ejecuten en plataformas y en sistemas operativos diferentes se pueden conectar a un servicio de Message Queue común para enviar y recibir información. Los desarrolladores de aplicaciones se pueden centrar en la lógica de negocios de sus aplicaciones en lugar de en los detalles concretos referentes a la forma en que sus aplicaciones se comunican a través de la red.

El programa de instalación de Java ES incluye Message Queue como único componente instalable.

## Portal Server 7.1

Sun Java System Portal Server (Portal Server) es una solución de servidor de portal habilitada para identidades. Portal Server combina servicios de portal claves como, por ejemplo, la personalización, la adición, la seguridad, la integración y la búsqueda.

El programa de instalación de Java ES proporciona Portal Server como único componente instalable.

## Service Registry 3.1

Sun Java System Service Registry (Service Registry) es un depósito que actúa como registro (UDDI) de los servicios web y como registro XML de negocios empresariales (ebXML) para que sea posible usar aplicaciones web con arquitectura orientada a servicios (SOA). El registro UDDI se utiliza para registrar y detectar servicios web, y el registro ebXML se usa para almacenar los artefactos de información necesarios para que sea posible la integración de los procesos empresariales. Estos elementos de información incluyen metadatos como, por ejemplo, el esquema XML, las reglas de procesos de negocio, los controles de versión, los esquemas de clasificación, etc.

Service Registry incluye los siguientes subcomponentes:

- Compatibilidad de cliente de Service Registry
- Compatibilidad de implementación de Service Registry

## Web Server 7.0

Sun Java System Web Server (Web Server) es un servidor web seguro para varios procesos y subprocesos creado a partir de estándares del mercado. Web Server proporciona un gran rendimiento, fiabilidad, escalabilidad y facilidad de administración para medianas y grandes empresas.

Web Server incluye los siguientes subcomponentes:

- CLI de Web Server
- Web Server Core
- Ejemplos de Web Server

# Componentes de calidad del servicio

Los componentes de Java ES se utilizan principalmente para mejorar la calidad de los servicios proporcionados por los componentes de servicios del sistema o por los componentes de aplicaciones distribuidas. Algunos de los componentes se utilizan para ofrecer un tiempo de actividad del sistema prácticamente continuo, otros son componentes de acceso que se utilizan para garantizar el acceso seguro de los usuarios finales a los servicios del sistema y otros son componentes de administración del sistema que se utilizan para mejorar la facilidad de mantenimiento de las soluciones de Java ES.

Los componentes que admiten los componentes de servicios de Java ES se agrupan en las siguientes categorías, que se describen en esta sección:

- “Componentes de disponibilidad” en la [página 74](#)
- “Componentes de acceso” en la [página 76](#)

- “Componentes de supervisión” en la página 77

## Componentes de disponibilidad

Los componentes de disponibilidad ofrecen un tiempo de actividad prácticamente continuo para los componentes de servicio del sistema y los componentes de aplicaciones. Los siguientes componentes de disponibilidad de Java ES se describen en esta sección:

- “High Availability Session Store 4.4.3” en la página 74
- “Sun Cluster 3.1 8/05 y Sun Cluster Agents 3.1” en la página 74
- “Sun Cluster Geographic Edition 3.1 2006Q4” en la página 75

### High Availability Session Store 4.4.3

Sun Java System High Availability Session Store (HADB) es un almacén de sesión de alta disponibilidad que proporciona un almacén de datos que se puede usar para que los datos de la aplicación estén disponibles, incluso aunque se produzca un fallo. Esta capacidad es especialmente importante para restaurar la información de estado asociada a una sesión de clientes. Sin esta capacidad, un fallo durante una sesión obliga a repetir todas las operaciones cuando se restablezca la sesión.

Los siguientes componentes de Java ES proporcionan servicios que almacenan la información de estado de la sesión: Application Server, Access Manager y Message Queue. Sin embargo, Application Server es el único componente que puede utilizar los servicios de HADB para mantener el estado de la sesión durante un fallo.

El programa de instalación de Java ES ofrece HADB como único componente instalable. Sin embargo, un subcomponente de servidor y de cliente son necesarios para proporcionar servicios HADB.

### Sun Cluster 3.1 8/05 y Sun Cluster Agents 3.1

---

**Nota** – Los componentes de Sun Cluster sólo se admiten en la plataforma Solaris.

---

El software de Sun Cluster proporciona servicios de alta disponibilidad y escalabilidad para Java ES, así como para aplicaciones basadas en la infraestructura de Java ES.

Un clúster es un conjunto de equipos que no están firmemente acoplados (nodos de clúster) y que proporcionan en conjunto al cliente una vista única de los servicios, los recursos de sistema y los datos. Internamente, el clúster utiliza equipos redundantes, interconexiones, almacenamiento de datos e interfaces de red para ofrecer alta disponibilidad en datos y servicios basados en clúster. El software de Sun Cluster supervisa continuamente el estado de los nodos de miembros y otros recursos de clúster y utiliza la redundancia interna para ofrecer acceso casi continuo a estos recursos aunque se produzca un error.

---

El programa de instalación de Java ES proporciona el subcomponente Sun Cluster Core y Sun Cluster Agents como componentes que se pueden instalar por separado. Los siguientes agentes de Sun Cluster se incluyen en Java Enterprise System.

---

**Nota** – HA hace referencia a la *alta disponibilidad* en esta lista.

---

- HA Application Server
- HA Message Queue
- HA Directory Server
- HA Messaging Server
- HA Application Server EE (HADB)
- HA/Scalable Web Server
- HA Instant Messaging
- HA Calendar Server
- HA Apache Tomcat
- HA Apache
- HA DHCP
- HA DNS
- HA MySQL
- HA Sun N1 Service Provisioning
- HA NFS
- HA Oracle
- HA Samba
- HA Sun N1 Grid Engine
- Contenedores de HA Solaris

---

**Nota** – Esta lista de agentes es diferente en las plataformas SPARC y x86. Para obtener información detallada sobre Sun Cluster Agents, consulte la documentación de Sun Cluster en <http://docs.sun.com/app/docs/prod/entsys.5>.

---

## Sun Cluster Geographic Edition 3.1 2006Q4

Sun Cluster Geographic Edition es una extensión por capas del software de Sun Cluster. Esta extensión protege las aplicaciones de interrupciones inesperadas mediante el uso de varios clústeres separados geográficamente y una infraestructura redundante que repite los datos entre estos clústeres. Java ES 5 es la primera versión que incluye Sun Cluster Geographic Edition como componente de producto de Java ES.

Sun Cluster Geographic Edition incluye los siguientes subcomponentes:

- Sun Cluster Geographic Edition Core
- Sun StorEdge Availability Suite
- Hitachi Truecopy Data Replication Support (sólo para SPARC)
- EMC SRDF Data Replication

---

**Nota** – Sun Cluster Geographic Edition no se admite en Solaris x86.

---

## Componentes de acceso

Los componentes de acceso proporcionan acceso cliente a los servicios del sistema, a menudo desde ubicaciones de Internet fuera de un servidor de seguridad de la empresa. Los siguientes componentes de acceso de Java ES se describen en esta sección:

- “Portal Server Secure Remote Access 7.1” en la página 76
- “Web Proxy Server 4.0.4” en la página 76

### Portal Server Secure Remote Access 7.1

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) amplía las funciones de Portal Server ofreciendo acceso remoto seguro basado en navegador al contenido y los servicios de Portal Server desde cualquier navegador remoto, eliminando así la necesidad de usar software de cliente. La integración con Portal Server garantiza que los usuarios reciban un acceso cifrado seguro al contenido y a los servicios a los que tienen permiso para acceder.

Portal Server Secure Remote Access incluye los siguientes subcomponentes:

- **Portal Server Secure Remote Access Core.** Proporciona funcionalidad central.
- **Puerta de enlace.** Proporciona una interfaz y una barrera de seguridad entre las sesiones de usuarios remotos que se originan desde Internet o una intranet corporativa. La puerta de enlace presenta el contenido de forma segura desde los servidores de aplicaciones y los servidores web internos mediante una única interfaz a los usuarios remotos, y controla la comunicación entre Portal Server y las diversas instancias de la puerta de enlace.
- **Proxy Netlet.** Hace posible que los usuarios ejecuten de forma segura servicios TCP/IP comunes en Internet y otras redes que no sean seguras. Netlet permite ejecutar aplicaciones como Telnet, SMTP, HTTP y otras de puerto fijo. Netlet permite el acceso y el funcionamiento remotos de los directorios y los sistemas de archivos y garantiza una comunicación segura entre el applet de Netlet en el navegador del cliente, la puerta de enlace y los servidores de aplicaciones.
- **Proxy Rewriter.** Permite el tráfico HTTP seguro entre la puerta de enlace y los equipos de la intranet. Rewriter proporciona un acceso seguro a las páginas web de una intranet corporativa desde el exterior de la intranet mediante la transformación de los vínculos web y la creación de conjuntos de reglas para administrar las páginas web de la intranet.

### Web Proxy Server 4.0.4

Sun Java System Web Proxy Server (Web Proxy Server) proporciona funciones de almacenamiento en caché, filtrado y distribución de contenido web. Web Proxy Server se utiliza

a menudo dentro de servidores de seguridad empresariales para reducir el número de solicitudes a los servidores de contenido remotos. Se utiliza también fuera de los servidores de seguridad para proporcionar una puerta de enlace segura para las solicitudes entrantes de Internet.

El programa de instalación de Java ES incluye Web Proxy Server como único componente instalable.

## Componentes de supervisión

Sun Java System Monitoring Console 1.0 (Monitoring Console) incluye un agente maestro que se conecta a todos los agentes del nodo en la implementación de Java ES. Monitoring Console es compatible con Sun Java System Monitoring Framework 2.0 (Monitoring Framework), un componente compartido que proporciona el agente del nodo y los instrumentos necesarios para que cada uno de los componentes supervisados muestre sus atributos para su comprobación. Cada componente de producto muestra los objetos que representan los atributos que pueden supervisarse y el agente del nodo proporciona una vista de varios componentes en un host. Para obtener información detallada sobre la supervisión, consulte la *Guía de supervisión de Sun Java Enterprise System 5*.

## Componentes compartidos

Los componentes compartidos proporcionan servicios locales y compatibilidad tecnológica que requieren los componentes de servicios de sistema de Java ES para funcionar. Estos componentes son bibliotecas locales que pueden compartir todos los componentes de Java ES que se estén ejecutando en un equipo host concreto. El programa de instalación de Java ES instala automáticamente los componentes compartidos necesarios para los demás componentes de Java ES que estén instalados en un equipo host.

Entre los componentes compartidos de Java ES, se incluyen:

- ACL (Apache Common Logging) 1.0.4
- ANT (Jakarta ANT Java/XML-based build tool) 1.6.5
- BDB (Berkeley Database) 4.2.52
- Common Agent Container 1.1 (sólo para Sun Cluster)
- Common Agent Container 2.0
- FastInfoSet 1.0.2
- ICU 3 (International Components for Unicode) 3.2
- J2SE (Java 2 Platform, Standard Edition) 5.0 Update 6 (la versión 5.0 Update 3 es compatible con HP-UX)
- JAF (JavaBeans™ Activation Framework) 1.0.3

- JATO (Java Studio Web Application Framework) 2.1.5
- JavaHelp™ 2.0
- JavaMail™ API 1.3.2
- JAXB (Java Architecture for XML Binding) 2.0.3
- JAXP (Java API for XML Processing) 1.3.1
- JAXR (Java API for XML Registries) 1.0.8
- JAXRPC (Java API for XML-based Remote Procedure Call) 1.1.3\_01
- JAXWS (Java API for Web Services) 2.0
- JDMK (Java Dynamic Management Kit) 5.1.2
- JSS (Java Security Services) 4.2.4
- JSS3 (Network Security Services for Java) 3.1.11
- JSTL (JavaServer Pages™ Standard Tag Library) 1.0.6
- KTSE (KT Search Engine) 1.3.4
- LDAP C SDK 6.0
- LDAP Java SDK 4.19
- MA Core (Mobile Access Core) 6.3.1
- NSPR (Netscape Portable Runtime) 4.6.3
- NSS (Network Security Services) 3.11
- NSSU (Network Security Service Utilities) 3.11
- SAAJ (SOAP with Attachments API for Java) 1.3
- SASL (Simple Authentication and Security Layer) 2.19
- Sun Explorer Data Collector (sólo para el SO Solaris) 4.3.1
- Sun Java System Monitoring Framework 2.0 (compatible con Monitoring Console 1.0)
- Sun Java Web Console 3.0.2
- WSCL (Web Services Common Library) 2.0
- XWSS (XML Web Services Security) 2.0

# Índice

---

## A

### Access Manager

- como componente de servicios del sistema, 19
- como servicio de infraestructura, 36
- descripción de, 70

### Apache Derby, 19

### aplicaciones

#### de empresa

- Ver* aplicaciones de empresa distribuidas

#### distribuidas

- Ver* aplicaciones de empresa distribuidas

### aplicaciones de empresa distribuidas

- acerca de, 15
- definición, 29
- infraestructura para, 16

### Application Platform Suite, 22

### Application Server

- como componente de servicios del sistema, 19
- como servicio de infraestructura, 36
- descripción de, 71

### arquitectura

- definición, 48
- dimensiones
  - Ver* dimensiones arquitectónicas
- implementación, 62-64
- introducción, 31
- solución, 32

### arquitecturas de implementación

- definición, 49
- diseño de, 62-64
- introducción, 31
- relación con las arquitecturas en capas, 40

- arquitecturas de implementación de referencia,
  - definición, 68

### arquitecturas lógicas

- definición, 49
- ejemplo, 46-48
- introducción, 31
- niveles de servicio de infraestructura, 33

### autenticación, 55-56

### autorización, 57

### Availability Suite, 22

## C

### Calendar Server, 23

### capas, lógicas

- arquitectura de aplicación, 38
- cliente, 39
- datos, 39
- presentación, 39
- servicios de negocio, 39

### casos de uso

- definición, 68
- introducción, 61

### categorías de usuario

- analista de sistemas, 26
- planificador de negocios, 26

### categorías de usuarios

- administrador de sistemas, 27
- administrador de sistemas especializado, 27
- administrador de TI, 27
- administrador delegado, 27

categorías de usuarios (*Continuación*)

- arquitecto, 26
- ingeniero de campo, 27
- integrador de sistemas, 27

## clientes

- componentes de servicios del sistema y, 18
- definición, 49

## clústeres

- Ver* Sun Cluster

## Communications Express, 23

## Communications Suite, 17, 23

## componentes

- acceso, 21
- calidad del servicio, 20-21, 73-77
- compartidos, 21, 77-78
- comunicaciones, 23
- definición, 29
- dependencias, 37-38
- detección de versiones instaladas, 52
- disponibilidad, 20-21
- distribuidos, 15
- EJB, 39
- J2EE, 39
- JSP, 39
- producto, 30
- servicio del sistema, 18-20
- servicios del sistema, 69-73
- servlet, 39
- sistema

- Ver* componentes del sistema

- supervisión, 21
- y servicios de infraestructura, 36

## componentes compartidos, 77

- componentes compartidos, 21
- definición, 30
- introducción, 21

## componentes de acceso

- descripciones, 76-77
- introducción, 21

## componentes de aplicación

- arquitecturas de capas lógicas, 38
- definición, 48

## componentes de calidad del servicio

- definición, 30

componentes de calidad del servicio (*Continuación*)

- descripciones, 73-77
- introducción, 20-21
- componentes de comunicación, 17
- componentes de comunicaciones, 23, 41
- componentes de disponibilidad
  - descripciones, 74-76
  - introducción, 20-21
- componentes de Java Servlet, 39
- componentes de JSP, 39
- componentes de los servicios del sistema
  - dependencias, 37-38
  - introducción, 18-20
- componentes de productos, definición, 30
- componentes de servicios del sistema, definición, 30
- componentes del sistema
  - acerca de, 18
  - componentes compartidos, 77-78
  - componentes de calidad del servicio, 20-21
  - componentes de los servicios del sistema
    - Ver* componentes de los servicios del sistema
    - definición, 30
    - servicios del sistema, 69-73
- componentes EJB, 39
- comprobación de dependencias, programa de instalación, 52
- conjuntos, 22

**D**

## Delegated Administrator, 23

## dependencias, 37-38, 52

## Derby Database, 19

## desarrollo

- como tarea de implementación, 65
- definición, 68

## detección del software instalado, 52

## dimensiones arquitectónicas

- calidad del servicio, 41-45
- capas lógicas, 38
- dependencias de los servicios de infraestructura, 33
- síntesis, 45

## directivas

- autorización, 57

directivas (*Continuación*)

definición, 58

directorios

definición, 57

en forma de almacenes de datos de usuarios, 54

Directory Server

como componente de servicios del sistema, 19

como servicio de infraestructura, 36

descripción de, 71

disponibilidad

requisitos, 42, 43-44

servicios, 17, 44, 74

distribuidas

aplicaciones

*Ver* aplicaciones de empresa distribuidas

distribuidos

servicios

*Ver* servicios distribuidos

documentación de Java ES, 12

## E

entrada de usuario, 54

escalabilidad

requisitos, 42, 43-44

servicios, 44, 74

escenarios de adopción, Java ES

acerca de, 27-29

actualización, 28

ampliación, 27

definición, 29

mejora, 27

sistema nuevo, 27

escenarios de implementación

definición, 67

introducción, 61

especificaciones de implementación, 64

etapa posterior a la implementación

definición, 68

etapas del ciclo de vida, 67

etapa previa a la implementación

definición, 68

etapas del ciclo de vida, 61

etapas del ciclo de vida

etapa posterior a la implementación, 26, 67

etapa previa a la implementación, 26, 61

implementación, 26, 62

## F

formación, escenarios de adopción de Java ES, 28

funciones de integración

componentes compartidos, 17

identidad y seguridad, 17, 54-57

instalador integrado, 17, 51-53

## G

glosario, vínculo a, 13

## H

hardware, escenarios de adopción de Java ES, 28

High Availability Session Store

como componente de calidad del servicio, 20

descripción, 74

## I

identidad

administración, 54

servicios, 17, 54-57

único usuario, 54-55

identidad única

definición, 58

introducción, 54

Identity Management Suite, 23

implementación

arquitectura, 62

definición, 67

desarrollo y personalización, 65

despliegue, 64-67

diseño, 62-64

implementación (*Continuación*)

escenarios

*Ver* escenarios de implementación

etapa del ciclo de vida, 62-67

etapas del ciclo de vida, 62

presentación de producción, 66-67

prueba de prototipos, 66

## infraestructura

dependencias de servicios

*Ver* servicios distribuidos

para aplicaciones de empresa distribuidas, 16

## inicio de sesión único

definición, 58

despliegue, 56

función de Java ES, 19, 55

niveles de los servicios de infraestructura, 35

## Instant Messaging, 23

## integración

escenarios de adopción de Java ES, 28

servicios, 35

**J**

## J2EE

componentes, 39

modelo de componentes distribuidos, 39

plataforma, 20

## Java DB

como componente de servicios del sistema, 19

descripción de, 72

## JMS (Java Message Service), 19

## JSS (Java Security Services), 21

**L**

## LDAP, 39, 57

**M**mejoras, *Ver* escenarios de adopción

## Message Queue

como componente de servicios del sistema, 19

Message Queue (*Continuación*)

como servicio de infraestructura, 36

descripción, 72

## Messaging Server, 23

migración, escenarios de adopción de Java ES, 28

## Monitoring Console, 21

descripción de, 77

en conjuntos, 22-24

**N**

## NSPR (Netscape Portable Runtime), 21

## NSS (Network Security Services), 21

**P**

perfiles de usuarios, 26

plataforma J2ME, 39

## Portal Server

como componente de servicios del sistema, 19

como servicio de infraestructura, 36

descripción, 72

## Portal Server Secure Remote Access

como componente de calidad del servicio, 21

como componente del sistema, 43

descripción, 76

presentación de producción, 66-67

## productos de Sun Java System

Access Manager

*Ver* Access Manager

Application Server

*Ver* Application Server

Directory Server

*Ver* Directory Server

High Availability Session Store

*Ver* High Availability Session Store

Java DB

*Ver* Java DB

Message Queue

*Ver* Message Queue

Portal Server

*Ver* Portal Server

productos de Sun Java System (*Continuación*)

- Portal Server, Secure Remote Access
  - Ver* Portal Server, Secure Remote Access
- Service Registry
  - Ver* Service Registry
- Sun Cluster
  - Ver* Sun Cluster
- Sun Cluster Geographic Edition
  - Ver* Sun Cluster Geographic Edition
- Web Proxy Server
  - Ver* Web Proxy Server
- Web Server
  - Ver* Web Server

programa de desinstalación, 53

prototipo, 66

provisión de usuarios, 64

## R

requisitos de calidad del servicio

- capacidad latente, 42
- disponibilidad, 42, 43-44
- escalabilidad, 42, 43-44
- facilidad de mantenimiento, 42, 43-44
- rendimiento, 42, 43-44
- seguridad, 43-44

requisitos de capacidad latente, 42

requisitos de facilidad de mantenimiento, 42, 43-44

requisitos de la calidad del servicio, seguridad, 42

requisitos de rendimiento, 42, 43-44

## S

seguridad

- requisitos, 42, 43-44
- servicios, 17
- servicios de directivas, 35

Service Registry

- como componente de servicios del sistema, 19
- descripción de, 73

servicios

- alta disponibilidad, 44, 74
- definición, 29

servicios (*Continuación*)

- escalabilidad, 44, 74
- infraestructura, 16
  - Ver* servicios de infraestructura distribuidos
- supervisión, 17
- web, 39

servicios de acceso, 17

servicios de aplicación, 17, 33

servicios de colaboración, 16

servicios de colaboración de usuarios, 35

servicios de comunicación, 16

servicios de mensajería, 34

servicios de negocio, capa de presentación, 39

servicios de negocios, definición, 48

servicios de nivel intermedio, 33

servicios de persistencia, 34

servicios de plataforma, 33

servicios de portal, 16

servicios de sistema operativo, 34

servicios de tiempo de ejecución, 35

servicios de transporte de red, 34

servicios del sistema

- acerca de, 18
- definición, 30

servicios distribuidos

- acceso, 17
- comunicación y colaboración, 16
- descripción general, 16
- disponibilidad, 17
- identidad, 17
- infraestructura, 16
- integración, 35
- mensajería, 34
- nivel de aplicación, 33
- nivel intermedio, 33
- persistencia, 34
- plataforma, 33, 34
- portal, 16
- seguridad, 17, 35
- supervisión, 17
- tiempo de ejecución, 17
- transporte de red, 34
- web, 17

servicios web, 17

- servicios web (*Continuación*)
  - componentes de J2EE, 39
  - definición, 49
- servidores
  - definición, 49
  - independientes, 39
- sistema
  - componentes
    - Ver* componentes del sistema
  - configuración, 52-53
  - servicios, 15-17
- soluciones, Java ES
  - arquitectura, 31
  - ciclo de vida, 24-27
  - ejemplo, 46
  - personalizadas y de fábrica, 45
- Sun Cluster
  - agentes, 44
  - como componente de calidad del servicio, 20
  - como servicio de disponibilidad, 44-45
  - descripción, 74
- Sun Cluster Geographic Edition
  - como componente de calidad del servicio, 20
  - descripción, 75
- supervisión, acerca de, 21, 77

## T

- tareas, Java Enterprise System, 59
- tareas, Java ES, 24

## U

- usuarios, provisión, 64
- usuarios finales
  - aplicaciones distribuidas y, 15
  - definición, 29

## W

- Web Infrastructure Suite, 24