



# Visión general técnica de Sun Java Enterprise System 2005Q4

---

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

Referencia: 819-3589  
Octubre de 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Reservados todos los derechos.

Este producto o documento está protegido por leyes de copyright y distribuido mediante licencia que restringe su uso, copia, distribución y descompilación. Ninguna parte de este producto ni de su documentación puede ser reproducido de ninguna forma y por ningún medio sin la autorización previa por escrito de Sun y sus licenciatarios, si los hubiera. El software de terceras partes, incluyendo la tecnología de fuentes, está protegido por leyes de copyright y se concede bajo licencia por parte de los proveedores de Sun.

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, docs.sun.com, AnswerBook, AnswerBook2, y Solaris son marcas comerciales o marcas comerciales registradas de Sun Microsystems, Inc. en Estados Unidos y en otros países. Todas las marcas comerciales de SPARC se utilizan bajo licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. en EE.UU. y en otros países. Los productos que llevan la marca comercial SPARC están basados en la arquitectura desarrollada por Sun Microsystems, Inc.

La interfaz gráfica de usuario de OPEN LOOK y Sun™ 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 es extensiva a los licenciatarios de Sun que implementen la interfaz gráfica de usuario OPEN LOOK y que actúen conforme a los acuerdos de licencia por escrito de Sun.

Derechos del gobierno de los 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 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.

---

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



051220@13215



# Contenido

---

<b>Prefacio</b>	<b>13</b>
<b>1 Introducción a Java Enterprise System</b>	<b>19</b>
¿Por qué usted necesita Java Enterprise System?	19
Componentes de Java Enterprise System	21
Componentes de los servicios del sistema	22
Componentes de calidad del servicio	24
Componentes compartidos	26
Uso de Java Enterprise System	27
Ciclo de vida de la solución Java Enterprise System	27
Escenarios de adopción de Java Enterprise System	30
Términos clave de este capítulo	32
<b>2 Arquitecturas de las soluciones de Java Enterprise System</b>	<b>35</b>
Marco arquitectónico de Java Enterprise System	35
Dimensión 1: dependencias de los servicios de infraestructuras	37
Dimensión 2: capas lógicas	42
Dimensión 3: calidad del servicio	46
Síntesis de las tres dimensiones arquitectónicas	50
Ejemplo de arquitectura de solución de Java Enterprise System	51
Escenario de comunicaciones de las empresas	51
Arquitectura lógica para el escenario de ejemplo	52
Arquitectura de implementación para el escenario de ejemplo	54
Términos clave de este capítulo	54

<b>3</b>	<b>Funciones de integración de Java Enterprise System</b>	<b>57</b>
	El instalador integrado de Java Enterprise System	57
	Comprobación del software previo	58
	Comprobación de relaciones de dependencia	58
	Configuración inicial	59
	Desinstalación	59
	Identidad integrada y servicios de seguridad	59
	Identidad única	60
	Autenticación e inicio de sesión único	61
	Autorización	63
	Términos clave de este capítulo	64
<b>4</b>	<b>Tareas del ciclo de vida de la solución Java Enterprise System</b>	<b>65</b>
	Tareas del ciclo de vida de las soluciones	65
	Implementación previa	67
	Implementación	68
	Diseño de implementación	69
	Despliegue de implementación	71
	Implementación posterior	73
	Términos clave de este capítulo	74
<b>5</b>	<b>Lista de referencia: componentes de Java Enterprise System</b>	<b>77</b>
	Descripciones de los componentes de servicios de sistemas	78
	Sun Java System Access Manager 7 2005Q4	78
	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	79
	Sun Java System Calendar Server 6 2005Q4	79
	Sun Java System Directory Server 5 2005Q4	80
	Sun Java System Instant Messaging 7 2005Q4	80
	Sun Java System Message Queue 3.6 2005Q4	80
	Sun Java System Messaging Server 6 2005Q4	81
	Sun Java System Portal Server 6 2005Q4	82
	Sun Java System Service Registry 3 2005Q4	82
	Sun Java System Web Server 6.1 2005Q4	82
	Descripciones de componentes de calidad del servicio	83
	Descripciones de los componentes de disponibilidad	83
	Descripciones de componentes de acceso	84
	Descripciones de componentes administrativos	86

Componentes compartidos 87

**Índice 89**



# Lista de tablas

---

<b>TABLA 1-1</b>	Componentes de los servicios del sistema de Java ES	23
<b>TABLA 1-2</b>	Componentes de disponibilidad de Java ES	25
<b>TABLA 1-3</b>	Componentes de acceso de Java ES	25
<b>TABLA 1-4</b>	Componentes administrativos de Java ES	26
<b>TABLA 1-5</b>	Categorías de usuarios para las tareas del ciclo de vida de Java ES	29
<b>TABLA 1-6</b>	Consideraciones sobre los escenarios de adopción de Java ES	32
<b>TABLA 2-1</b>	Relaciones entre los componentes de los servicios del sistema de Java ES	41
<b>TABLA 2-2</b>	Calidades de servicio que afectan a la arquitectura de solución	47
<b>TABLA 2-3</b>	Componentes de calidad de servicio y calidades de sistema afectadas	48
<b>TABLA 2-4</b>	Resumen de requisitos de negocio: escenario de comunicaciones	52



# Lista de figuras

---

<b>FIGURA 1-1</b>	Compatibilidad necesaria para aplicaciones de empresa distribuidas 20
<b>FIGURA 1-2</b>	Categorías de los componentes de Java ES 22
<b>FIGURA 1-3</b>	Etapas del ciclo de vida de la solución y categorías de usuarios 28
<b>FIGURA 2-1</b>	Dimensiones de la arquitectura de las soluciones de Java ES 36
<b>FIGURA 2-2</b>	Dimensión 1: niveles de los servicios de infraestructuras 38
<b>FIGURA 2-3</b>	Componentes de los servicios del sistema de Java ES 40
<b>FIGURA 2-4</b>	Dimensión 2: capas lógicas para aplicaciones de empresa distribuidas 43
<b>FIGURA 2-5</b>	Messaging Server: ejemplo de arquitectura en capas 45
<b>FIGURA 2-6</b>	Diseño de disponibilidad usando nodos de Sun Cluster 49
<b>FIGURA 2-7</b>	Arquitectura lógica para el escenario de comunicaciones de la empresa 53
<b>FIGURA 3-1</b>	La entrada única de usuario es compatible con varios servicios 61
<b>FIGURA 3-2</b>	Secuencia de autenticación 62
<b>FIGURA 3-3</b>	Secuencia de autorización 63
<b>FIGURA 4-1</b>	Tareas del ciclo de vida de las soluciones 66
<b>FIGURA 4-2</b>	Especificación de un escenario de implementación 68
<b>FIGURA 4-3</b>	Conversión de un escenario de implementación en una arquitectura de implementación 69



## Lista de ejemplos

---



# Prefacio

---

*Java Enterprise System Visión general técnica* presenta las nociones técnicas y conceptuales de Java Enterprise System. También describe los componentes, la arquitectura, los procesos y las funciones de Java Enterprise System.

Esta visión general intenta aclarar los conceptos y la terminología utilizada en la documentación de Java Enterprise System. Los términos técnicos clave se explican en la sección “Términos clave” de cada capítulo, que clarifica cómo se utilizan estos términos en el contexto de Java Enterprise System.

---

## Usuarios a los que está destinada esta guía

*Java Enterprise System Visión general técnica* está destinada a usuarios que deseen diseñar, implementar o mantener soluciones de software basadas en Java Enterprise System. Esto constituye un gran público, que incluye analistas de negocios, arquitectos de sistemas, ingenieros de campo y administradores de sistemas.

Los usuarios que utilicen la *Java Enterprise System Visión general técnica* 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 Java y los componentes de Java 2 Standard Edition y Java 2 Enterprise Edition

---

# Conjunto de documentación de Java Enterprise System

El conjunto de documentación de Java ES describe el procedimiento de planificación de la instalación y el de instalación del sistema. La dirección URL para la documentación del sistema es <http://docs.sun.com/coll/1286.1>. Para ver una introducción a Java ES, consulte los manuales en el orden en que figuran en la siguiente tabla.

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

Título del documento	Contenido
<i>Notas de la versión de Sun Java Enterprise System 2005Q4</i>	Contiene la información más actualizada acerca de Java ES, incluidos los problemas conocidos. Además, cada componente tiene sus propias notas de la versión.
<i>Guía de documentación de Sun Java Enterprise System 2005Q4</i>	Proporciona descripciones de toda la documentación relacionada con Java ES, tanto para los componentes de sistema como para los componentes individuales.
<i>Visión general técnica de Sun Java Enterprise System 2005Q4</i>	Presenta los fundamentos técnicos y conceptuales de Java ES. Describe los componentes, la arquitectura, los procesos y las funciones.
<i>Guía de planificación de la implementación de Sun Java Enterprise System 2005Q4</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 2005Q4</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 2005Q4 para UNIX</i>	Le orienta a través de proceso de instalación de Java ES en los sistemas operativos Solaris o Linux. También muestra cómo se configuran los componentes después de instalarlos y cómo se comprueba si funcionan correctamente.

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

Título del documento	Contenido
<i>Referencia de la instalación de Java ES System 2005Q4</i>	Incluye información adicional acerca de los parámetros de configuración, proporciona hojas de trabajo para usarlas durante la planificación de la configuración e incluye listas de material de referencia como, por ejemplo, los directorios y los números de puerto predeterminados.
<i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i>	Describe cómo se instala Java ES en un sistema, cómo se establece un conjunto de servicios principal, compartido y de red y cómo se configuran las cuentas de usuario para que accedan a los servicios establecidos.
<i>Guía de actualización de Sun Java Enterprise System 2005Q4</i>	Proporciona instrucciones para actualizar Java ES en el sistema operativo Solaris o en el entorno operativo Linux.
<i>Sun Java Enterprise System Glossary</i>	Define los términos usados en la documentación de Java ES.

## Convenciones tipográficas

En la siguiente tabla se describen las convenciones tipográficas usadas en esta guía.

**TABLA P-2** Convenciones tipográficas

Tipos de letra	Significado	Ejemplo
<i>AaBbCc123</i>	Los nombres de los comandos, los archivos, los directorios y el texto que se muestra en la pantalla del equipo	Edite el archivo <code>.login</code> . Utilice <code>ls -a</code> para ver todos los archivos. <code>machine_name% you have mail.</code>
<b>AaBbCc123</b>	Lo que escribe el usuario, en contraposición a lo que el equipo muestra en pantalla.	<code>machine_name% su</code> Password:
<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> .

**TABLA P-2** Convenciones tipográficas (Continuación)

Tipos de letra	Significado	Ejemplo
<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>User's Guide</i> . Un elemento <i>almacenado en caché</i> es una copia que se almacena localmente. <i>No</i> guarde el archivo.

---

## Acceso a los recursos de Sun en línea

El sitio web docs.sun.com<sup>SM</sup> le permite acceder a la documentación técnica en línea de Sun. Puede desplazarse por docs.sun.com o buscar un tema o un título de manual específico. Los manuales están disponibles como archivos en línea con los formatos PDF y HTML. Los dos formatos se pueden leer mediante tecnologías de asistencia para usuarios con discapacidades.

Para acceder a los siguientes recursos de Sun, acceda a <http://www.sun.com>:

- Descargas de productos de Sun
- Servicios y soluciones
- Asistencia (incluidas las revisiones y las actualizaciones)
- Formación
- Investigación
- Comunidades (por ejemplo, Sun Developer Network)

---

## Referencias a sitios web de terceros

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 respalda ni se hace responsable de ningún contenido, anuncio, producto o 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. Para publicar sus comentarios, acceda a <http://docs.sun.com> y haga clic en Send Comments (Enviar comentarios). 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 819-3589.



# 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 se denominan en esta guía como aplicaciones de empresas distribuidas.

Java Enterprise System también es una versión de software de Sun, además de una metodología de envío y una estrategia empresarial y de asignación de precios. El planteamiento de esta guía, sin embargo, está centrado en los componentes de software de Java Enterprise System y los servicios que proporcionan.

Este capítulo presenta Java Enterprise System y las tareas que conlleva la utilización del sistema. Incluye los siguientes temas:

- “¿Por qué usted necesita Java Enterprise System?” en la página 19
- “Componentes de Java Enterprise System” en la página 21
- “Uso de Java Enterprise System” en la página 27
- “Términos clave de este capítulo” en la página 32

---

## ¿Por qué usted necesita Java Enterprise System?

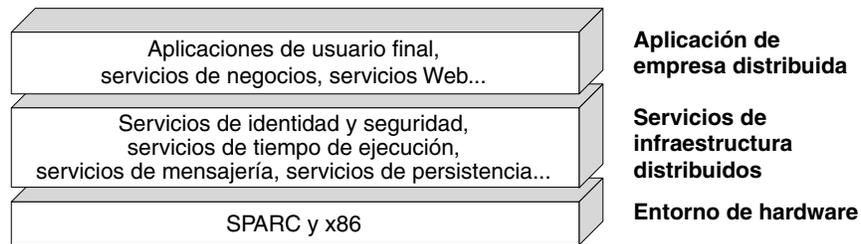
En la actualidad, los requisitos empresariales demandan soluciones de software distribuidas en la red o en entornos de Internet y que tengan altos niveles de rendimiento, disponibilidad, seguridad, escalabilidad y facilidad de mantenimiento.

Java Enterprise System proporciona los servicios de infraestructura necesarios para estas **aplicaciones de empresa distribuida**, es decir, las aplicaciones que generalmente tienen las siguientes características:

- **Distribuidas.** 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 están, a su vez, admitidos por 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 Enterprise System proporciona la capa de servicios de infraestructura distribuida que se muestra en la **Figura 1-1**. Sin embargo, los servicios de Java Enterprise System también incluyen una serie de servicios de nivel de aplicaciones accesibles para los usuarios finales.



**FIGURA 1-1** Compatibilidad necesaria para aplicaciones de empresa distribuidas

Entre los servicios que incluye Java Enterprise System se encuentran 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, a la vez 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.
- **Identidad de red y servicios de seguridad.** Estos servicios mejoran la seguridad y la protección de los activos 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 están basados en la tecnología Java 2 Platform, Enterprise Edition (J2EE™).

Java Enterprise System 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 que incluye Java Enterprise System se encuentran las siguientes:

- **Servicios de disponibilidad.** Estos servicios proporcionan una disponibilidad prácticamente continua, también para los componentes de aplicaciones y para los componentes de infraestructura que los respaldan.
- **Servicios de acceso.** Estos servicios proporcionan acceso de Internet o acceso basado en navegador a los servicios de Java Enterprise System.
- **Servicios administrativos.** Estos servicios ayudan a mantener y ajustar el rendimiento de las aplicaciones respaldadas por Java Enterprise System.

Puede implementar uno o varios servicios de Java Enterprise System, cada uno de los cuales puede incluir un número de componentes de Java Enterprise System.

---

## Componentes de Java Enterprise System

Java Enterprise System es una integración de distintos productos de software y componentes en un único sistema de software. Los componentes de este sistema se han probado conjuntamente para garantizar su interoperabilidad. La integración es posible mediante una serie de funciones de nivel de sistema:

- 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.

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

- **Componentes de los servicios del sistema.** Estos componentes proporcionan los servicios de infraestructura principales de Java Enterprise System que admiten el uso de aplicaciones de empresa distribuidas.
- **Componentes de calidad del servicio.** Estos componentes 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.

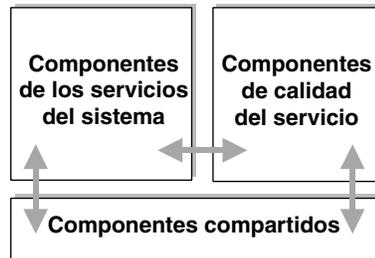


FIGURA 1-2 Categorías de los componentes de Java ES

## Componentes de los servicios del sistema

Una serie de componentes de Java ES proporcionan los servicios principales que dan respaldo a las soluciones de software distribuidas. Estos [servicios de sistema](#) incluyen servicios de portal, servicios de comunicación y colaboración, servicios de identidad y seguridad, servicios de contenedor web y servicios de aplicación J2EE.

Los [componentes de servicio de sistema](#) que proporcionan estos servicios distribuidos y los servicios que éstos proporcionan se describen brevemente en la siguiente tabla. Cada componente de los 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 [“Descripciones de los componentes de servicios de sistemas”](#) en la página 78.

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

<b>Componente</b>	<b>Servicios de sistema proporcionados</b>
Sun Java System Web Server	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	Proporciona servicios de programación y de calendario a los usuarios finales individuales y a los grupos de usuarios finales. Calendar Server incluye un cliente basado en navegador que interactúa con el servidor.
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.
Sun Java System Instant Messaging	Proporciona comunicación segura y en tiempo real entre usuarios finales, como, por ejemplo, mensajería instantánea (chat), conferencias, alertas, noticias, encuestas y transferencias de archivos. El servicio incluye un administrador de presencia que indica a los usuarios quién está actualmente en línea e incluye un cliente basado en explorador que interactúa con el servidor.
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 Messaging Server	Incluye mensajería de almacenamiento y reenvío segura, fiable y de alta capacidad compatible con correo electrónico, fax, buscapersonas, voz y vídeo. Messaging Server puede obtener acceso simultáneamente a varios almacenes de mensajes e incluye filtrado de contenido para ayudar a rechazar correo electrónico no solicitado y evitar ataques de virus.

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

Componente	Servicios de sistema proporcionados
Sun Java System Web 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 Web Server	Proporciona servicios de contenedor web 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 aportar contenido web dinámico y estático, como, por ejemplo, secuencias de comandos CGI y páginas Active Server.
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 industriales para registrar y detectar servicios web, así como para administrar los hechos y los fragmentos de información asociada como, por ejemplo, XMLSchema, las reglas de los procesos de negocio, el control de acceso, el control de la versión, etc.

## Componentes de calidad del servicio

Además de los componentes de los servicios del sistema que se muestran en la [Tabla 1-1](#), Java Enterprise System incluye una serie de componentes que se usan para mejorar la calidad de los servicios proporcionados por los componentes de los 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 administrativos

## 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 Enterprise System y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más detalles acerca de cualquier componente, consulte [“Descripciones de los componentes de disponibilidad”](#) en la [página 83](#).

**TABLA 1-2** Componentes de disponibilidad de Java ES

Componente	Servicios de disponibilidad proporcionados
Sun Cluster	Proporciona servicios de escalabilidad y de alta disponibilidad para Java Enterprise System, las aplicaciones que se ejecutan en la parte superior de la infraestructura de Java Enterprise System y el entorno de hardware donde se implementan.
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.

## 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. Los componentes de acceso incluidos en Java Enterprise System y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más detalles acerca de los componentes, consulte [“Descripciones de componentes de acceso” en la página 84](#).

**TABLA 1-3** Componentes de acceso de Java ES

Componente	Servicios de acceso proporcionados
Sun Java System Directory Proxy Server	Proporciona servicios de seguridad para Directory Server desde fuera de un servidor de seguridad corporativo. Directory Proxy Server proporciona control de acceso al directorio y redireccionamiento para varias instancias de Directory Server.
Sun Java System Web Proxy Server	Proporciona funciones de almacenamiento en caché, filtrado y distribución de contenido web para peticiones de Internet entrantes y salientes.
Sun Java System Communications Express	Proporciona acceso basado en web a Messaging Server, Calendar Server y Directory Server, en función de la configuración.
Sun Java System Portal Server 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 Connector para Microsoft Outlook	Proporciona clientes de escritorio utilizando Microsoft Outlook con una interfaz para Messaging Server y Calendar Server.

## Componentes administrativos

Los componentes administrativos proporcionan funciones de administración, como configuración y supervisión, para los servicios del sistema. Los componentes administrativos incluidos en Java Enterprise System y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más detalles acerca de cualquier componente, consulte [“Descripciones de componentes administrativos” en la página 86](#).

**TABLA 1-4** Componentes administrativos de Java ES

Componente	Servicios administrativos proporcionados
Sun Java System Administration Server (y la consola)	Proporciona una herramienta de administración gráfica que le permite configurar y administrar Directory Server y Messaging Server.
Sun Java System Directory Preparation Tool	Proporciona una secuencia de comandos para configurar Directory Server con el esquema necesario para la provisión de los usuarios para Messaging Server y Calendar Server.
Sun Java System Delegated Administrator	Proporciona tanto herramientas de línea de comandos como de GUI para cumplimentar las entradas de los usuarios en Directory Server con los atributos de usuario requeridos por Messaging Server y Calendar Server.

## Componentes compartidos

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

Los componentes compartidos se utilizan a menudo para proporcionar portabilidad entre los distintos sistemas operativos. Ejemplos de componentes compartidos de Java Enterprise System: Java 2 Platform, Standard Edition (plataforma J2SE™), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Network Security Services for Java (JSS), etc. Para ver la lista completa, consulte [“Componentes compartidos” en la página 87](#).

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.

---

## Uso de Java Enterprise System

La creación de soluciones empresariales basadas en el software de Java Enterprise System 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 Enterprise System y de la naturaleza de la solución que desee crear e implementar.

En esta sección se describen dos aspectos del trabajo con Java Enterprise System: el ciclo de vida de la solución de Java Enterprise System y los distintos escenarios de adopción generalmente implicados.

### Ciclo de vida de la solución Java Enterprise System

Las tareas que se usan para crear soluciones de negocio basadas en el software de Java ES se pueden dividir en varias fases, tal y como se muestra en la [Figura 1-3](#). La ilustración también muestra la categoría del usuario de Java Enterprise System que generalmente realiza las distintas tareas.

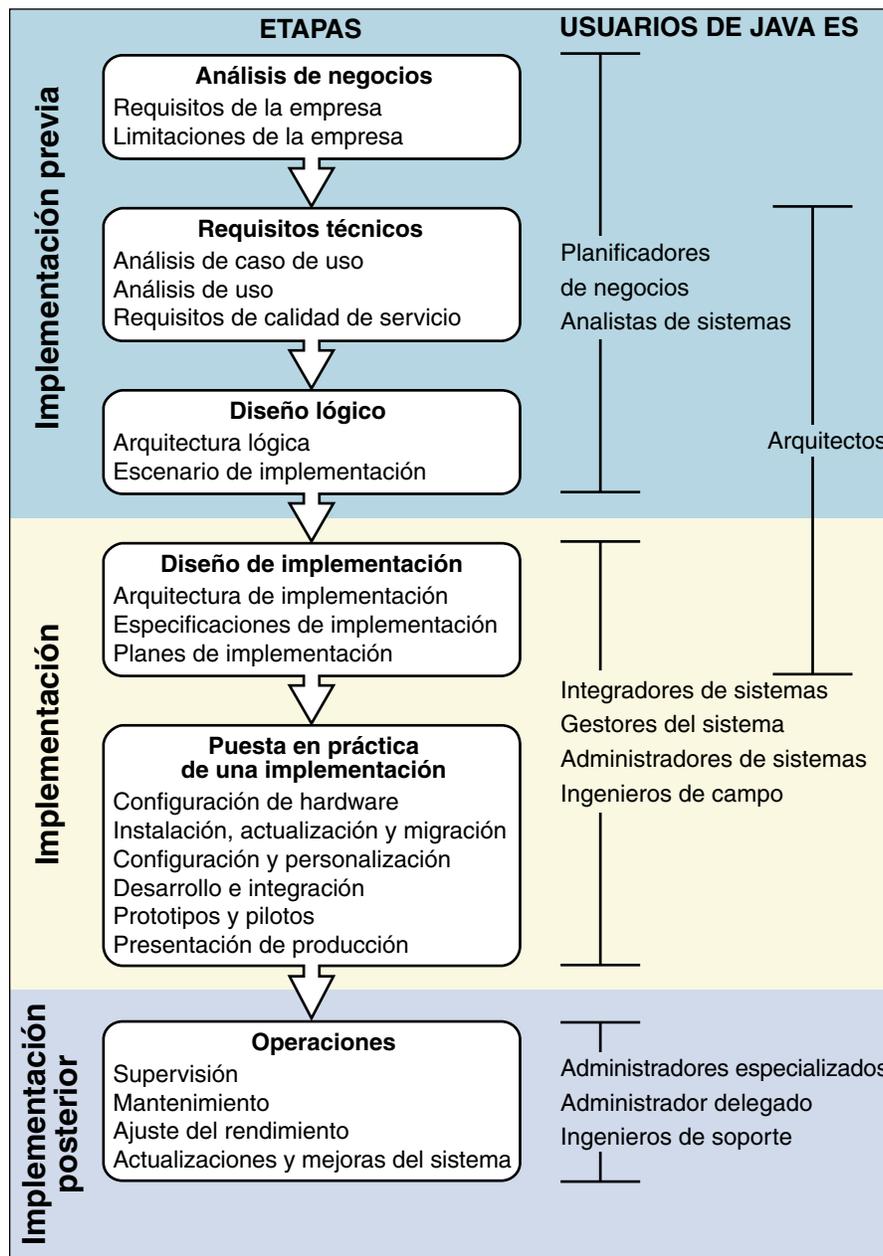


FIGURA 1-3 Etapas del ciclo de vida de la solución y categorías de usuarios

Las etapas del ciclo de vida que se muestran en la [Figura 1-3](#) se pueden estructurar en los siguientes grupos generales:

- **Fase previa a la implementación.** En esta fase, una necesidad de la empresa se traslada a un escenario de implementación: una arquitectura lógica y un conjunto de requisitos de calidad de servicio. El escenario de implementación sirve como especificación para el diseño de la arquitectura de implementación.
- **Implementación.** En esta etapa, un escenario de implementación se transforma 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.
- **Fase posterior a la implementación.** En esta fase, 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.

El ciclo de vida de la solución y las tareas de cada etapa 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. Si trabaja con Java ES, su trabajo se deje ajustar a una o a varias de las categorías de usuarios que se muestran en la [Figura 1-3](#). 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 de negocios 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.	Diseño lógico Diseño de 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 Ingeniero de campo Administrador de sistemas Gestor del sistema	<p>Conocimiento altamente técnico.</p> <p>Está íntimamente familiarizado con los entornos de las tecnologías de la información.</p> <p>Tiene experiencia en la implementación de soluciones de software distribuidas.</p> <p>Conoce la arquitectura, los protocolos, los dispositivos y la seguridad de redes.</p> <p>Conoce los lenguajes de secuencias de comandos y de programación.</p>	<p>Diseño de implementación</p> <p>Implementación de despliegue</p>
Administrador de sistemas especializado Administrador delegado Ingeniero de soporte	<p>Conocimiento técnico especializado o del producto.</p> <p>Familiarizado con hardware, plataformas, directorios y bases de datos.</p> <p>Capacidad de supervisión, solución de problemas y actualización de software.</p> <p>Conoce la administración del sistema para las plataformas del sistema operativo.</p>	Operaciones

## Escenarios de adopción de Java Enterprise System

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 Enterprise System para que se pueda usar una nueva solución de negocio.
- **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 sistemas secundarios porque son demasiado complicados, porque 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. En otras palabras, desea una mejor rentabilidad de la inversión que la que ofrece el sistema existente.
- **Ampliación.** Se parte de una infraestructura de TI existente en la que se implementa el software Java Enterprise System 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 Java para los servicios existentes.

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

Cada escenario de adopción cuenta con sus propios retos y consideraciones. Con independencia del escenario de adopción que se aproxime más a su situación, el proceso del ciclo de vida de la solución que se muestra en la [Figura 1-3](#) sigue siendo aplicable. Sin embargo, en función del escenario de adopción, es posible que varíen los elementos que tendrá que tratar y los recursos en los que necesitará invertir en las distintas etapas del ciclo de vida.

Las siguientes consideraciones se aplican generalmente en varios 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 Enterprise System.
- **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
Nuevo sistema	No hay consideraciones	Es relativamente fácil integrar nuevos componentes	Es normalmente 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
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

<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.

## Términos clave de este capítulo

Esta sección explica los términos técnicos clave utilizados en este capítulo. Se pone un interés especial en clarificar las relaciones entre estos términos y cómo se utilizan en el contexto de Java Enterprise System.

- escenario de adopción** Un motivo general para implementar el software de Java Enterprise System, teniendo en cuenta el sistema de software desde el que se parte y el objetivo que se desea conseguir. Existen cuatro escenarios de adopción básicos para Java Enterprise System: nuevo sistema, sustitución, extensión 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 de sistema](#) incluidos en Java Enterprise System 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 realiza algunas funciones informáticas específicas. Estos componentes, ya sea de forma individual o combinados, proporcionan [servicios de negocio](#) y se pueden encapsular como [servicios web](#).

<b>aplicación de empresa distribuida</b>	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).
<b>usuario final</b>	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 admita una aplicación es un factor importante de la <a href="#">arquitectura de implementación</a> de la aplicación.
<b>servicio</b>	Una función de software realizada para uno o varios <a href="#">clientes</a> . Esta función podría ser de muy bajo nivel, como la administración de memoria, o de alto nivel, como el <a href="#">servicio de negocios</a> 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 calidad del servicio</b>	Uno de los tipos de <a href="#">componentes de sistema</a> incluidos en Java Enterprise System. Los componentes mejoran la disponibilidad, la seguridad, la escalabilidad, la facilidad de mantenimiento y otros aspectos de los componentes de los servicios del sistema y los componentes de aplicaciones distribuidas.
<b>componente compartido</b>	Uno de los tipos de <a href="#">componentes de sistema</a> incluidos en Java Enterprise System. Los componentes compartidos, normalmente bibliotecas, proporcionan servicios locales a otros componentes del sistema. Por contraposición, un <a href="#">componente de los servicios del sistema</a> proporciona servicios de infraestructura a otros componentes de sistema (o a los <a href="#">componentes de aplicación</a> ).
<b>componente del sistema</b>	Cualquier paquete de software o conjunto de paquetes incluidos en Java Enterprise System e instalados mediante el instalador de Java Enterprise System. Existen varios tipos de componentes del sistema: <a href="#">componentes de servicio de sistema</a> que proporcionan <a href="#">servicios de infraestructuras distribuidas</a> ; <a href="#">componentes de servicio de calidad</a> que aportan compatibilidad para los componentes de los servicios del sistema al proporcionar servicios administrativos y de acceso; y, por último, <a href="#">componentes compartidos</a> que proporcionan servicios locales a otros componentes de sistemas.
<b>servicio del sistema</b>	Uno o varios <a href="#">servicios</a> distribuidos que definen la función única que proporciona Java Enterprise System. Los servicios del sistema normalmente requieren la compatibilidad de una serie de <a href="#">componentes de servicio de calidad</a> , de <a href="#">componentes compartidos</a> o puede que requieran ambos.

**componente de los  
servicios del sistema**

Uno de los tipos de [componentes de sistema](#) incluidos en Java Enterprise System. Los componentes de los servicios del sistema proporcionan los servicios de infraestructura principales de Java Enterprise System: servicios de portal, servicios de comunicación y colaboración, servicios de identidad y seguridad, servicios web y de aplicaciones, así como servicios de disponibilidad.

# Arquitecturas de las soluciones de Java Enterprise System

---

Este capítulo proporciona una visión general de los conceptos arquitectónicos en los que se basan las soluciones de Java Enterprise System (Java ES). El capítulo muestra cómo se utilizan los componentes de Java ES, tanto los componentes de los servicios del sistema como los componentes de calidad del servicio, para admitir las 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.

Este capítulo describe un marco arquitectónico para diseñar arquitecturas de soluciones de Java ES, seguido por una arquitectura de solución de ejemplo basada en dicho marco arquitectónico.

En este capítulo se describen los siguientes temas:

- [“Marco arquitectónico de Java Enterprise System”](#) en la página 35
- [“Ejemplo de arquitectura de solución de Java Enterprise System”](#) en la página 51
- [“Términos clave de este capítulo”](#) en la página 54

---

## Marco arquitectónico de Java Enterprise System

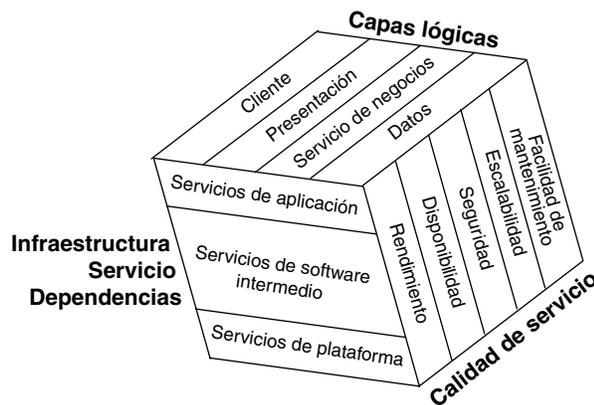
Los componentes de Java ES admiten la implementación de soluciones de software de fortaleza de empresa distribuidas.

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 implicadas en el diseño de soluciones de software de fortaleza de empresa distribuidas. 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 destaca el papel de los componentes de los servicios del sistema (consulte [“Componentes de los servicios del sistema” en la página 22](#)) para que sea posible usar soluciones distribuidas.
- **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 capacidad de mantenimiento. Se incluye también la función de los componentes de calidad del servicio (consulte [“Componentes de calidad del servicio” en la página 24](#)).

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



**FIGURA 2-1** Dimensiones de la arquitectura de las soluciones 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 un conjunto subyacente de servicios de infraestructura 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.

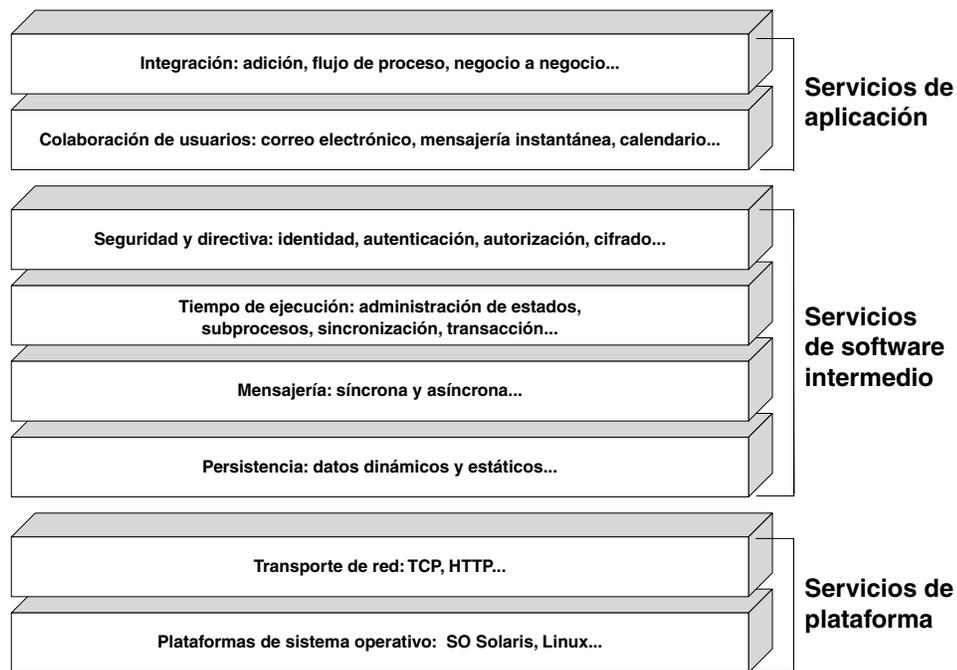
### Niveles de servicio de infraestructura

Al diseñar un sistema de software distribuido, con independencia de que conste principalmente de los componentes desarrollados de forma personalizada o de que incluya todos los componentes "de fábrica" de Java ES, 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 las bases conceptuales para comprender la función de los componentes de los servicios del sistema de Java ES (consulte "[Componentes de los servicios del sistema](#)" en la [página 22](#)).

En general, los servicios que se muestran en la [Figura 2-2](#) se pueden clasificar 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

Los siguientes párrafos describen los distintos niveles de servicio de infraestructura y hacen referencia a artefactos de lenguaje de programación Java, cuando corresponde. Los niveles de servicio se describen desde el nivel inferior al superior, tal y como se muestra en la [Figura 2-2](#):

- **Plataforma de sistema operativo.** Proporciona la compatibilidad básica para cualquier proceso que se ejecute en un equipo. El sistema operativo (como Solaris™, Linux o Microsoft Windows) administra dispositivos físicos, así como la memoria, los subprocesos y otros recursos necesarios para la compatibilidad con Java Virtual Machine (máquina 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.** Compatible con la comunicación síncrona y asíncrona entre componentes de aplicaciones. La mensajería síncrona es el envío y la recepción en tiempo real de mensajes; incluye invocación de método remoto (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.

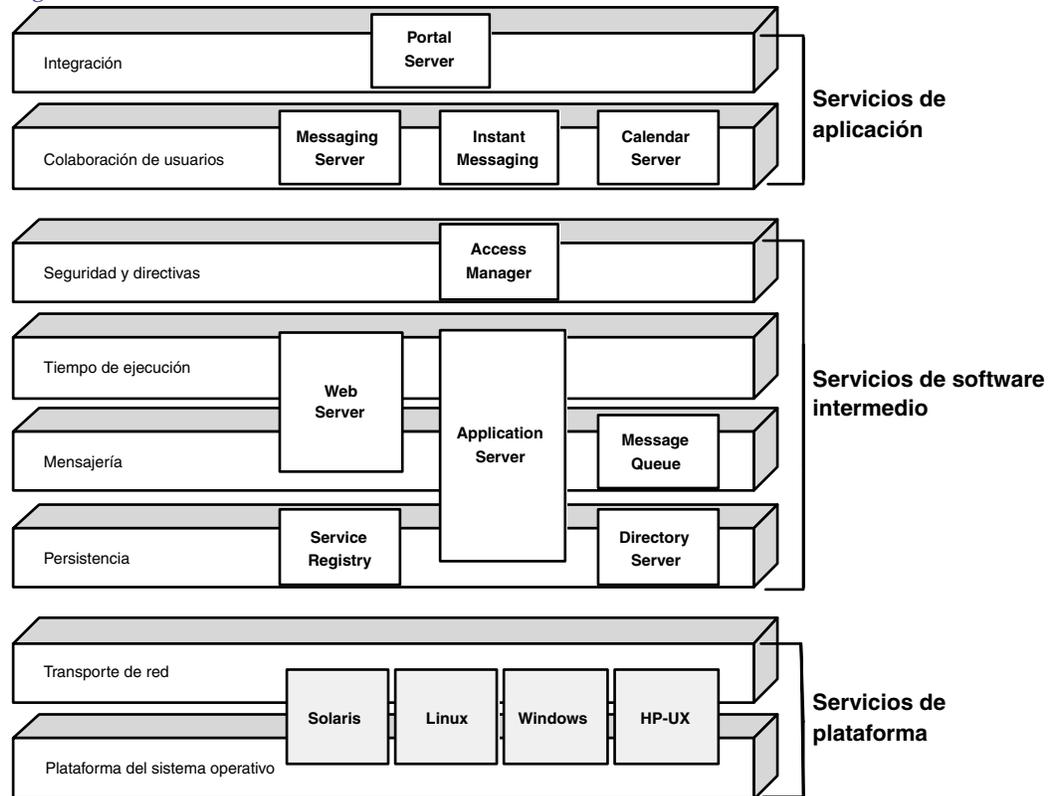
- **Tiempo de ejecución.** Ofrece la asistencia necesaria para cualquier modelo de componente distribuido, como los modelos J2EE o CORBA. Además de la invocación de métodos remotos necesaria para componentes distribuidos y bien acoplados, los servicios de tiempo de ejecución incluyen administración de 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 de EJB™, Web y beans controlados por mensajes en un servidor de aplicación o en un servidor web.
- **Seguridad y directiva.** Proporciona asistencia para acceder de forma segura a los recursos de la aplicación. Estos servicios incluyen compatibilidad para las directivas que rigen el acceso basado en funciones 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.** Incluye servicios que desempeñan un papel importante al permitir la comunicación directa entre usuarios y la colaboración entre usuarios en entornos de Internet y de empresas. Como tales, son servicios de negocios de aplicaciones proporcionados normalmente por servidores independientes (como un servidor de correo electrónico o servidor de calendario).
- **Integración.** Proporciona los servicios que agregan servicios de negocio 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 con 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 general entre los distintos servicios de infraestructura, desde los servicios del sistema operativo de nivel inferior hasta los servicios de aplicación 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 Enterprise System

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 los servicios del sistema de Java ES en los diferentes niveles es la que se muestra en la [Figura 2-3](#).



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

**Nota** – Las plataformas de los sistemas operativos que se muestran en la [Figura 2-3](#) no son una parte formal de Java Enterprise System; sin embargo, se incluyen para mostrar las plataformas de los sistemas operativos en las que se admite el uso de los componentes de Java ES.

## Dependencias de los servicios de infraestructuras de Java Enterprise System

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 [Tabla 2-1](#) se muestran las relaciones específicas entre los componentes de los 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	Respalda el funcionamiento de
Portal Server	Application Server o Web Server Access Manager Directory Server Si se configura para usar los canales de: Calendar Server Messaging Server Instant Messaging	
Messaging Server	Directory Server Access Manager (para el inicio de sesión único)	Calendar Server (para las notificaciones de correo electrónico) Portal Server (para el canal de mensajería)
Instant Messaging	Directory Server Access Manager (para el inicio de sesión único)	Portal Server (para el canal de mensajería instantánea)
Calendar Server	Directory Server Messaging Server (para el servicio de notificación de correo electrónico) Access Manager (para el inicio de sesión único)	Portal Server (para el canal de calendario)
Access Manager	Application Server o Web Server Directory Server	Portal Server Si se configura para el inicio de sesión único: Calendar Server Messaging Server Instant Messaging

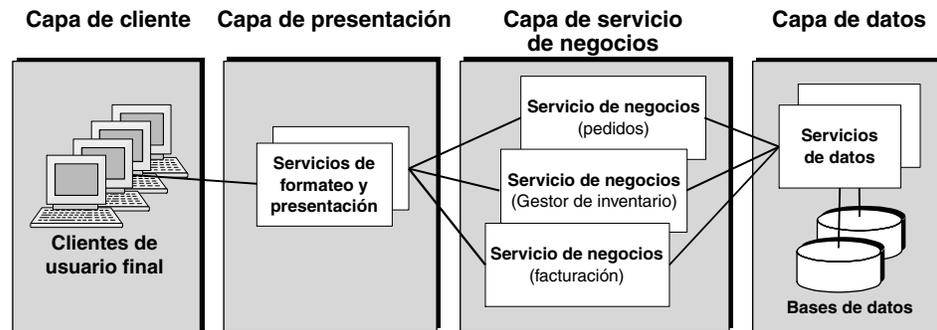
**TABLA 2-1** Relaciones entre los componentes de los servicios del sistema de Java ES  
(Continuación)

Componente	Depende de	Respalda el funcionamiento de
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 Calendar Server Messaging Server Instant Messaging Access Manager
Service Registry	Ninguno	Componentes basados en Application Server

## Dimensión 2: capas lógicas

Los componentes de software que interactúan de las aplicaciones de empresa distribuidas se pueden visualizar como residentes en un número de capas lógicas. Estas capas representan la independencia física y lógica de los componentes de software basándose en 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 37](#) para proporcionar asistencia a los componentes de aplicaciones de todas las capas lógicas que se muestran en la [Figura 2-4](#). Sin embargo, los conceptos de capas lógicas también se aplican a los componentes de los servicios del sistema que proporcionan servicios de nivel de aplicación, tales como Messaging Server y Calendar Server.

## 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 componentes de aplicaciones desplegados usando el modelo del componente Java 2 Platform, Enterprise Edition (plataforma 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 aplicación a la que obtiene acceso directamente un usuario final 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 2 Platform, Micro Edition (plataforma J2ME™) que se ejecuten un dispositivo de mano.
- **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 consiste normalmente en componentes de J2EE como, por ejemplo, los componentes Java Servlet o los componentes 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 negocio.** 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 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 y que se ajuste a los estándares de la interfaz Simple Object Access Protocol (SOAP). Los servicios de negocio 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 consta de servicios que ofrecen 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 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.

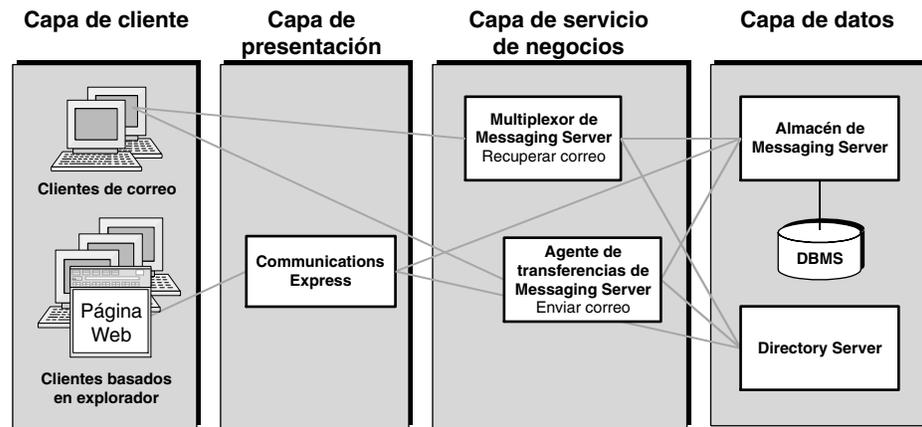
La forma de asignar componentes de aplicación o componentes de infraestructura a un entorno de hardware (es decir, la arquitectura de implementación) depende de muchos factores, en función de la escala y la complejidad de la solución de

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 la infraestructura de Java ES proporcionan la infraestructura subyacente que permite utilizar soluciones de software distribuidas. Sin embargo, algunas de estas soluciones incluyen servicios de nivel de aplicaciones proporcionados directamente por los 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. Estas diferentes configuraciones proporcionan conjuntos distintos de servicios. Al diseñar las soluciones de mensajería, estas configuraciones distintas se representan como componentes separados que están situados en distintas capas lógicas, como se muestra en la siguiente figura.



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

---

**Nota** – La [Figura 2-5](#) no pretende reflejar una arquitectura lógica completa puesto que se han omitido algunos componentes de Java ES para simplificar la ilustración. Las líneas que conectan los componentes representan interacciones.

---

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 46](#)). 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. No obstante, una dimensión igualmente importante de cualquier solución implementada es la capacidad de ésta para cumplir los requisitos de calidad de 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 de 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 de servicio para las arquitecturas de implementación de alto rendimiento y de gran escala.

Para diseñar una solución de software con éxito, se deberá determinar los requisitos de calidad de servicios relevantes y diseñar una arquitectura que satisfaga dichos requisitos. Se utiliza un número de importantes calidades de servicios para especificar los requisitos de calidad de 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.
Capacidad 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 de servicio influye en gran medida en la arquitectura de despliegue 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 Enterprise System

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, equilibradores de carga y 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 24, se resumen a continuación:

- **Componentes de disponibilidad.** Estos componentes proporcionan un tiempo de actividad prácticamente continuo en una solución implementada.
- **Componentes de acceso.** Estos componentes proporcionan un acceso a Internet seguro a los servicios del sistema y, a menudo, proporcionan también una función de enrutamiento.
- **Componentes administrativos.** Estos componentes proporcionan una mejor capacidad de mantenimiento para los componentes del sistema.

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

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

Componente	Calidades de sistema afectadas
Communications Express	Seguridad Escalabilidad
Directory Proxy Server	Seguridad Escalabilidad
almacén de sesión de alta disponibilidad	Disponibilidad
Portal Server Secure Remote Access	Seguridad Escalabilidad
Sun Cluster	Disponibilidad Escalabilidad
Web Proxy Server	Seguridad Rendimiento Capacidad de mantenimiento

## 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 recuperación de los fallos de los recursos que supervisa y utiliza, por tanto, la redundancia interna para proporcionar un acceso prácticamente continuo a estos recursos.

En la figura que aparece a continuación se representa un clúster de dos nodos que hace posible el uso de servicios de almacén de datos para Messaging Server y Calendar Server.

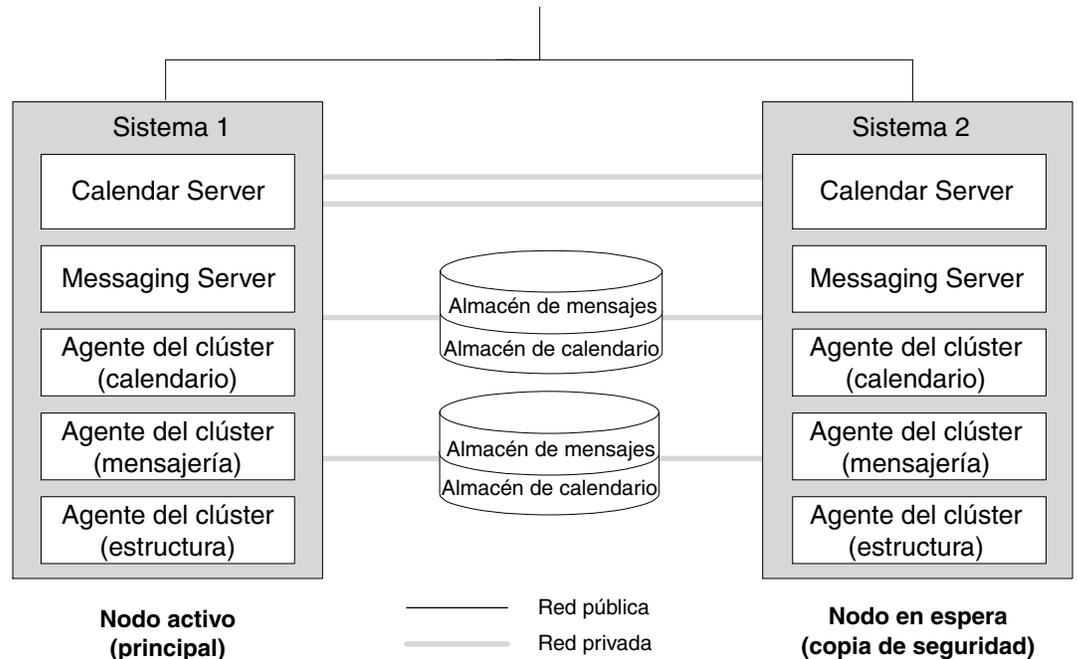


FIGURA 2-6 Diseño de disponibilidad usando nodos de Sun Cluster

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.

## 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 un marco 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. Cualquier arquitectura de soluciones deberá tomarlas todas 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 a la centralidad de Directory Server con respecto a estas dos dimensiones, los asuntos de calidad del servicio (dimensión 3) son vitales para este componente de Java ES. Un fallo de Directory Server tendría una tremenda repercusión en un sistema de negocios, de forma que el diseño de alta disponibilidad para este componentes es fundamental; y debido a que Directory Server se utiliza para almacenar información de configuración o de usuario importante, el diseño de seguridad de este componente también es muy importante.

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.

Queda fuera del ámbito de este manual la descripción detallada de las metodologías de diseño basadas en el marco arquitectónico que representa “[Marco arquitectónico de Java Enterprise System](#)” en la página 35. Sin embargo, el marco arquitectónico tridimensional destaca aspectos de diseño que son importantes para comprender la implementación de soluciones de software basadas en Java Enterprise System.

---

## Ejemplo de arquitectura de solución de Java Enterprise System

Java Enterprise System admite una amplia gama de soluciones de software.

Muchas soluciones se pueden diseñar e implementar con los valores de fábrica, sin que sea necesario hacer ningún desarrollo, utilizando los componentes incluidos en Java Enterprise System. Sin embargo, es posible que otras soluciones requieran mayores esfuerzos de desarrollo, por lo que deberá desarrollar componentes J2EE personalizados que proporcionen nuevos servicios de presentación o de negocios. Puede encapsular estos componentes personalizados como servicios web que cumplan con los estándares de interfaz de SOAP (Simple Object Access Protocol). Gran cantidad de soluciones implican una combinación de estos dos planteamientos.

Esta sección proporciona un ejemplo que muestra la forma en que Java Enterprise System 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.

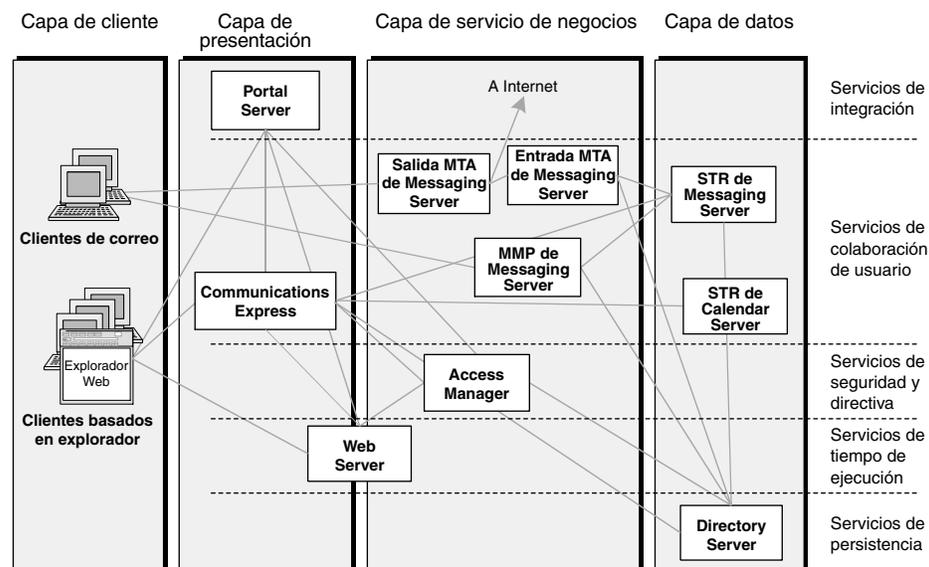
**TABLA 2-4** Resumen de requisitos de negocio: escenario de comunicaciones

Requisitos de la empresa	Descripción	Servicios necesarios de Java ES
Inicio de sesión único	Acceso a recursos y servicios empresariales seguros basándose en una única identidad con un único inicio de sesión 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	Un único punto de acceso personalizado basado en web para servicios de comunicación como correo electrónico y 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 aparece una arquitectura lógica para proporcionar los servicios de identidad, comunicación y portal identificados en la [Tabla 2-4](#) usando los componentes de Java ES. La arquitectura trata configuraciones distintas lógicamente de Messaging Server como componentes separados debido a los distintos servicios que cada una proporciona.



**FIGURA 2-7** 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 de Java ES. 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 *Guía de planificación de la implementación de Sun Java Enterprise System 2005Q4*.

## Arquitectura de implementación para el escenario de ejemplo

Al cambiar de la arquitectura lógica a la arquitectura de implementación, los requisitos de calidad de 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 implementaciones de soluciones de Java ES, consulte la *Guía de planificación de la implementación de Sun Java Enterprise System 2005Q4*.

---

## Términos clave de este capítulo

Esta sección explica los términos técnicos clave utilizados en este capítulo. Se pone un interés especial en clarificar las relaciones entre estos términos y cómo se utilizan en el contexto de Java Enterprise System.

<b>componente de aplicación</b>	Un <b>componente</b> de software desarrollado de forma personalizada para alguna función informática específica que proporciona <b>servicios de negocios</b> a los <b>usuarios finales</b> o a otros componentes de aplicaciones. 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 <b>servicios web</b> .
<b>arquitectura</b>	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 <b>aplicación de empresa distribuida</b> , el diseño arquitectónico utiliza generalmente la <b>arquitectura lógica</b> de la aplicación y la <b>arquitectura de implementación</b> .
<b>servicio de negocios</b>	Un <b>componente de aplicación</b> o un ensamblado de componentes que realizan la lógica de negocio en nombre de varios clientes (y es, en consecuencia, un proceso con varios subprocesos). Un servicio de

negocio también puede ser un ensamblado de componentes distribuidos encapsulados como un [servicio web](#) o puede ser un [servidor](#) independiente.

<b>cliente</b>	Software que solicita <a href="#">servicios</a> de software. (Nota: no se trata de una persona; consulte <a href="#">usuario final</a> .)Un cliente puede ser un servicio que solicita otro servicio o un componente de GUI al que obtiene acceso 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. Los estándares incluyen el protocolo de mensajería SOAP (del inglés ), la definición de interfaz WSDL (del inglés ) y el estándar de registro UDDI (del inglés ).



## Funciones de integración de Java Enterprise System

---

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 Enterprise System, frente a la integración manual de productos de infraestructura dispares.

En este capítulo se describen las siguientes funciones:

- “El instalador integrado de Java Enterprise System” en la página 57
- “Identidad integrada y servicios de seguridad” en la página 59
- “Términos clave de este capítulo” en la página 64

---

## El instalador integrado de Java Enterprise System

Todos los componentes de Java ES se instalan utilizando un único instalador (también llamado "programa de instalación"). Este programa de instalación proporciona procedimientos de instalación y desinstalación y comportamientos coherentes en todos los componentes.

El instalador de Java ES es una estructura integrada que transfiere el software de Java ES a un sistema host. Este programa permite seleccionar e instalar cualquier número de componentes de Java ES en cualquier equipo del entorno informático. También proporciona algunas opciones 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 instalador admite hasta siete idiomas: francés, alemán, español, coreano, chino simplificado, chino tradicional y japonés.

En esta sección se describen los siguientes aspectos del programa de instalación integrado de Java ES (para obtener información más detallada, consulte la *Guía de instalación de Sun Java Enterprise System 2005Q4 para UNIX*):

- [“Comprobación del software previo” en la página 58](#)
- [“Comprobación de relaciones de dependencia” en la página 58](#)
- [“Configuración inicial” en la página 59](#)
- [“Desinstalación” en la página 59](#)

## Comprobación del software previo

El programa de instalación examina el equipo donde está realizando la instalación e identifica los componentes de Java ES que ya están instalados. El programa de instalación realiza comprobaciones en varios niveles para asegurarse de que todos los componentes instalados previamente tienen un nivel de versión que permite 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 instalador comprueba si hay componentes compartidos instalados de Java ES (consulte [“Componentes compartidos” en la página 26](#)), como J2SE o NSS. Si el programa de instalación encuentra componentes compartidos cuyas versiones son incompatibles, informará sobre ello. Si continúa con la instalación, el programa de instalación actualiza automáticamente los componentes compartidos con las nuevas versiones.

## Comprobación de relaciones de dependencia

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. El programa de instalación proporciona la lógica para garantizar el cumplimiento de dichas relaciones de dependencia. Por esta razón, cuando selecciona un componente para instalarlo, el programa de instalación incluye automáticamente los componentes y subcomponentes con los que el componente seleccionado guarda 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. Para algunos componentes, el programa de instalación de Java ES puede realizar esta configuración inicial.

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 explícitamente para cada componente instalado tras completar 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 Enterprise System también cuenta con un programa de desinstalación. Puede utilizar este programa para quitar componentes que se instalaron en el equipo local mediante el programa de instalación de Java ES. El programa de desinstalación comprueba las relaciones de dependencia y advierte cuando descubre una de estas relaciones. El programa de desinstalación no elimina los componentes compartidos de Java ES.

Este programa, como el de instalación, se puede ejecutar en los modos gráfico, basado en texto o silencioso.

---

## Identidad integrada y servicios de seguridad

Una función importante de Java Enterprise System es su administración integrada de las identidades de los usuarios, así como su estructura integrada de autorización y autenticación.

Las siguientes secciones proporcionan información técnica para comprender la identidad y los servicios de seguridad integrados que ofrece Java Enterprise System:

- “Identidad única” en la página 60
- “Autenticación e inicio de sesión único” en la página 61

## Identidad única

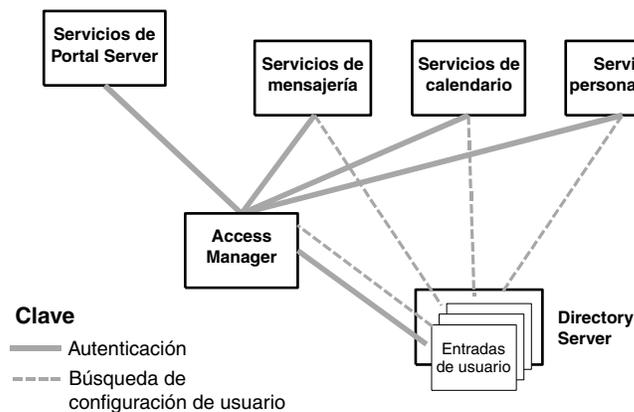
Dentro de un entorno de Java ES, cada usuario final tiene su identidad única integrada. En función de su **identidad única**, a un usuario se le puede conceder acceso a varios recursos como, por ejemplo, portales, páginas web o servicios como el uso de mensajes y calendarios y la mensajería instantánea.

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 **directorío** de usuarios. Esa información normalmente incluye datos como una contraseña y un nombre únicos, 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 Enterprise System, las entradas de los usuarios se almacenan en un directorio proporcionados 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 función muy útil que proporciona Java Enterprise System.

## Autenticación e inicio de sesión único

Los servicios de autorización y de autenticación de Java ES los proporciona Access Manager. 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.

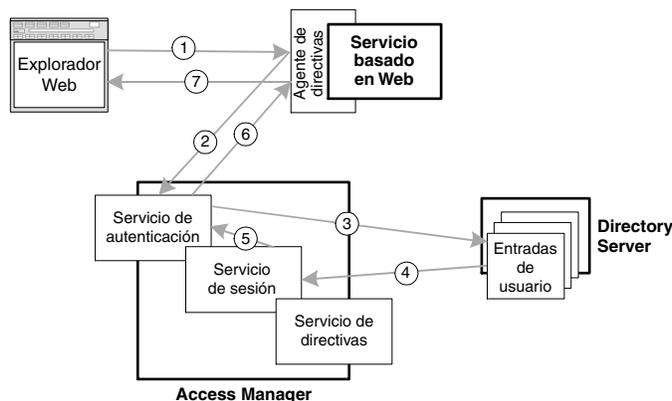
Access Manager también utiliza un componente externo denominado "agente de directivas". El agente de directivas se conecta al servidor web que alberga un servicio o recurso asegurado por Access Manager. El agente de directivas intercede en nombre de Access Manager en solicitudes que realizan los usuarios a los recursos asegurados. Para algunos componentes de Java ES como, por ejemplo, Portal Server y Communications Express, el funcionamiento del agente de directivas lo proporciona el subcomponente Access Manager (consulte ["Sun Java System Access Manager 7 2005Q4"](#) en la página 78).

## Autenticación

Access Manager incluye un servicio de autenticación para comprobar las identidades de los usuarios que solicitan acceso (mediante HTTP o HTTPS) a 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 la 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

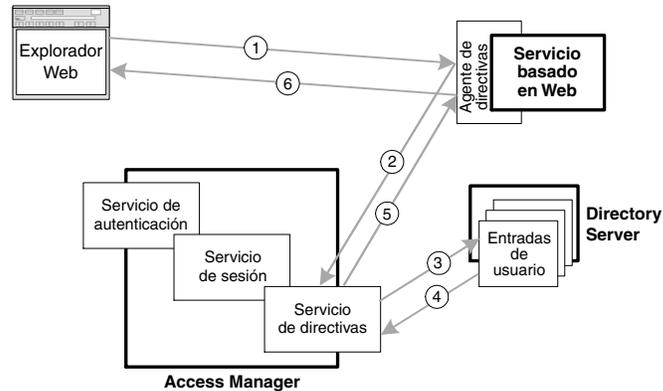
El escenario de autenticación descrito en los párrafos anteriores, pasa por alto un paso importante. Cuando se comprueba la solicitud de autenticación de un usuario, el servicio de sesiones de Access Manager se activa (4), tal y como se muestra en la [Figura 3-2](#). El servicio de sesiones genera un 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 políticas (6) que lo envía (como una cookie) al explorador (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 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 está autorizado a 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 políticas rechaza una solicitud de acceso, el agente de políticas impide el acceso a los recursos asegurados al usuario que lo solicita.

---

## Términos clave de este capítulo

Esta sección explica los términos técnicos clave utilizados en este capítulo. Se pone un interés especial en clarificar las relaciones entre estos términos y cómo se utilizan en el contexto de Java Enterprise System.

<b>directorio</b>	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 Enterprise System. En función de esa entrada de usuario única, un usuario puede obtener acceso a varios recursos de Java Enterprise System, como un portal, páginas web y servicios como 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.

## Tareas del ciclo de vida de la solución Java Enterprise System

---

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.

El capítulo está organizado alrededor de los tres grupos de etapas del ciclo de vida siguientes:

- “Implementación previa” en la página 67
- “Implementación” en la página 68
- “Implementación posterior” en la página 73
- “Términos clave de este capítulo” en la página 74

---

## 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. Este capítulo describe las tareas incluidas en cada etapa del ciclo de vida. El diagrama del ciclo de vida aparece de nuevo en la [Figura 4-1](#) para que sea más fácil consultarlo.

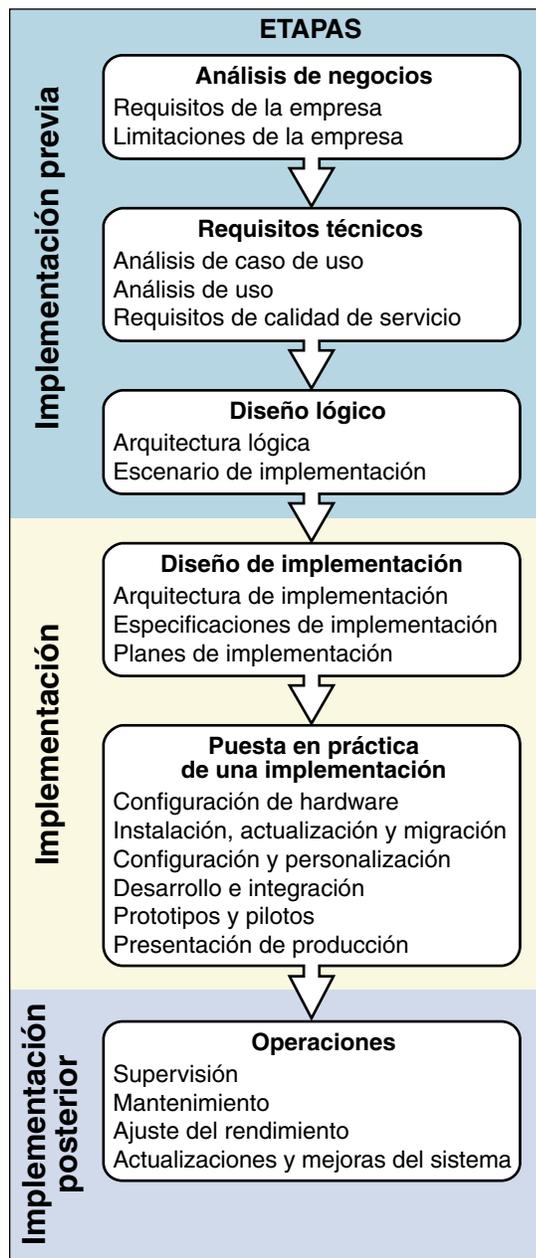


FIGURA 4-1 Tareas del ciclo de vida de las soluciones

---

## Implementación previa

En las [fase 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 del negocio.** En esta fase, puede definir los objetivos empresariales de una implementación propuesta y especificar las limitaciones y los requisitos de negocios que se deben cumplir para conseguir ese objetivo.
- **Requisitos técnicos.** En esta etapa se utilizan los resultados del análisis del negocio para crear [casos de uso](#) que perfilen la interacción del usuario con un sistema de software anticipado. También puede determinar patrones de utilización para esos casos de uso. Sirviéndose tanto del análisis del negocio 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.** En esta etapa, se analizan los casos de uso desarrollados en la etapa de requisitos técnicos para determinar los componentes de infraestructura de Java ES y los componentes de aplicaciones 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 de servicio se encapsulan en un escenario de implementación, tal y como se muestra en la siguiente figura. Para obtener más información acerca de la fase previa a la implementación del ciclo de vida, consulte la *Guía de planificación de la implementación de Sun Java Enterprise System 2005Q4*.

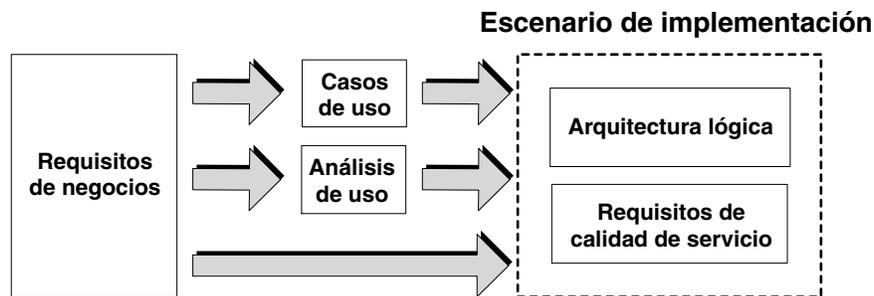


FIGURA 4-2 Especificación de un escenario de implementación

## Implementación

En las etapas de [implementación](#) del ciclo de vida, se convierte un escenario de implementación en un diseño de implementación, que, posteriormente, se desplegará, 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 servicio de infraestructura necesarios para dar soporte a una solución de software. En general, hay que implementar tanto los componentes de aplicaciones desarrolladas de forma personalizada (componentes J2EE, servicios web u otros servidores) como los componentes de Java ES necesarios para lograr que la solución sea compatible.

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

- “[Diseño de implementación](#)” en la [página 69](#). El diseño de 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 de 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.
- “[Despliegue de implementación](#)” en la [página 71](#). La implementación de un diseño de despliegue es a menudo un proceso repetitivo que implica la configuración de hardware, la instalación y configuración de software, el desarrollo y la integración, así como pruebas y otros aspectos del despliegue en un entorno de producción.

Las siguientes secciones detallan estas dos fases del proceso de implementación.

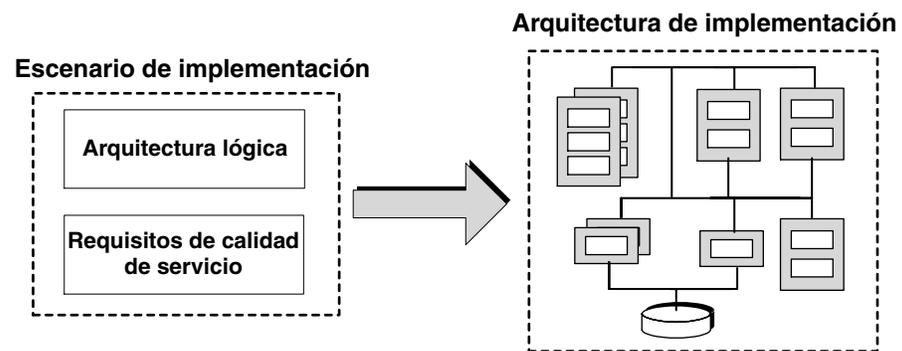
## Diseño de implementación

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 de 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 de servicio, así como el diseño en los equipos de gran potencia y en el amplio ancho de banda de red. Cuando el hardware es limitado, o excesivamente caro, quizá tenga que evaluar concesiones entre los costes fijos (hardware) y los costes variables (requisitos de recursos humanos) o entre los diferentes requisitos de calidad de 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. Sin embargo, como punto de partida para el diseño de implementación, Java Enterprise System desarrolla un conjunto de [arquitectura de implementación de referencia](#).

Una arquitectura de referencia se basa en un escenario de implementación específico: una arquitectura lógica con requisitos de calidad de 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 de 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 de hardware

- Instalación, actualización y migración de software
- Configuración y personalización del sistema
- Desarrollo e integración
- Comprobación
- Despliegue en producción

## Despliegue de implementación

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. Las siguientes subsecciones describen cada tarea principal de implementación en el orden en que se realizan normalmente. Para obtener información detallada sobre estas tareas, consulte la *Guía de documentación de Sun Java Enterprise System 2005Q4*.

### Configuración de 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 da soporte a la solución 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. El instalador que se incluye con Java ES se utiliza para instalar los componentes adecuados de Java ES en cada equipo de su arquitectura de implementación (consulte “El instalador integrado de Java Enterprise System” en la [página 57](#)).

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 Enterprise System, si está actualizando componentes de Java ES instalados anteriormente o si está sustituyendo componentes de terceros por otros de Java Enterprise System. 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, cada componente de Java ES debe estar configurado para poder 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 59).

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, normalmente se puede personalizar Portal Server para que proporcione canales de portal, Access Manager para que ejecute tareas de autorización y Messaging Server para que use comprobaciones de virus y filtros anti-spam.

## 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 empresariales. Sun Java Studio simplifica 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 ofrece un marco conector que permite integrar las aplicaciones existentes en el entorno de Application Server desarrollando adaptadores de recursos J2EE, mientras que Message Queue ofrece una función robusta de mensajería asíncrona para integrar diversas aplicaciones.

## Pruebas de prototipos y pilotos

En función del grado de personalización o trabajo de desarrollo necesario, en un momento dado será necesario comprobar la arquitectura de implementación; para ello, deberá comprobar la solución en los casos de uso para verificar que satisface los requisitos de calidad de 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 fase supone instalar, configurar e iniciar aplicaciones distribuidas y servicios de infraestructura en un entorno de producción, realizar la provisión de usuarios finales del sistema de producción, configurar el inicio de sesión único, 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 aplican cargas progresivamente mayores para confirmar que se cumplen los requisitos de calidad de servicio.

---

## Implementación posterior

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

- **Supervisión.** Estas tareas incluyen la supervisión periódica del rendimiento del sistema y de sus funciones.
- **Mantenimiento.** Estas tareas incluyen las funciones administrativas diarias, como añadir usuarios finales nuevos al sistema, cambiar contraseñas, añadir nuevos usuarios administrativos, cambiar los privilegios de acceso, realizar copias de seguridad periódicas, etc.
- **Ajuste del rendimiento.** Estas tareas incluyen 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, añadiendo capacidad, etc.
- **Actualizaciones y mejoras del sistema.** Entre estas tareas se incluye la adición de nuevos componentes de Java ES a un sistema para incluir nuevas funciones o para sustituir componentes que no sean de Java ES. En cualquiera de los casos, 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.

Cada componente de Java ES tiene sus propias herramientas de administración para configurar, ajustar o administrar sus operaciones. El objetivo es proporcionar una infraestructura común de supervisión y administración, y herramientas comunes para gestionar el sistema en su conjunto.

---

## Términos clave de este capítulo

Esta sección explica los términos técnicos clave utilizados en este capítulo. Se pone un interés especial en clarificar las relaciones entre estos términos y cómo se utilizan en el contexto de Java Enterprise System.

<b>implementación</b>	Etapa del ciclo de vida de una solución de Java Enterprise System en la que un escenario de implementación se convierte en un diseño de implementación, que luego se desplegará. También se hará un prototipo del mismo 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 Enterprise System y los requisitos de calidad del servicio que debe satisfacer la solución para ajustarse a las necesidades del negocio. Entre los requisitos de calidad de servicio se incluyen aquéllos relacionados con: el rendimiento, la disponibilidad, la seguridad, la facilidad de mantenimiento y las funciones de escalabilidad o latencia. 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 Enterprise System mediante la que los componentes personalizados de una <a href="#">arquitectura de implementación</a> se programan y prueban.
<b>fase previa a la implementación</b>	Una fase del ciclo de vida de una solución de Java Enterprise System en la que las necesidades de negocio 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 a los que debe ajustarse una solución.
<b>fase posterior a la implementación</b>	Una fase del proceso de vida de una solución de Java Enterprise System en la que las aplicaciones distribuidas se inician, se supervisan, se ajustan para optimizar su rendimiento y se actualizan dinámicamente para incluir nuevas funciones.

**arquitectura de implementación de referencia**

Una **arquitectura de implementación** 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.

**caso de uso**

Una tarea específica del usuario final o un conjunto de tareas realizadas por una **aplicación de empresa distribuida** que se utiliza como base para el diseño, la prueba y la evaluación del rendimiento de una aplicación.



## Lista de referencia: componentes de Java Enterprise System

---

Este apéndice ofrece una lista de referencia de todos los componentes de Java ES, agrupados en las siguientes categorías:

- “Descripciones de los componentes de servicios de sistemas” en la página 78. Estos componentes ofrecen los servicios de infraestructura clave de Java ES necesarios para el uso de aplicaciones de empresa distribuidas. Estos servicios, tal y como se describe en “¿Por qué usted necesita Java Enterprise System?” en la página 19, incluyen servicios de portal, de comunicación y colaboración, de seguridad e identidad, servicios web y de aplicaciones y, por último, servicios de disponibilidad.
- “Descripciones de componentes de calidad del servicio” en la página 83. Estos componentes se utilizan para mejorar la calidad de los servicios proporcionados por los componentes de servicio 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.
- “Componentes compartidos” en la página 87. Estos componentes son bibliotecas locales que pueden compartir todos los componentes de Java ES que se estén ejecutando en un equipo host concreto.

En este apéndice, los componentes de Java ES se enumeran alfabéticamente con sus respectivas categorías y subcategorías.

Si desea obtener una guía de la documentación de los distintos componentes, consulte *Guía de documentación de Sun Java Enterprise System 2005Q4*.

---

## Descripciones de los componentes de servicios de sistemas

Los componentes de los servicios del sistema de Java ES ofrecen los servicios de infraestructura necesarios para el uso de aplicaciones de empresa distribuidas. Los componentes de los servicios del sistema de Java ES se describen en las siguientes secciones:

- [“Sun Java System Access Manager 7 2005Q4”](#) en la página 78
- [“Sun Java System Application Server Enterprise Edition 8.1 2005Q4”](#) en la página 79
- [“Sun Java System Calendar Server 6 2005Q4”](#) en la página 79
- [“Sun Java System Directory Server 5 2005Q4”](#) en la página 80
- [“Sun Java System Instant Messaging 7 2005Q4”](#) en la página 80
- [“Sun Java System Message Queue 3.6 2005Q4”](#) en la página 80
- [“Sun Java System Messaging Server 6 2005Q4”](#) en la página 81
- [“Sun Java System Portal Server 6 2005Q4”](#) en la página 82
- [“Sun Java System Service Registry 3 2005Q4”](#) en la página 82
- [“Sun Java System Web Server 6.1 2005Q4”](#) en la página 82

### Sun Java System Access Manager 7 2005Q4

Sun Java System Access Manager (Access Manager) proporciona a las organizaciones la infraestructura necesaria para administrar los procesos utilizados para gestionar las identidades digitales de los clientes, los empleados y los socios que usan servicios basados en web y aplicaciones no basadas en 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.

El programa de instalación de Java ES ofrece Access Manager como único componente instalable. Si es necesario, los siguientes subcomponentes de Access Manager pueden instalarse por separado:

- **Identity Management y Policy Services Core.** 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. Este subcomponente también incluye los subcomponentes Access Manager SDK y Delegated Administrator (consulte [“Sun Java System Delegated Administrator 6 2005Q4”](#) en la página 87).
- **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.

- **Access Manager Administration Console.** Se trata de una interfaz gráfica que consolida servicios de identidades y administración de directivas. También proporciona una interfaz única para 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.

## Sun Java System Application Server Enterprise Edition 8.1 2005Q4

Sun Java System Application Server (Application Server) proporciona una plataforma compatible con J2EE para desarrollar e implementar servicios de aplicaciones y servicios web. Application Server proporciona servicios de infraestructura para la interacción entre componentes distribuidos acoplados firmemente, incluida la invocación de métodos remotos y otros servicios de tiempo de ejecución.

El programa de instalación de Java ES ofrece Application Server como único componente instalable. Si es necesario, los siguientes subcomponentes de Application Server pueden instalarse por separado:

- **Domain Administration Server.** Proporciona funciones administrativas de servidor, como administración y configuración de Application Server e implementación de componentes y aplicaciones J2EE.
- **Application Server Administration Client.** Proporciona clientes para administración gráfica que ayudan a administrar y configurar instalaciones de Application Server y aplicaciones alojadas. Administration Client también ayuda a implementar aplicaciones.
- **Command Line Administration Tool** (Herramienta de administración de línea de comandos). Proporciona clientes para administración mediante línea de comandos que ayudan a administrar y configurar instalaciones de Application Server y aplicaciones alojadas. La herramienta también ayuda a implementar aplicaciones.
- **Load Balancing Plug-in** (Complemento de balance de carga).
- **PointBase.** Proporciona una base de datos incorporada que se puede utilizar para las operaciones de persistencia.
- **Aplicaciones de ejemplo.**

## Sun Java System Calendar Server 6 2005Q4

Sun Java System Calendar Server (Calendar Server) es una solución escalable, basada en web que se usa para funciones de calendario y de programación en empresas y proveedores de servicios. Calendar Server admite calendarios de grupo y personales, así como calendarios para recursos como, por ejemplo, salas de conferencias y equipos.

El programa de instalación de Java ES ofrece Calendar Server como único componente instalable.

## Sun Java System Directory Server 5 2005Q4

Sun Java System Directory Server (Directory Server) proporciona un servicio de directorio centralizado para la información de la red, la intranet o 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.

El programa de instalación de Java ES ofrece Directory Server como único componente instalable.

## Sun Java System Instant Messaging 7 2005Q4

Sun Java System Instant Messaging (Instant Messaging) hace posible que los usuarios finales puedan usar las funciones de mensajería instantánea y sesiones de chat, así como enviar mensajes de alerta y compartir grupos de noticias de forma inmediata. Instant Messaging se puede usar con intranets y con Internet. También admite la interacción con otros proveedores de mensajería instantánea.

El programa de instalación de Java ES ofrece Instant Messaging como único componente instalable. Los siguientes subcomponentes de Instant Messaging se pueden instalar por separado:

- **Instant Messaging Server Core.** Incluye software para el servidor y el multiplexor.
- **Instant Messaging Resources.**
- **Access Manager Instant Messaging Service.**

## Sun Java System Message Queue 3.6 2005Q4

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 mensajería Message Queue común para enviar y recibir información. Los desarrolladores de aplicaciones se pueden centrar en la lógica de negocio de sus aplicaciones en lugar de en procesos concretos sobre cómo se comunican sus aplicaciones a través de la red.

Message Queue está disponible en dos ediciones:

- **Enterprise Edition** (predeterminada). Incluye compatibilidad con servicios de mensajería multibroker, conexiones HTTP/HTTPS, conexiones seguras y escalables, conmutación por error de la conexión del cliente y asistencia al cliente para el lenguaje C. Esta edición es ideal para implementar y ejecutar aplicaciones de mensajería en un entorno de producción a gran escala.
- **Platform Edition**. Ofrece compatibilidad básica con JMS y es ideal para implementaciones a pequeña escala y entornos de desarrollo.

El programa de instalación de Java ES incluye Message Queue Enterprise Edition y Message Queue Platform Edition como componentes para instalarlos por separado.

## Sun Java System Messaging Server 6 2005Q4

Sun Java System Messaging Server (Messaging Server) es un potente servidor de mensajería de Internet basado en estándares destinado a los proveedores de servicios y a las empresas. Messaging Server es una aplicación de gran capacidad diseñada para gestionar mensajes de forma fiable. Está formada por varios componentes modulares, que se pueden configurar de forma independiente y que son compatibles con varios protocolos de correo electrónico.

El programa de instalación de Java ES ofrece Messaging Server como único componente instalable. Sin embargo, después de la instalación, cada instancia de Messaging Server se puede configurar para proporcionar varios servicios de mensajería distintos, representando de hecho el siguiente conjunto de subcomponentes de Messaging Server:

- **Message Store**. Proporciona funciones de almacenamiento y recuperación.
- **Message Transfer Agent (MTA)**. Admite el envío de correo electrónico administrando las conexiones SMTP, encaminando los correos electrónicos y enviando los mensajes a los almacenes de mensajes adecuados. Se puede configurar para enviar un correo electrónico a un almacén interno (entrada) o a almacenes externos de correo (salida).
- **Message Multiplexor (MMP)**. Permite recuperar correos electrónicos accediendo al almacén de mensajes (o a un conjunto de almacenes) para clientes de correo electrónico mediante protocolos IMAP o POP.
- **Message Express Multiplexor (MEM)**. Hace posible la recuperación y el envío de mensajes de correo electrónico usando clientes de correo basados en web (HTML).

## Sun Java System Portal Server 6 2005Q4

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, personalización, adición, seguridad, integración y búsqueda. Mobile Acces (Acceso móvil), un subcomponente de Portal Server, proporciona acceso inalámbrico a Portal Server desde dispositivos móviles, como teléfonos móviles y asistentes digitales personales.

El instalador de Java ES proporciona Portal Server, incluido el acceso móvil, como el único componente instalable.

## Sun Java System Service Registry 3 2005Q4

Sun Java System Service Registry (Service Registry) es un depósito que actúa como registro (UDDI) para los servicios web y como registro XML para 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 elementos de información necesarios para que sea posible la integración de los procesos de negocio. 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.

## Sun Java System Web Server 6.1 2005Q4

Sun Java System Web Server (Web Server) es un servidor web seguro para varios procesos y subprocesos creado a partir de estándares abiertos. Web Server proporciona un gran rendimiento, fiabilidad, escalabilidad y facilidad de administración para empresas de todos los tamaños. Web Server admite una amplia gama de estándares de software web, incluidos JDK 1.4.1, Java Servlet 2.3, JavaServer Pages™ (JSP™) 1.2, HTTP/1.1, PKCS #11, FIPS-140, certificados configurados de 168 bits y otros estándares basados en la seguridad.

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

---

## Descripciones de componentes de calidad del servicio

Los componentes que se utilizan para apoyar a los componentes de servicios de Java ES están agrupados en las siguientes categorías:

- Componentes de disponibilidad
- Componentes de acceso
- Componentes administrativos

Los componentes de estas categorías se describen en las siguientes secciones.

## Descripciones de los 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:

- [“Sun Cluster 3.1 8/05 y Sun Cluster Agents” en la página 83](#)
- [“High Availability Session Store 2005Q4” en la página 84](#)

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

El software de Sun Cluster proporciona servicios de alta disponibilidad y escalabilidad para Java Enterprise System, 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 Sun Cluster Core y Sun Cluster Agents como componentes que se pueden instalar por separado. Existen agentes adicionales de Sun Cluster que están disponibles en CD separados.

## High Availability Session Store 2005Q4

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 de estos componentes que puede utilizar los servicios 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.

## Descripciones de 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:

- “Sun Java System Communications Express 6 2005Q4” en la página 84
- “Sun Java System Web Proxy Server 4.0.1 2005Q4” en la página 85
- “Sun Java System Connector para Microsoft Outlook 6 2005Q4” en la página 85
- “Sun Java System Directory Proxy Server 5 2005Q4” en la página 85
- “Sun Java System Portal Server Secure Remote Access 6 2005Q4” en la página 86

## Sun Java System Communications Express 6 2005Q4

Sun Java System Communications Express (Communications Express) proporciona un cliente integrado de comunicación y colaboración basado en web que permite a los usuarios finales disponer de una interfaz web para los servicios de calendario, correo electrónico y libreta de direcciones. Communications Express consta de estos tres módulos de cliente: libreta de direcciones, calendario y correo. Communications Express se puede configurar para proporcionar acceso a Messaging Server o a Calendar Server o a ambos. Funciona con Sun Java System LDAP Schema, Versión 1 (Schema 1) o Schema 2.

El programa de instalación de Java ES ofrece Communications Express como único componente instalable.

## Sun Java System Web Proxy Server 4.0.1 2005Q4

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.

## Sun Java System Connector para Microsoft Outlook 6 2005Q4

Sun Java System Connector para Microsoft Outlook permite que Outlook se pueda usar como cliente de escritorio con Sun Java Enterprise System. El conector es un complemento de Outlook que debe instalarse en el escritorio del usuario.

Connector para Microsoft Outlook efectúa consultas a Messaging Server para obtener información sobre las jerarquías de carpetas y los mensajes de correo electrónico y después convierte esa información en propiedades de API de mensajería (MAPI) para su visualización en Outlook. Del mismo modo, usa WCAP para efectuar consultas a Calendar Server para acontecimientos y tareas que se convierten entonces en propiedades MAPI. Con este modelo, Sun Java System Connector para Microsoft Outlook crea una vista de Outlook de usuario final a partir de dos orígenes de información distintos: información de correo procedente de Messaging Server e información de calendario procedente de Calendar Server.

Sun Java System Connector para Microsoft Outlook se incluye en el CD de accesorios, con su propio programa de instalación.

## Sun Java System Directory Proxy Server 5 2005Q4

Sun Java System Directory Proxy Server (Directory Proxy Server) es un componente básico de cualquier servicio de directorio cuyo funcionamiento sea vital para las soluciones de comercio electrónico. Directory Proxy Server es una puerta de enlace de protocolo de aplicación LDAP que ofrece funciones mejoradas de control de acceso al directorio, compatibilidad de esquema y alta disponibilidad usando técnicas de conmutación por error y de balance de la carga de la aplicación.

El programa de instalación de Java ES ofrece Directory Proxy Server como único componente instalable.

## Sun Java System Portal Server Secure Remote Access 6 2005Q4

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.

El programa de instalación de Java ES ofrece Portal Server Secure Remote Access como único componente instalable. Si es necesario, los siguientes subcomponentes de Portal Server Secure Remote Access pueden instalarse por separado:

- **Portal Server Secure Remote Access Core.**
- **Puerta de enlace.** Proporciona una interfaz y una barrera de seguridad para una intranet corporativa que permite el acceso remoto desde el exterior de la intranet. La puerta de enlace presenta al usuario remoto de forma segura el contenido de servidores web internos y servidores de aplicación a través de una única interfaz.
- **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.
- **Proxy Rewriter.** Proporciona un acceso seguro a las páginas web de una intranet corporativa desde el exterior de la intranet mediante la transformación de los enlaces web y la creación de conjuntos de reglas para gestionar las páginas web de la intranet.

## Descripciones de componentes administrativos

Los componentes administrativos proporcionan funciones de administración, como configuración y supervisión, para los servicios del sistema. Los siguientes componentes administrativos de Java ES se describen en esta sección:

- [“Sun Java System Administration Server \(y la consola\) 5 2005Q4”](#) en la página 86
- [“Sun Java System Directory Preparation Tool 2005Q4”](#) en la página 87
- [“Sun Java System Delegated Administrator 6 2005Q4”](#) en la página 87

## Sun Java System Administration Server (y la consola) 5 2005Q4

Sun Java System Administration Server (Administration Server) y Server Console proporcionan juntos una herramienta gráfica que permite administrar Directory Server, Directory Proxy Server y Messaging Server. Administration Server procesa las peticiones de los servidores instalados en un grupo de servidores en el mismo directorio root e inicia los programas requeridos para atender las peticiones.

Server Console es una aplicación Java independiente que trabaja en su red junto con una instancia de Directory Server y otra de Administration Server. Server Console actúa como aplicación de administración cliente del software de Java ES en su empresa.

El programa de instalación de Java ES proporciona Server Console y Administration Server juntos como un único componente instalable.

## Sun Java System Directory Preparation Tool 2005Q4

Sun Java System Directory Preparation Script se usa para configurar Directory Server con el esquema necesario para realizar la provisión de los usuarios para Messaging Server y Calendar Server.

## Sun Java System Delegated Administrator 6 2005Q4

Sun Java System Delegated Administrator es una utilidad de línea de comandos (`commadmin`) para realizar la provisión de usuarios, grupos, dominios y recursos para Calendar Server, Messaging Server y otros proveedores de servicios de Java ES.

Delegated Administrator se instala automáticamente cuando elige instalar Access Manager.

---

# Componentes compartidos

Los componentes compartidos proporcionan servicios locales y compatibilidad tecnológica que requieren los componentes de servicios de sistema de Java ES para funcionar. El instalador de Java ES instala automáticamente los componentes compartidos requeridos que necesiten otros componentes de Java ES que estén instalados en un equipo host.

Java Enterprise System incluye los componentes compartidos que aparecen a continuación:

- ANT (herramienta de creación basada en Jakarta ANT Java/XML)
- Apache Inicio de sesión común
- Berkeley DB
- Contenedor de agentes común
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)

- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML basado en llamadas de procedimiento remoto) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- MA (Mobile Access) Core
- MFWK (Java ES Monitoring Framework)
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- Sun Explorer Data Collector (sólo Solaris)
- Sun Java Web Console
- WSCL (Web services Common Library)

# Índice

---

## A

- Access Manager
  - como componente de los servicios del sistema, 23
  - como servicio de infraestructura, 40
  - descripción de, 78
- Administration Server
  - como componente de calidad del servicio, 26
  - descripción, 86
- Ant (herramienta de creación basada en Jakarta ANT Java/XML), 87
- Apache SOAP Runtime, 87
- aplicaciones
  - de empresa
    - Ver* aplicaciones de empresa distribuidas
  - distribuidas
    - Ver* aplicaciones de empresa distribuidas
  - de empresa distribuida, acerca de, 19
  - de empresa distribuidas
    - definición, 33
    - infraestructura para, 20
- Application Server
  - como componente de los servicios del sistema, 23
  - como servicio de infraestructura, 40
  - descripción, 79
- arquitectura
  - definición, 54
  - dimensiones
    - Ver* dimensiones arquitectónicas
  - implementación, 69-70

- arquitectura (Continuación)
  - introducción, 35
  - referencia, 70
  - solución, 36
- arquitecturas de implementación
  - definición, 55
  - diseño de, 69-70
  - introducción, 35
  - relación con las arquitecturas en capas, 44
- arquitecturas de implementación de referencia
  - definición, 75
  - introducción, 70
- arquitecturas lógicas
  - definición, 55
  - ejemplo, 52-53
  - introducción, 35
  - niveles de servicio de infraestructura, 37
- autenticación, 61-62
- autorización, 63

## B

- Berkeley DB, 87

## C

- Calendar Server
  - como componente de los servicios del sistema, 23
  - como servicio de infraestructura, 40
  - descripción, 79

- capa lógica de cliente, 43
- capas, lógicas
  - arquitectura de aplicación, 42
  - cliente, 43
  - datos, 44
  - presentación, 43
  - servicios de negocio, 44
- CAPI (Java Calendar API), 88
- casos de uso
  - definición, 75
  - introducción, 67
- categorías de usuario
  - analista de sistemas, 29
  - planificador de negocios, 29
- categorías de usuarios
  - administrador de sistemas, 30
  - administrador de sistemas especializado, 30
  - administrador de TI, 30
  - administrador delegado, 30
  - arquitecto, 29
  - ingeniero de campo, 30
  - integrador de sistemas, 30
- clientes
  - componentes de los servicios del sistema
    - y, 22
    - definición, 55
- clústeres
  - Ver* Sun Cluster
- Communications Express
  - como componente de calidad del servicio, 25
  - descripción, 84
- compatibilidad con idiomas, 58
- componentes
  - acceso, 25-26
  - administrativos, 26
  - calidad del servicio, 24-26, 83-87
  - compartidos, 26, 87-88
  - definición, 32
  - dependencias, 41-42
  - detección de versiones instaladas, 58
  - disponibilidad, 24-25
  - distribuidos, 20
  - EJB, 44
  - J2EE, 43
  - JSP, 43
  - servicio de sistema, 78-82
  - servicio del sistema, 22-24
- componentes (Continuación)
  - servicios de infraestructura, 40
  - Servlet, 43
  - sistema
    - Ver* componentes del sistema
- componentes administrativos
  - descripciones, 86-87
  - introducción, 26
- componentes compartidos, 87
  - componentes compartidos, 26
  - definición, 33
  - introducción, 26
- componentes de acceso
  - descripciones, 84-86
  - introducción, 25-26
- componentes de aplicación
  - arquitecturas de capas lógicas, 43
  - definición, 54
- componentes de calidad del servicio
  - definición, 33
  - descripciones, 83-87
  - introducción, 24-26
- componentes de disponibilidad
  - descripciones, 83-84
  - introducción, 24-25
- componentes de Java Servlet, 43
- componentes de JSP, 43
- componentes de los servicios del sistema
  - definición, 34
  - dependencias, 41-42
  - introducción, 22-24
- componentes de sistema
  - acerca de, 22
  - componentes compartidos, 87-88
  - servicio de sistema, 78-82
- componentes del sistema
  - componentes de calidad del servicio, 24-26
  - componentes de los servicios del sistema
    - Ver* componentes de los servicios del sistema
    - definición, 33
  - componentes EJB, 44
- comprobación de relaciones de dependencia, programa de instalación, 58-59
- Connector for Microsoft Outlook
  - como componente de calidad del servicio, 25
  - descripción, 85

contenedor de agentes común, 87

## D

Delegated Administrator  
  como componente de calidad del servicio, 26  
  descripción, 87  
dependencias, 41-42, 59  
desarrollo  
  como tarea de implementación, 72  
  definición, 74  
detección del software instalado, 58  
dimensiones arquitectónicas  
  calidad del servicio, 46-50  
  capas lógicas, 42  
  dependencias de los servicios de infraestructura, 37  
  síntesis, 50-51  
directivas  
  autorización, 63  
  definición, 64  
directorios  
  definición, 64  
  en forma de almacenes de datos de usuarios, 60  
Directory Preparation Tool  
  como componente de calidad del servicio, 26  
  descripción, 87  
Directory Proxy Server  
  como componente de calidad del servicio, 25  
  como componente del sistema, 48  
  descripción, 85  
Directory Server  
  como componente de los servicios del sistema, 23  
  como servicio de infraestructura, 40  
  descripción, 80  
disponibilidad  
  requisitos, 47, 48  
  servicios, 49, 83  
distribuida  
  aplicaciones  
    *Ver* aplicaciones de empresa distribuida

distribuidos  
  servicios  
    *Ver* servicios distribuidos

## E

entrada de usuario, 60  
escalabilidad  
  requisitos, 47, 48  
  servicios, 49, 83  
escenarios de adopción, Java ES  
  acerca de, 30-32  
  actualización, 31  
  ampliación, 30  
  definición, 32  
  mejora, 30  
  sistema nuevo, 30  
escenarios de implementación  
  definición, 74  
  introducción, 67  
especificaciones de implementación, 70  
etapa previa a la implementación, etapas del ciclo de vida, 67  
etapas del ciclo de vida  
  etapa previa a la implementación, 67  
  fase posterior a la implementación, 29, 73  
  fase previa a la implementación, 29  
  implementación, 29, 68

## F

fase posterior a la implementación  
  definición, 74  
  etapas del ciclo de vida, 73  
fase previa a la implementación, definición, 74  
formación, escenarios de adopción de Java ES, 31  
funciones de integración  
  componentes compartidos, 21  
  identidad y seguridad, 22, 59-63  
  instalador integrado, 22, 57-59  
  introducción, 21

## G

glosario, vínculo al, 15

## H

- hardware, escenarios de adopción de Java ES, 31
- High Availability Session Store
  - como componente de calidad del servicio, 25
  - descripción, 84

## I

- ICU (International Components for Unicode), 87
- identidad
  - administración, 59
  - servicios, 21, 59-63
  - único usuario, 60-61
- identidad única
  - definición, 64
  - introducción, 60
- implementación
  - arquitectura, 69
  - arquitecturas de referencia, 70
  - definición, 74
  - desarrollo y personalización, 72
  - despliegue, 71-73
  - diseño, 69-71
  - escenarios
    - Ver* escenarios de implementación
  - etapa del ciclo de vida, 68-73
  - etapas del ciclo de vida, 68
  - presentación de producción, 73
  - prueba de prototipos, 72
- infraestructura
  - dependencias de servicio
    - Ver* servicios distribuidos
  - para aplicaciones de empresa distribuidas, 20
- inicio de sesión único
  - definición, 64
  - despliegue, 62
  - función de Java ES, 23, 61
  - niveles de los servicios de infraestructuras, 39
- Instant Messaging
  - como componentes de los servicios del sistema, 23
  - como servicio de infraestructura, 40

## Instant Messaging (Continuación)

- descripción, 80
- integración
  - escenarios de adopción de Java ES, 31
  - funciones
    - Ver* funciones de integración
  - servicios, 39

## J

- J2EE
  - componentes, 43
  - modelo de componentes distribuidos, 44
  - plataforma, 24
- J2SE (Java 2 Platform, Standard Edition), 26, 87
- JAF (JavaBeans™ Activation Framework), 87
- JATO (Java Studio Enterprise Web Application Framework), 87
- JavaHelp, 88
- JavaMail, 88
- JAX-RPC, 88
- JAXB (Java Architecture for XML Binding), 88
- JAXP (Java API for XML Processing), 88
- JAXR (Java API for XML Registries), 88
- JDMK (Java Dynamic Management™ Kit), 88
- JMS (Java Message Service), 23
- JSS (Java Security Services), 26, 88

## K

- KT Search Engine (KTSE), 88

## L

- LDAP, 44, 64
- LDAP C SDK, 88
- LDAP Java SDK, 88
- Linux, 40

## M

- mejoras, *Ver* escenarios de adopción
- Message Queue
  - como componente de los servicios del sistema, 23
  - como servicio de infraestructura, 40
  - descripción, 80
- Messaging Server
  - como componente de los servicios del sistema, 23
  - como servicio de infraestructura, 40
  - descripción, 81
- migración, escenarios de adopción de Java ES, 31

## N

- NSPR (Netscape Portable Runtime), 26, 88
- NSS (Network Security Services), 26, 88

## P

- perfiles del usuario, 29
- plataforma J2ME, 43
- Portal Server
  - como componente de los servicios del sistema, 24
  - como servicio de infraestructura, 40
  - descripción, 82
- Portal Server Secure Remote Access
  - como componente de calidad del servicio, 25
  - como componente del sistema, 48
  - descripción, 86
- presentación de producción, 73
- productos de Sun Java System
  - Access Manager
    - Ver* Access Manager
  - Administration Server
    - Ver* Administration Server
  - Application Server
    - Ver* Application Server
  - Calendar Server
    - Ver* Calendar Server
  - Communications Express
    - Ver* Communications Express

## productos de Sun Java System (Continuación)

- Connector for Microsoft Outlook
    - Ver* Connector for Microsoft Outlook
  - Delegated Administrator
    - Ver* Delegated Administrator
  - Directory Preparation Tool
    - Ver* Directory Preparation Tool
  - Directory Proxy Server
    - Ver* Directory Proxy Server
  - Directory Server
    - Ver* Directory Server
  - High Availability Session Store
    - Ver* High Availability Session Store
  - Instant Messaging
    - Ver* Instant Messaging
  - Message Queue
    - Ver* Message Queue
  - Messaging Server
    - Ver* Messaging Server
  - Portal Server
    - Ver* Portal Server
  - Portal Server, Secure Remote Access
    - Ver* Portal Server, Secure Remote Access
  - Web Server
    - Ver* Web Server
- programa de desinstalación, 59
  - prototipo, 72
  - provisión de usuarios, 70

## R

- requisitos de calidad del servicio
  - capacidad de mantenimiento, 47, 48
  - capacidad latente, 47
  - disponibilidad, 47, 48
  - escalabilidad, 47, 48
  - rendimiento, 47
- requisitos de capacidad de mantenimiento, 47, 48
- requisitos de capacidad latente, 47
- requisitos de la calidad del servicio
  - seguridad, 47, 48
- requisitos de rendimiento, 47

## S

- SAAJ (SOAP with Attachments API for Java), 88
- SASL (Simple Authentication and Security Layer), 88
- seguridad
  - requisitos, 47, 48
  - servicios, 21
  - servicios de directivas, 39
- servicios
  - alta disponibilidad, 49, 83
  - definición, 33
  - escalabilidad, 49, 83
  - infraestructura, 20
    - Ver servicios de infraestructura distribuida
  - web, 44
- servicios administrativos, 21
- servicios de acceso, 21
- servicios de aplicación, 21, 37
- servicios de colaboración, 21
- servicios de colaboración de usuarios, 39
- servicios de comunicación, 21
- servicios de disponibilidad, 21
- servicios de mensajería, 38
- servicios de negocio, capa de presentación, 43
- servicios de negocios, definición, 54
- servicios de nivel intermedio, 37
- servicios de persistencia, 38
- servicios de plataforma, 37
- servicios de portal, 20
- servicios de sistema, acerca de, 22
- servicios de sistema operativo, 38
- servicios de tiempo de ejecución, 39
- servicios de transporte de red, 38
- servicios del sistema, definición, 33
- servicios distribuidos
  - acceso, 21
  - administrativos, 21
  - colaboración de usuarios, 39
  - comunicación y colaboración, 21
  - disponibilidad, 21
  - identidad, 21
  - infraestructura, 20
  - integración, 39
  - mensajería, 38
  - nivel de aplicación, 37
  - nivel intermedio, 37
  - persistencia, 38

- servicios distribuidos (Continuación)
  - plataforma, 37, 38
  - portal, 20
  - seguridad, 21, 39
  - tiempo de ejecución, 21, 39
  - transporte de red, 38
  - visión general, 20
  - web, 21
- servicios web, 21
  - componentes de J2EE, 44
  - definición, 55
- servidores
  - definición, 55
  - independientes, 44
- sistema
  - componentes
    - Ver componentes del sistema
  - configuración, 59
  - servicios, 19-21
- SOAP (Apache), 87
- Solaris, 40
- soluciones, Java ES
  - arquitectura, 35
  - ciclo de vida, 27-30
  - ejemplo, 51
  - personalizadas y de fábrica, 51
- Sun Cluster
  - agentes, 49
  - como componente de calidad del servicio, 25
  - como servicio de disponibilidad, 49-50
  - descripción, 83
- Sun Explorer Data Collector, 88
- Sun Java Web Console, 88

## T

- tareas, Java Enterprise System, 27, 65

## U

- usuarios, provisión, 70
- usuarios finales
  - aplicaciones distribuidas y, 20
  - definición, 33

## **W**

Web Server

como componente de los servicios del sistema, 24

como servicio de infraestructura, 40

descripción, 82

WSCL (Web services Common Library), 88

