



Sun Java™ System

Sun Java Enterprise System 2005Q1

Visión general técnica

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
EE.UU.

Nº de pieza: 819-1924

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, EE.UU. Todos los derechos reservados.

Sun Microsystems, Inc. tiene derechos de propiedad intelectual relacionados con la tecnología incluida en el producto descrito en este documento. Especialmente, aunque sin limitarse a ello, estos derechos de propiedad intelectual pueden incluir una o varias patentes en Estados Unidos, las cuales figuran en <http://www.sun.com/patents>, así como una o varias patentes adicionales (o patentes pendientes de adjudicación) en Estados Unidos y en otros países.

ESTE PRODUCTO CONTIENE INFORMACIÓN CONFIDENCIAL Y SECRETOS COMERCIALES DE SUN MICROSYSTEMS, INC. EL USO, LA REVELACIÓN O LA REPRODUCCIÓN DE DICHOS DATOS ESTÁN PROHIBIDOS SIN EL EXPRESO CONSENTIMIENTO PREVIO Y POR ESCRITO DE SUN MICROSYSTEMS, INC.

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

Esta distribución puede incluir materiales desarrollados por terceras partes.

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

Sun, Sun Microsystems, el logotipo de Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet y los logotipos de Duke, de Java Coffee Cup, de Solaris, de SunTone Certified y de Sun ONE son marcas comerciales o marcas comerciales registradas de Sun Microsystems, Inc. en EE.UU. y 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 arquitecturas desarrolladas por Sun Microsystems, Inc.

Legato y el logotipo de Legato son marcas comerciales registradas y Legato NetWorker es una marca comercial o una marca comercial registrada de Legato Systems, Inc. El logotipo de Netscape Communications Corp es una marca comercial o una marca comercial registrada de Netscape Communications Corporation.

La interfaz gráfica de usuario de OPEN LOOK y Sun™ fue desarrollada por Sun Microsystems, Inc. para sus usuarios y licenciarios.

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 licenciarios de Sun que implementen la interfaz gráfica de usuario OPEN LOOK y que actúen conforme a los acuerdos de licencia por escrito de Sun.

Los productos que se tratan y la información contenida en este manual de servicio están controlados por las leyes de control de exportación de los Estados Unidos y pueden estar sujetos a leyes de exportación o importación en otros países. Queda terminantemente prohibido el uso final (directo o indirecto) de esta documentación para el desarrollo de armas nucleares, químicas, biológicas, de uso marítimo nuclear o misiles. Queda terminantemente prohibida la exportación o reexportación a países sujetos al embargo de los Estados Unidos o a entidades identificadas en las listas de exclusión de exportación de los Estados Unidos, incluidas, pero no de forma excluyente, las personas con acceso denegado y las listas de ciudadanos designados especialmente.

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

Contenido

Lista de figuras	7
Lista de tablas	9
Prefacio	11
Público	12
Uso de la documentación	12
Convenciones	14
Recursos Web	14
Comunicar problemas	15
Sun valora sus comentarios	15
Capítulo 1 Introducción a Java Enterprise System	17
¿Por qué necesita Java Enterprise System?	18
Componentes de Java Enterprise System	21
Componentes de los servicios del sistema	22
Componentes de calidad del servicio	23
Componentes de disponibilidad	24
Componentes de acceso	24
Componentes administrativos	25
Componentes compartidos	26
Uso de Java Enterprise System	27
Ciclo de vida de soluciones de Java Enterprise System	27
Java Enterprise System Escenarios de adopción	29
Términos clave de este capítulo	32

Capítulo 2 Arquitecturas de soluciones de Java Enterprise System	35
Marco arquitectónico de Java Enterprise System	36
Dimensión 1: dependencias de servicio de infraestructura	37
Niveles de servicio de infraestructura	37
Componentes de los servicios de infraestructura de Java Enterprise System	41
Dependencias de servicio de infraestructura de Java Enterprise System	42
Dimensión 2: capas lógicas	43
Descripción de capas lógicas	44
Independencia lógica y física	46
Arquitectura en capas aplicada a los componentes del sistema	46
Dimensión 3: calidad de servicio	48
Calidades de servicio	48
Componentes de calidad del servicio de Java Enterprise System	49
Software de Sun Cluster	50
Síntesis de las tres dimensiones arquitectónicas	52
Ejemplo de arquitectura de soluciones de Java Enterprise System	53
Escenario de comunicaciones de las empresas	53
Arquitectura lógica para el escenario de ejemplo	54
Arquitectura de implementación para el escenario de ejemplo	56
Términos clave de este capítulo	56
Capítulo 3 Funciones de integración de Java Enterprise System	59
Programa de instalación integrado de Java Enterprise System	60
Comprobación del software previo	61
Comprobación de relaciones de dependencia	61
Configuración inicial	62
Desinstalación	62
Identidad integrada y servicios de seguridad	62
Identidad única	63
Autenticación e inicio de sesión único	64
Autenticación	64
Inicio de sesión único	65
Autorización	66
Términos clave de este capítulo	67

Capítulo 4 Tareas del ciclo de vida de las soluciones de Java Enterprise System	69
Implementación previa	71
Implementación	72
Diseño de implementación	73
Arquitectura de implementación	73
Especificaciones de implementación	74
Planes de implementación	75
Puesta en práctica de una implementación	75
Configuración de hardware	75
Instalación, actualización y migración de software	75
Configuración y personalización del sistema	76
Desarrollo e integración	76
Pruebas de prototipos y pilotos	77
Presentación de producción	78
Implementación posterior	78
Términos clave de este capítulo	79
Apéndice A Lista de referencia: Componentes de Java Enterprise System	81
Descripciones de los componentes de servicios de sistemas	82
Sun Java System Access Manager 6 2005Q1	82
Sun Java System Application Server Enterprise Edition 8 2005Q1	83
Sun Java System Calendar Server 6 2005Q1	84
Sun Java System Directory Server 5 2005Q1	84
Sun Java System Instant Messaging 7 2005Q1	84
Sun Java System Message Queue 3 2005Q1	85
Sun Java System Messaging Server 6 2005Q1	85
Sun Java System Portal Server 6 2005Q1	86
Sun Java System Web Server 6.1 2005Q1	86
Descripciones de componentes de calidad del servicio	87
Descripciones de los componentes de disponibilidad	87
Sun Cluster 3.1 9/04 y Sun Cluster Agents	87
Almacén de sesión de alta disponibilidad 2005Q1	88
Descripciones de componentes de acceso	88
Sun Java System Communications Express 2005Q1	88
Sun Java System Connector para Microsoft Outlook 6 2005Q1	89
Sun Java System Directory Proxy Server 5 2005Q1	89
Sun Java System Portal Server Secure Remote Access 6 2005Q1	89
Descripciones de componentes administrativos	90
Sun Java System Administration Server (y Console) 5 2005Q1	90
Herramienta de preparación de Sun Java System Directory 2005Q1	91
Sun Java System Delegated Administrator 6 2005Q1	91
Sun Remote Services Net Connect 2005Q1	91
Componentes compartidos	92
Índice	95

Lista de figuras

Figura 1-1	Compatibilidad necesaria para aplicaciones de empresa distribuidas	19
Figura 1-2	Categorías de los componentes de Java Enterprise System	21
Figura 1-3	Etapas del ciclo de vida de la solución y categorías de usuarios	28
Figura 2-1	Dimensiones de la arquitectura de soluciones de Java ES	37
Figura 2-2	Dimensión 1: Niveles de servicio de infraestructura	38
Figura 2-3	Java ES Componentes de los servicios del sistema	41
Figura 2-4	Dimensión 2: capas lógicas para aplicaciones de empresa distribuidas	44
Figura 2-5	Messaging Server: Ejemplo de arquitectura con capas	47
Figura 2-6	Diseño de disponibilidad utilizando nodos de Sun Cluster	51
Figura 2-7	Arquitectura lógica para el escenario de comunicaciones de la empresa	55
Figura 3-1	La entrada única de usuario es compatible con varios servicios	63
Figura 3-2	Secuencia de autenticación	65
Figura 3-3	Secuencia de autorización	66
Figura 4-1	Tareas del ciclo de vida de las soluciones	70
Figura 4-2	Especificación de un escenario de implementación	72
Figura 4-3	Conversión de un escenario de implementación en una arquitectura de implementación	73

Lista de tablas

Tabla 1	Documentación de Java Enterprise System	12
Tabla 2	Convenciones de tipos de letra	14
Tabla 1-1	Componentes de los servicios del sistema Java ES	22
Tabla 1-2	Componentes de disponibilidad de Java ES	24
Tabla 1-3	Componentes de acceso de Java ES	24
Tabla 1-4	Componentes administrativos de Java ES	25
Tabla 1-5	Categorías de usuario de Java ES para realizar tareas del ciclo de vida	29
Tabla 1-6	Consideraciones relacionadas con los distintos escenarios de adopción de Java ES	31
Tabla 2-1	Relaciones entre los componentes de los servicios del sistema de Java ES	42
Tabla 2-2	Calidades de servicio que afectan a la arquitectura de solución	48
Tabla 2-3	Componentes de calidad de servicio y calidades de sistema afectadas	50
Tabla 2-4	Resumen de requisitos de la empresa: escenario de comunicaciones	54

Prefacio

Sun Java™ Enterprise System Visión general técnica presenta las bases 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.

Este prefacio incluye los siguientes apartados:

- “Público” en la página 12
- “Uso de la documentación” en la página 12
- “Convenciones” en la página 14
- “Recursos Web” en la página 14
- “Comunicar problemas” en la página 15
- “Sun valora sus comentarios” en la página 15

Público

Java Enterprise System Technical Overview está concebida para individuos que vayan a diseñar, implementar o realizar el mantenimiento de 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.

Las personas que lean la *Java Enterprise System Technical Overview* deben estar familiarizadas 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

Uso de la documentación

Los manuales de Java Enterprise System se encuentran disponibles como archivos en línea en formato PDF y HTML. Los dos formatos se pueden leer mediante tecnologías de asistencia para usuarios con discapacidades. Se puede acceder al sitio Web de la documentación de Sun™ en la siguiente ubicación:

<http://docs.sun.com>.

Se puede acceder a la documentación de Java Enterprise System en:

<http://docs.sun.com/prod/entsys.05q1>

La siguiente tabla enumera los manuales sobre el sistema incluidos en el conjunto de documentación de Java Enterprise System. La columna de la izquierda especifica el nombre y la ubicación del número de pieza de cada documento y la columna de la derecha describe el contenido general del documento.

Tabla 1 Documentación de Java Enterprise System

Documento	Contenido
<i>Java Enterprise System Release Notes</i> http://docs.sun.com/doc/819-0057	Incluye la información más reciente acerca de Java Enterprise System, entre la que se encuentran los problemas conocidos. Además, cada componente tiene sus propias notas de la versión.

Tabla 1 Documentación de Java Enterprise System (*continúa*)

Documento	Contenido
<p><i>Java Enterprise System Documentation Roadmap</i> http://docs.sun.com/doc/819-0055</p>	Proporciona una descripción de la documentación relacionada con Java Enterprise System. Incluye enlaces a la documentación asociada a los componentes.
<p><i>Java Enterprise System Technical Overview</i> http://docs.sun.com/doc/819-0061</p>	Presenta los conceptos básicos técnicos y conceptuales de Java Enterprise System. Describe los componentes, la arquitectura, los procesos y las funciones.
<p><i>Guía de planificación de implementación de Java Enterprise System</i> http://docs.sun.com/doc/819-0058</p>	Proporciona una introducción a la planificación y diseño de soluciones de implementación empresariales basadas en Java Enterprise System. Presenta los conceptos básico y los principios de la planificación y diseño de la implementación, plantea el ciclo de vida de la solución y proporciona ejemplos y estrategias de alto nivel para utilizarlas cuando se planifican soluciones basadas en Java Enterprise System.
<p><i>Guía de administración de usuarios de Java Enterprise System</i> http://docs.sun.com/doc/817-5761</p>	Ayuda a planificar, implementar y administrar información acerca de los usuarios de su solución Java Enterprise System. Sirve de complemento a <i>Guía de planificación de implementación de Java Enterprise System</i> puesto que describe los problemas de administración de usuarios en cada etapa del ciclo de vida de la solución.
<p><i>Serie de ejemplos de implementación de Sun Java Enterprise System: Escenario de evaluación</i> http://docs.sun.com/doc/819-0059</p>	Describe cómo instalar Java Enterprise System en un equipo, establecer una serie de servicios de red básicos, compartidos y de núcleo, así como la forma de configurar cuentas de usuario que puedan acceder a los servicios configurados.
<p><i>Java Enterprise System Installation Guide</i> http://docs.sun.com/doc/819-0056</p>	Ofrece una guía a lo largo del proceso de instalación de Java Enterprise System para el sistema operativo Solaris™ o el sistema operativo Linux. Muestra cómo seleccionar los componentes que se instalarán, cómo configurarlos después de la instalación y cómo verificar que los componentes configurados funcionan adecuadamente.
<p><i>Guía de actualización y migración de Sun Java Enterprise System</i> http://docs.sun.com/doc/819-0062</p>	Proporciona la información e instrucciones para actualizar Java Enterprise System para el sistema operativo Solaris™ o el entorno operativo Linux.
<p><i>Java Enterprise System Glossary</i> http://docs.sun.com/doc/816-6873</p>	Define los términos utilizados en la documentación de Java Enterprise System.

Además de la documentación del sistema que figura en esta tabla, el conjunto de documentación de Java Enterprise System incluye documentos específicos para muchos de los componentes de Java Enterprise System. Consulte *Java Enterprise System Documentation Roadmap* para obtener más información.

Convenciones

La siguiente tabla muestra las convenciones de tipos de letra utilizadas en esta guía.

Tabla 2 Convenciones de tipos de letra

Tipos de letra	Significado	Ejemplos
AaBbCc123 (Espacio sencillo)	Elementos API y de idioma, etiquetas HTML, direcciones de sitios Web, nombres de comandos, nombres de archivos, rutas de directorios, texto que el equipo muestra en pantalla y código de ejemplo.	Edite el archivo <code>.login</code> . Utilice <code>ls -a</code> para ver todos los archivos. % Tiene un mensaje.
<i>AaBbCc123</i> (Cursiva)	Títulos de las guías. Palabras o términos nuevos. Palabras resaltadas. Las variables de la línea de comandos que se deben sustituir por nombres o valores reales.	Consulte el capítulo 6 de <i>User's Guide</i> . Se denominan opciones de <i>clase</i> . <i>Tiene</i> que ser superusuario para realizar esto. El archivo se encuentra en el directorio <code>dir_instalación/bin</code> .

Recursos Web

La siguiente ubicación incluye información acerca de Java Enterprise System y sus productos componentes:

<http://www.sun.com/software/javaenterprisesystem/index.html>

Comunicar problemas

Si experimenta problemas con Java Enterprise System, póngase en contacto con el servicio de atención al cliente de Sun, utilizando uno de los siguientes procedimientos:

- Servicios de atención al cliente en línea sobre el software de Sun en:

<http://www.sun.com/service/sunone/software>

Este sitio dispone de enlaces a la base de datos de soluciones, al centro de asistencia en línea y al rastreador de productos, así como a programas de mantenimiento y números de contacto de asistencia.

- El número de teléfono del distribuidor asociado al contrato de mantenimiento.

Para que podamos ayudarle de forma óptima en la resolución de problemas, tenga a mano la siguiente información cuando se ponga en contacto con el servicio de asistencia:

- Descripción del problema, incluida la situación en la que éste se produce y la forma en que afecta al funcionamiento
- Tipo de máquina, versión del sistema operativo y versión del producto, incluida cualquier revisión del producto y otro software que pudiera influir en el problema
- Pasos detallados de los métodos que haya usado para solucionar el problema
- Cualquier registro de error o volcado del núcleo

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 del documento y el número de pieza. El número de pieza consta de siete o de nueve dígitos y se encuentra en la página que contiene el título de la guía o al principio del documento. Por ejemplo, el título de esta guía es *Sun Java Enterprise System Technical Overview* y el número de pieza es 819-1924.

Sun valora sus comentarios

Introducción a Java Enterprise System

Sun Java™ Enterprise System (Java ES) es un conjunto de componentes de software que proporciona los servicios necesarios para brindar apoyo a aplicaciones empresariales distribuidas en una red o entorno 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 necesita Java Enterprise System?” en la página 18
- “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 necesita Java Enterprise System?

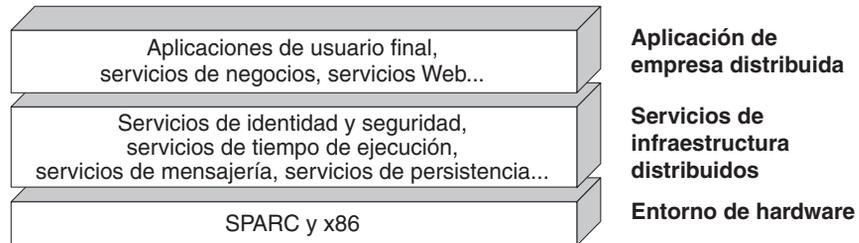
En la actualidad, los requisitos empresariales demandan soluciones de software distribuidas en la red o en entornos de Internet y que tengan altos niveles de rendimiento, disponibilidad, seguridad, escalabilidad y facilidad de mantenimiento.

Java Enterprise System proporciona los servicios de infraestructura necesarios para ser compatibles con funciones de *aplicación de empresa distribuida*, es decir, con aplicaciones que normalmente tienen las siguientes características:

- **Distribuidas.** La aplicación consta de *componentes* de software de interacción implementados en un entorno de red que pueden incluir sitios remotos geográficamente. Estos componentes distribuidos que se ejecutan en diversos equipos del entorno funcionan juntos para aportar funciones de negocios específicas a cada *usuario final* y a otras aplicaciones de negocios.
- **Fortaleza de empresa.** El ámbito y la escala de la aplicación satisfacen las necesidades de un entorno de producción o del 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 de *servicios* de infraestructuras subyacente que permita a los componentes distribuidos comunicarse entre sí, coordinar su trabajo, implementar un acceso seguro, etc. Estos servicios de infraestructura son compatibles, a su vez, con un entorno de hardware de equipos y enlaces 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 la 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 que los empleados, trabajadores desplazados, trabajadores cualificados, socios empresariales, proveedores y clientes accedan a los recursos corporativos desde cualquier punto dentro o 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 colaboración y comunicaciones.** 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.
- **Servicios de seguridad e identidad de red.** Mejoran la seguridad y la protección de los activos clave de la información corporativa, a la vez que garantizan la aplicación de las directivas de control de acceso adecuadas en todas las comunidades, las aplicaciones y los 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 aplicaciones y contenedor Web.** Estos servicios permiten que los componentes distribuidos se comuniquen entre sí en el tiempo de ejecución y son compatibles con el desarrollo, la implementación y la administración de las 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, escalabilidad, facilidad de mantenimiento y otras cualidades de 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 disponibilidad prácticamente continua para los componentes de aplicación y para los componentes de infraestructura que los respaldan.
- **Servicios de acceso.** Estos servicios proporcionan acceso mediante Internet o basado en explorador a los servicios de Java Enterprise System.
- **Servicios administrativos.** Estos servicios ayudan a mantener y optimizar el rendimiento de las aplicaciones respaldadas por Java Enterprise System.

Puede implementar uno o más servicios de Java Enterprise System y cada uno puede incluir una serie de componentes de Java Enterprise System.

Componentes de Java Enterprise System

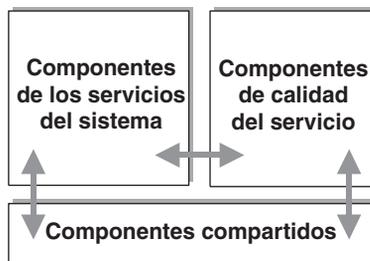
Java Enterprise System supone la integración en un único sistema de software de productos de software que anteriormente se ofrecían de forma independiente. Los componentes de este sistema se han probado conjuntamente para garantizar su interoperabilidad. Se ha logrado su integración gracias a una serie de funciones del sistema:

- Todos los componentes están sincronizados en un conjunto común de bibliotecas compartidas.
- Todos los componentes de Java Enterprise System se instalan con un único programa de instalación.
- Todos los componentes de Java Enterprise System comparten una identidad de usuario integrada y un sistema de administración de 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. Cada *componente del sistema* puede pertenecer a una de las tres categorías principales, 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 las aplicaciones empresariales distribuidas.
- **Componentes de calidad del servicio.** Estos componentes mejoran la disponibilidad, seguridad, escalabilidad y 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 Enterprise System



Componentes de los servicios del sistema

Una serie de componentes de Java Enterprise System proporcionan los servicios principales que dan respaldo a las soluciones de software distribuidas. Cada *servicio de sistemas* incluye servicios de portal, servicios de comunicación y colaboración, servicios de identidad y seguridad, servicios de contenedores Web y servicios de aplicaciones J2EE.

En la siguiente tabla se describe cada *componente de los servicios del sistema* que proporciona estos servicios distribuidos. También se describen los servicios en sí. Un componente de los servicios del sistema es un proceso de servidor con varios subprocesos capaz de admitir una gran cantidad de clientes. Para obtener más información sobre algún componente, consulte [“Descripciones de los componentes de servicios de sistemas” en la página 82.](#)

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

Componente	Servicios de sistema proporcionados
Sun Java System Access Manager	Incluye servicios de administración de acceso y de administración de identidades digitales. Los servicios de administración de acceso incluyen autenticación (incluido el inicio de sesión único) y autorización basada en roles para acceder a aplicaciones o 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 incluye servicios de contenedor Web.
Sun Java System Calendar Server	Incluye servicios de programación y de calendario para usuarios finales y grupos de usuarios finales. Calendar Server incluye un cliente basado en explorador 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.

Tabla 1-1 Componentes de los servicios del sistema Java ES (*continúa*)

Componente	Servicios de sistema proporcionados
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	Incluye mensajería asíncrona y fiable entre aplicaciones y componentes distribuidos y acoplados libremente. Message Queue implementa la especificación API de Java Message Service (JMS) y agrega funciones de empresa como seguridad, escalabilidad y administración remota.
Sun Java System Messaging Server	Proporciona mensajería segura, fiable de gran capacidad de almacenamiento y reenvío compatible con correo electrónico, fax, buscapersonas, voz y vídeo. Messaging Server puede acceder a la vez a varios almacenes de mensajes y proporciona un filtro de contenido para rechazar el correo electrónico no solicitado y evitar los ataques de virus.
Sun Java System Portal Server	Incluye servicios de portal importantes, como la adición y personalización de contenido, para los clientes basados en explorador que obtienen acceso a servicios o aplicaciones de negocios. Portal Server también ofrece un motor de búsqueda que permite su configuración.
Sun Java System Web Server	Proporciona servicios de contenedor Web J2EE™ para los componentes Web de Java, como Java Servlet y JavaServer Pages™ (JSP™). Web Server también es compatible con otras tecnologías de aplicaciones Web para ofrecer contenido Web estático y dinámico, como secuencias CGI y páginas de servidor activas del sistema Java.

Componentes de calidad del servicio

Además de los componentes de los servicios del sistemas que se muestran en la [Tabla 1-1](#), Java Enterprise System incluye una serie de componentes que se utilizan para mejorar la calidad de los servicios ofrecida por los componentes de los servicios del sistema o los componentes de aplicaciones personalizados. Un *componente de calidad del servicio* se puede clasificar en una de 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 servicio del sistema y los componentes de aplicaciones. Los componentes de disponibilidad incluidos en Java Enterprise System y los servicios que proporcionan se muestran en la siguiente tabla. Para obtener más información sobre algún componente, consulte [“Descripciones de los componentes de disponibilidad” en la página 87.](#)

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 infraestructura de Java Enterprise System y el entorno de hardware donde se implementan.
Almacén de sesión de alta disponibilidad	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 información sobre algún componente, consulte [“Descripciones de componentes de acceso” en la página 88.](#)

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 un control de acceso a los directorios, una compatibilidad de esquemas, un filtro de atributos y enrutamiento mejorados para múltiples instancias de Directory Server.
Sun Java System Portal Server Secure Remote Access	Proporciona acceso seguro a Internet desde fuera de un servidor de seguridad corporativo a servicios y contenido de Portal Server, incluidos portales internos y aplicaciones de Internet.

Tabla 1-3 Componentes de acceso de Java ES (*continúa*)

Componente	Servicios de acceso proporcionados
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 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 información sobre algún componente, consulte [“Descripciones de componentes administrativos” en la página 90](#).

Tabla 1-4 Componentes administrativos de Java ES

Componente	Servicios administrativos proporcionados
Sun Java System Administration Server (y Console)	Proporciona una herramienta de administración gráfica que permite configurar y administrar Directory Server y Messaging Server.
Herramienta de preparación de Sun Java System Directory	Proporciona una secuencia de comandos para configurar Directory Server con el esquema necesario para ofrecer Messaging Server y Calendar Server a los usuarios.
Sun Java System Delegated Administrator	Proporciona herramientas de interfaz gráfica de usuario y de línea de comandos para rellenar las entradas de usuario en Directory Server con los atributos de usuario que Messaging Server y Calendar Server necesitan.
Sun Remote Services Net Connect	Proporciona capacidad de supervisión remota.

Componentes compartidos

Java Enterprise System incluye una serie de bibliotecas compartidas instaladas localmente de las que dependen muchos componentes de los servicios del sistema y componentes de calidad del servicio. Un *componente compartido* de Java Enterprise System ofrece servicios locales a los componentes de Java Enterprise System 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 una lista completa, consulte [“Componentes compartidos” en la página 92](#).

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 alcance y dificultad en función del punto de inicio para la adopción de Java Enterprise System y en la naturaleza de la solución que esté intentando 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 soluciones de Java Enterprise System

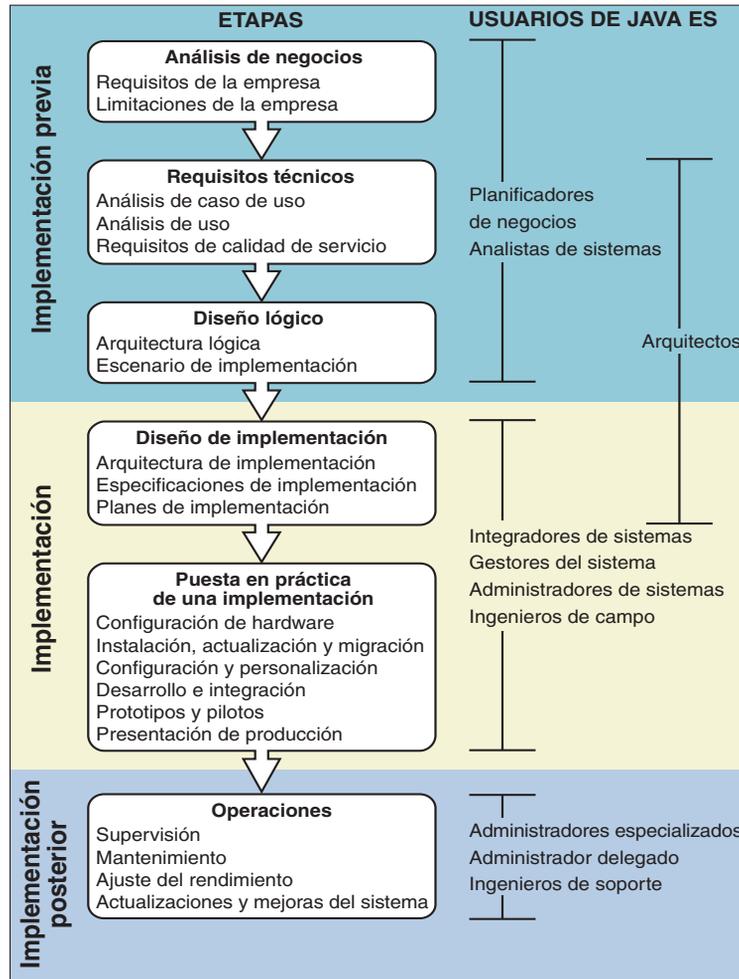
Las tareas implicadas en la creación de soluciones empresariales basadas en el software de Java Enterprise System se pueden dividir en varias fases, tal y como se muestra en la [Figura 1-3 en la página 28](#). La ilustración también muestra la categoría del usuario de Java Enterprise System que generalmente realiza las distintas tareas.

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

- **Implementación previa.** 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, el escenario de implementación se traslada a 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.
- **Implementación posterior.** en la etapa de operaciones, se ejecuta una solución implementada en las condiciones de producción y se supervisa y optimiza su rendimiento. La solución implementada también se actualiza para incluir las nuevas funciones según sea necesario.

El ciclo de vida de la solución y las tareas de cada una de las etapas que se muestran en la [Figura 1-3](#) se presentan con más detalle en el [Capítulo 4, “Tareas del ciclo de vida de las soluciones de Java Enterprise System”](#).

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



La [Figura 1-3](#) muestra los usuarios de Java Enterprise System que normalmente realizan las tareas que se muestran para las etapas del ciclo de vida. Si está trabajando con Java Enterprise System, su trabajo debe ajustarse a una o varias de las categorías de usuario que aparecen en la [Figura 1-3](#). La siguiente tabla describe las capacidades e información de cada categoría de usuario.

Tabla 1-5 Categorías de usuario de Java ES para realizar tareas del ciclo de vida

Usuario	Capacidades y formación	Etapas
Organizador de negocios	Conocimiento general, en lugar de un conocimiento técnico en profundidad.	Análisis de negocios Requisitos técnicos
Analista de sistemas	Comprende la dirección estratégica de la empresa. Conoce los procesos, objetivos y requisitos empresariales.	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
Integrador de sistemas	Conocimiento altamente técnico.	Diseño de implementación
Ingeniero de campo	Está íntimamente familiarizado con los entornos de las tecnologías de la información.	Implementación de despliegue
Administrador de sistemas	Tiene experiencia en la implementación de soluciones de software distribuidas.	
Gestor del sistema	Conoce la arquitectura, los protocolos, los dispositivos y la seguridad de redes. Conoce los lenguajes de secuencias de comandos y de programación.	
Administrador de sistemas especializado	Conocimiento técnico especializado o del producto.	Operaciones
Administrador delegado	Familiarizado con hardware, plataformas, directorios y bases de datos.	
Ingeniero de soporte	Capacidad de supervisión, solución de problemas y actualización de software. Conoce la administración del sistema para las plataformas del sistema operativo.	

Java Enterprise System Escenarios de adopción

Las necesidades empresariales que impulsan la adopción de Java Enterprise System varían significativamente. Sin embargo, el objetivo de alto nivel para prácticamente todas las implantaciones de Java Enterprise System se encuadra dentro de un *escenario de adopción* de entre los que aparecen a continuación:

- **Nuevo sistema.** Comenzando con un sistema de software que no existe, el usuario implementa el software de Java Enterprise System para dar respaldo a una nueva solución empresarial.

- **Mejora.** Comenzando con una infraestructura de tecnología de la información (TI) existente, el usuario sustituye una o varias partes de dicho sistema por el software de Java Enterprise System. 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.** Comenzando con una infraestructura de TI, el usuario implementa el software de Java Enterprise System que no está instalado actualmente en el 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 Enterprise System, o autenticación y autorización Java para los servicios existentes.
- **Actualización.** Comenzando con una infraestructura de TI que consta de una versión anterior de Java Enterprise System o de productos de Sun anteriores a Java Enterprise System, el usuario puede actualizar Java Enterprise System a la versión más actual.

Cada escenario de adopción cuenta con sus propios retos y consideraciones. Independientemente del escenario de adopción que se ajuste a su situación, el proceso del ciclo de vida de la solución que se muestra en la [Figura 1-3](#) es válido. 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 actualización de la infraestructura existente con el nuevo software requiere a menudo la migración de los datos desde el sistema existente al 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 un nuevo software a un sistema existente o la sustitución de sistemas secundarios de software a menudo requieren que integre nuevos componentes de software con los sistemas secundarios 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.** Prácticamente, cualquier cambio en la infraestructura implica cambios en los procedimientos de TI y en los conocimientos. El departamento de TI debe tener el tiempo suficiente para adquirir los nuevos conocimientos o transferir los antiguos, para brindar soporte a las tecnologías de Java Enterprise System.
- **Hardware.** Cuando sustituye o mejora un sistema o un sistema secundario existente, las limitaciones empresariales pueden requerir que reutilice el 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 Enterprise System.

Tabla 1-6 Consideraciones relacionadas con los distintos 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*
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

* 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, en el que se indica el sistema de software disponible y el objetivo que se intenta conseguir. Existen cuatro escenarios básicos de adopción de Java Enterprise System: nuevo sistema, sustitución, ampliación y actualización.

componente Unidad de lógica de software desde la que se crean aplicaciones distribuidas. Un componente puede ser un *componente del sistema* incluido en Java Enterprise System o un *componente de aplicación* personalizado. Normalmente, un componente de aplicación cumple con un modelo de componentes distribuido (como CORBA y la plataforma J2EE™) y realiza alguna función informática específica. Estos componentes, individuales o combinados, ofrecen *servicio de negocios* y pueden encapsularse como *servicio Web*.

aplicación de empresa distribuida Aplicación cuya lógica se aplica a un entorno de red o de Internet (el aspecto distribuido) y cuyo ámbito y escala satisfacen las necesidades de un entorno de producción o de un proveedor de servicios (el aspecto de empresa).

usuario final Una persona que utiliza una aplicación distribuida, a menudo mediante una interfaz gráfica de usuario, como un explorador de Internet o una interfaz gráfica de usuario de un dispositivo móvil. El número de usuarios finales simultáneos admitidos por una aplicación es un factor importante de la *arquitectura de implementación* de la aplicación.

servicio Función de software realizada para un *cliente* o para varios. Esta función podría ser en un nivel muy bajo, como una administración de memoria o, en un nivel alto, como un *servicio de negocios* de comprobación de crédito. Un servicio de alto nivel puede estar formado por un conjunto de servicios individuales. Los servicios pueden ser locales (disponibles para clientes locales) o distribuidos (disponibles para clientes remotos).

componente de calidad del servicio Un *componente del sistema* de los incluidos en Java Enterprise System. Admite componentes, que incluyen componentes de acceso y administrativos, y que ofrece respaldo para usar un *componente de los servicios del sistema*.

componente compartido Un *componente del sistema* de los incluidos en Java Enterprise System. Los componentes compartidos, normalmente bibliotecas, proporcionan servicios locales a otros componentes del sistema. Por contraste, un *componente de los servicios del sistema* incluye servicios de infraestructura distribuidos a otros componentes del sistema (o a un *componente de aplicación*).

componente del sistema Cualquier paquete de software o conjunto de paquetes incluido en Java Enterprise System e instalado mediante el programa de instalación de Java Enterprise System. Existen varios tipos de componentes del sistema: *componente de los servicios del sistema*, que proporciona un *servicio* de infraestructura distribuido; *componente de calidad del servicio*, que admite los componentes de los servicios del sistema proporcionando servicios de acceso y administrativos; y *componente compartido*, que proporciona servicios locales a otros componentes del sistema.

servicio de sistemas Un *servicio* distribuido (o varios) que definen la funcionalidad única proporcionada por Java Enterprise System. Los servicios del sistema normalmente requieren la compatibilidad de varios de los siguientes elementos: *componente de calidad del servicio*, *componente compartido*, ya sea individualmente o un conjunto de ambos.

componente de los servicios del sistema Un *componente del sistema* de los que se incluyen en Java Enterprise System. Los componentes de los servicios del sistema proporcionan los principales servicios de infraestructura de Java Enterprise System: servicios de portal, servicios de comunicación y colaboración, servicios de identidad y seguridad, servicios de contenedores Web y servicios de aplicaciones y servicios de disponibilidad.

Términos clave de este capítulo

Arquitecturas de soluciones de Java Enterprise System

Este capítulo ofrece una introducción de los conceptos arquitectónicos sobre los que se basan las soluciones de Java Enterprise System (Java ES). El capítulo muestra cómo los componentes de Java ES, tanto los componentes de los servicios del sistema como los componentes de calidad de servicio, se utilizan para admitir las soluciones de empresa distribuidas.

La *arquitectura* de soluciones de Java ES tiene 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 describe la asignación de la arquitectura lógica a un entorno informático físico. Los componentes de Java ES desempeñan unas funciones importantes en la arquitectura lógica y 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 36
- “Ejemplo de arquitectura de soluciones de Java Enterprise System” en la página 53
- “Términos clave de este capítulo” en la página 56

Marco arquitectónico de Java Enterprise System

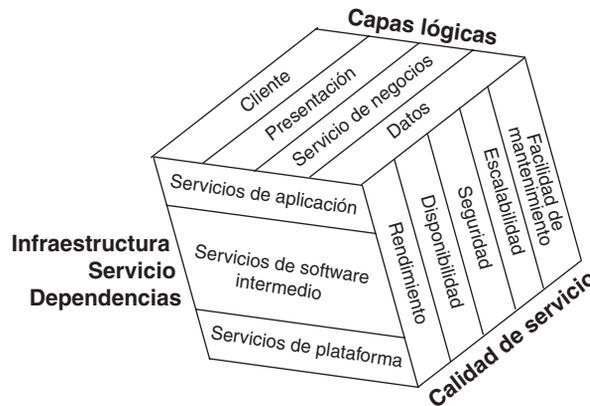
Los componentes de Java Enterprise System admiten la implementación de soluciones de software de fortaleza de empresa distribuidas.

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 servicios de infraestructura.** Esta dimensión se centra en el papel de los componentes del servicio del sistema (consulte [“Componentes de los servicios del sistema” en la página 22](#)) en el apoyo a las soluciones distribuidas.
- **Capas lógicas.** Esta dimensión se centra en la independencia física y lógica de las componentes de soluciones con el fin de implementarlos en una red o en un entorno de Internet.
- **Calidad de servicio.** Esta dimensión se centra en cómo se cumplen los requisitos de calidad de servicio, como disponibilidad, seguridad, escalabilidad y facilidad de mantenimiento, incluida la función de los componentes de calidad de servicio (consulte [“Componentes de calidad del servicio” en la página 23](#)).

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

Figura 2-1 Dimensiones de la arquitectura de soluciones de Java ES

Juntas, estas tres dimensiones representan un único marco que incorpora las relaciones entre los siguientes elementos: componente de software, *componente de aplicación* y componente 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 servicio de infraestructura

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

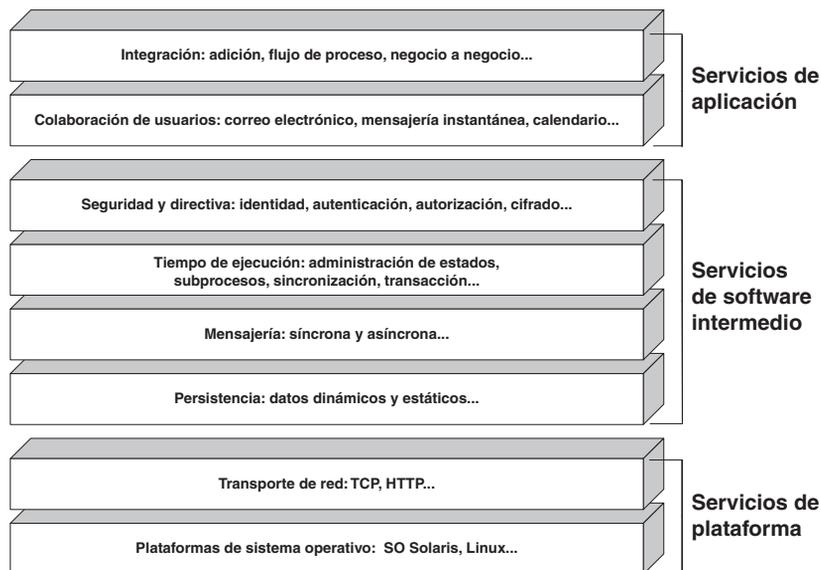
Al diseñar un sistema de software distribuido, tanto si está compuesto por componentes personalizados o se trata de componentes estándar de Java ES, deberá incorporar una serie de servicios de infraestructura. Estos servicios operan en varios niveles.

La dimensión de dependencia de servicios de infraestructura de la arquitectura de la solución se ilustra en la [Figura 2-2 en la página 38](#). Los niveles que se muestran en esta figura son una vista ampliada de la capa de servicios de infraestructura que aparece en la [Figura 1-1 en la página 19](#).

La jerarquía de los servicios que aparece en la [Figura 2-2](#) y las dependencias entre ellas constituyen una dimensión importante de la arquitectura lógica de una solución. Estos servicios de infraestructura proporcionan la base conceptual 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 de la [Figura 2-2](#) se dividen en tres grandes grupos: servicios de plataforma de nivel inferior, servicios de aplicación de nivel superior y un grupo de servicios de software intermedio, denominados así por su ubicación entre los otros dos grupos.

Figura 2-2 Dimensión 1: Niveles de servicio de infraestructura



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 inferior al superior, tal y como se muestra en la [Figura 2-2](#):

- **Plataformas de sistema operativo.** Proporciona asistencia 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.** Incluye compatibilidad de red básica para la comunicación entre componentes de aplicación distribuidos que se ejecutan en varios 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.** Compatible para obtener acceso a datos estáticos (como información de usuario, directorio o configuración) y datos de aplicación dinámicos (información que se actualiza frecuentemente) y almacenarlos.
- **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, los contenedores de EJB™, Web y beans controlados por mensajes en un servidor de aplicación o en un servidor Web ofrecen estos servicios de tiempo de ejecución.
- **Seguridad y directiva.** Compatible con el acceso seguro a recursos de aplicaciones. Estos servicios son compatibles con directivas que controlan el acceso basado en roles o en grupos a recursos distribuidos, así como 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 usuario.** 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.** Ofrece los servicios que juntan los servicios de empresa 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.

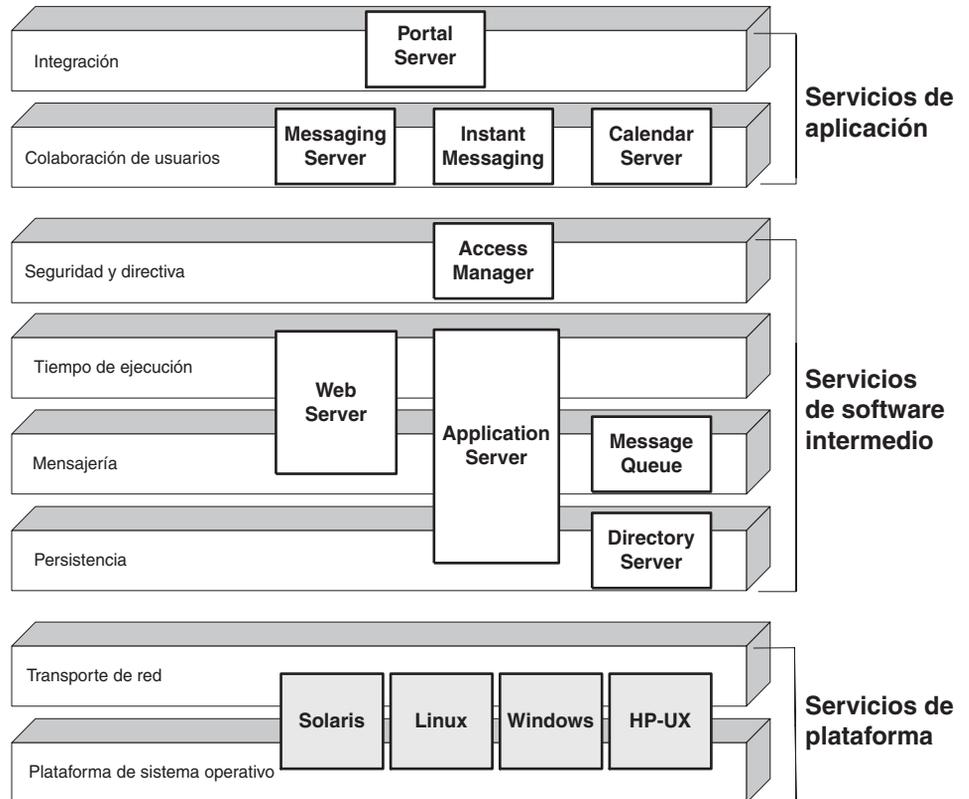
Los niveles de servicios de la [Figura 2-2](#) reflejan una dependencia general de varios servicios de infraestructura entre sí, desde los servicios de sistema operativo de nivel inferior a los servicios de aplicación e integración de nivel superior. Cada servicio depende normalmente de servicios de niveles inferiores y respalda el funcionamiento de servicios superiores.

La [Figura 2-2](#), sin embargo, no representa una división 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 Java ES implementan los niveles de servicio de infraestructura distribuida que se muestran en la [Figura 2-2](#). La posición de los componentes de los servicios del sistema de Java ES en los distintos niveles se muestra en la [Figura 2-3](#).

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



NOTA Las plataformas del sistema operativo que se muestran en la [Figura 2-3](#) no son parte formal de Java Enterprise System; sin embargo, se incluyen para mostrar las plataformas de sistema operativo compatibles con los componentes de Java ES.

Dependencias de servicio de infraestructura de Java Enterprise System

En general, cada componente de servicios del sistema de Java ES que se muestra en la [Figura 2-3](#) depende de los componentes que se encuentran debajo de él en la infraestructura y respalda el funcionamiento de los componentes que se encuentran por encima. Estas relaciones de dependencia y compatibilidad son un factor clave para diseñar las arquitecturas lógicas.

La [Tabla 2-1](#) muestra las relaciones específicas entre los componentes de los servicios del sistema de Java ES, mostrados de arriba a abajo, tal y como se muestran 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 notificaciones de correo electrónico) Portal Server (para canal de mensajería)
Instant Messaging	Directory Server Access Manager (para el inicio de sesión único)	Portal Server (para canal de mensajería instantáneo)
Calendar Server	Directory Server Messaging Server (para servicio de notificaciones de correo electrónico) Access Manager (para el inicio de sesión único)	Portal Server (para canal de calendario)

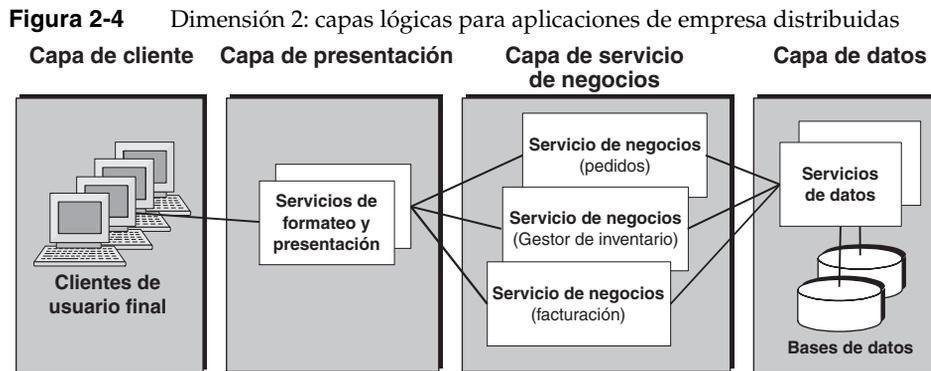
Tabla 2-1 Relaciones entre los componentes de los servicios del sistema de Java ES *(continúa)*

Componente	Depende de	Respalda el funcionamiento de
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
Application Server	Message Queue Directory Server (para objetos administrados)	Portal Server Access Manager
Message Queue	Directory Server (para objetos administrados)	Application Server
Web Server	Access Manager (para control de acceso)	Portal Server Access Manager
Directory Server	Ninguno	Portal Server Calendar Server Messaging Server Instant Messaging Access Manager

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 lógica y física de los componentes de software, basándose en la naturaleza de los servicios que proporcionan.

La dimensión de capa lógica de la arquitectura de soluciones se ilustra en la siguiente figura.



En su mayor parte, las arquitecturas de capas lógicas representan la capa de aplicaciones de empresas distribuidas de la [Figura 1-1 en la página 19](#). Los componentes de los servicios del sistema de Java ES descritos en “[Niveles de servicio de infraestructura](#)” en la [página 37](#) proporcionan respaldo a los componentes de aplicaciones en 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 a nivel de aplicaciones, como Messaging Server y Calendar Server.

Descripción de capas lógicas

Esta sección describe brevemente las cuatro capas lógicas que aparecen en la [Figura 2-4](#). Las descripciones hacen referencia a componentes de aplicaciones implementados con el modelo de 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 exploradores, componentes de Java que se ejecutan en un equipo de escritorio o clientes móviles de Java 2 Platform, Micro Edition (plataforma J2ME™) que se ejecutan en 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 de la capa de presentación normalmente consta de componentes de J2EE como, por ejemplo, componentes de Java Servlet o componentes de JSP que preparan datos para el envío en formato HTML o XML o que reciben solicitudes de procesamiento. Esta capa también puede incluir un servicio de portal que proporcione acceso personalizado y seguro a un *servicio de negocios* de la capa de servicio de negocios.
- **Capa de servicios de negocios.** La capa de servicio de negocios está formada por 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. Normalmente, esta capa consta de componentes bien acoplados que se ajustan al modelo de componente distribuido J2EE, como los objetos Java, los componentes EJB o los beans controlados por 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 montajes de servicios pueden encapsularse en forma de *servicio Web* acoplados libremente que en un modelo de arquitectura orientado al servicio y que se ajustan a los estándares de interfaz de SOAP (Simple Object Access Protocol). Un servicio de negocio también pueden crearse como un *servidor* independiente, como, por ejemplo, un servidor de calendario de empresa o de mensajería.
- **Capa de datos.** La capa de datos consta de servicios que ofrecen datos persistentes utilizados por la lógica de negocios. Los datos pueden ser 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 muestra en la [Figura 2-4 en la página 44](#) destaca la independencia lógica y física de los componentes, representada por cuatro 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.

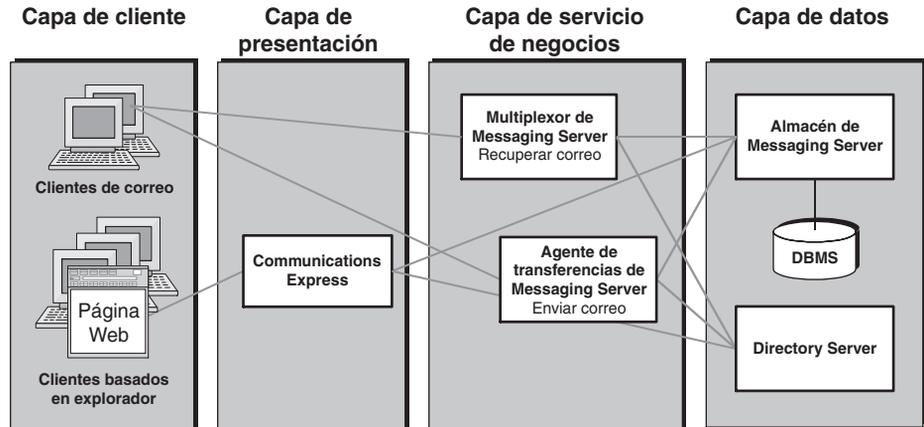
La forma en que se asignan los componentes de aplicaciones o los componentes de infraestructura en un entorno de hardware (es decir, la arquitectura de implementación) depende de varios factores, en función de la escala y la complejidad de la solución de software. Para implementaciones muy pequeñas, 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

Como se muestra en la [Figura 2-3 en la página 41](#), los componentes de los servicios de infraestructura de Java ES proporcionan la compatibilidad de infraestructura subyacente para las 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 por correo electrónico que proporciona Messaging Server se implementan utilizando una serie de configuraciones distintas lógicamente de Messaging Server. Estas configuraciones distintas proporcionan un conjunto de servicios distintos. 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 con capas



NOTA La [Figura 2-3](#) no tiene como objetivo representar una arquitectura lógica completa; un número de componentes de Java ES se han omitido 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 la flexibilidad necesaria para satisfacer los requisitos de calidad del servicio (consulte "[Dimensión 3: calidad de servicio](#)"). Por ejemplo, proporciona diversas soluciones de disponibilidad para distintas instancias y distintas implementaciones de seguridad para funciones de Messaging Server diferentes.

Dimensión 3: calidad de 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 de 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 de accesibilidad a servicios y recursos de un sistema para usuarios finales (el <i>tiempo de actividad</i> de un sistema).
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.

Tabla 2-2 Calidades de servicio que afectan a la arquitectura de solución (*continúa*)

Calidades de servicio del sistema	Descripción
Función latente	La capacidad de un sistema para gestionar el uso de carga máxima inusual sin recursos adicionales.
Facilidad de mantenimiento	La facilidad con que un sistema implementado puede mantenerse, incluidas tareas 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 de servicio de Java ES, presentados en [“Componentes de calidad del servicio” en la página 23](#), se resumen de la siguiente manera:

- **Componentes de disponibilidad.** Estos componentes proporcionan un tiempo de actividad prácticamente continuo de 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 facilidad de mantenimiento de los componentes del sistema.

La siguiente tabla muestra los componentes de calidad de servicio de Java ES más importantes desde una perspectiva arquitectónica con las calidades de sistema a los 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
Sun Remote Services Net Connect	Facilidad de mantenimiento

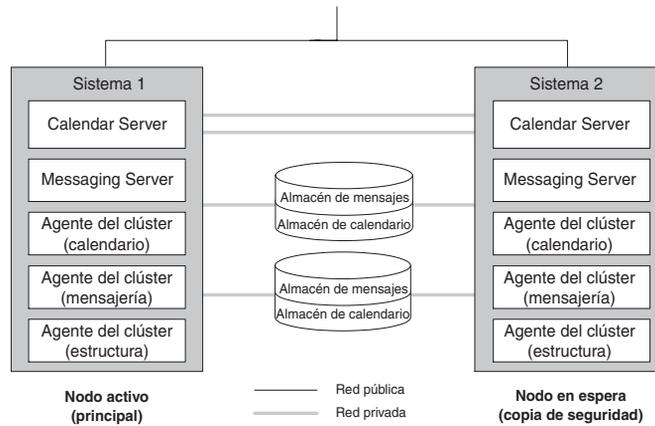
Software de Sun Cluster

El software de Sun Cluster incluye servicios de alta disponibilidad y escalabilidad para componentes de Java ES y para aplicaciones compatibles con la infraestructura de Java ES.

Un clúster es un conjunto de equipos acoplados libremente que ofrecen de forma conjunta una única visión de los servicios, recursos de sistemas y datos de cliente. 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 la salud 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 siguiente figura se muestra un clúster de dos nodos para dar apoyo a los servicios de almacenamiento de datos de Messaging Server y Calendar Server.

Figura 2-6 Diseño de disponibilidad utilizando nodos de Sun Cluster

Los paquetes de servicios de datos de Sun Cluster (en ocasiones denominados agentes de Sun Cluster) están disponibles para todos los componentes del servicio de sistemas de 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 incluir alta disponibilidad y escalabilidad en una aplicación de empresa distribuida.

Debido a la redundancia necesaria para admitir los entornos de Sun Cluster, la inclusión de Sun Cluster en una solución aumenta sustancialmente el número de equipos y enlaces 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 peer-to-peer distribuidos. Por tanto, el software de Sun Cluster debe instalarse en cada equipo en un clúster.

Síntesis de las tres dimensiones arquitectónicas

Cuando se ven en su conjunto, las tres dimensiones arquitectónicas que se muestran en la [Figura 2-1](#) y que se han presentado en las secciones anteriores proporcionan un marco para diseñar soluciones de software distribuidas. Las tres dimensiones (dependencias de servicios de infraestructura, capas lógicas y calidad de 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 de 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 componente es vital; 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 de arquitecturas de implementación de soluciones.

Queda fuera del objetivo de esta guía describir las metodologías de diseño detalladas basadas en el marco arquitectónico de la [Figura 2-1](#). 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 soluciones de Java Enterprise System

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

Gran cantidad de soluciones se pueden diseñar e implementar directamente de forma estándar, sin necesidad ningún esfuerzo de desarrollo, utilizando los componentes que se incluyen 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 cumplen con los estándares de interfaz de SOAP (Simple Object Access Protocol). Gran cantidad de soluciones implican una combinación de estos dos planteamientos.

Esta sección muestra un ejemplo que demuestra cómo Java Enterprise System admite una solución estándar, partiendo de los conceptos arquitectónicos explicados en 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.

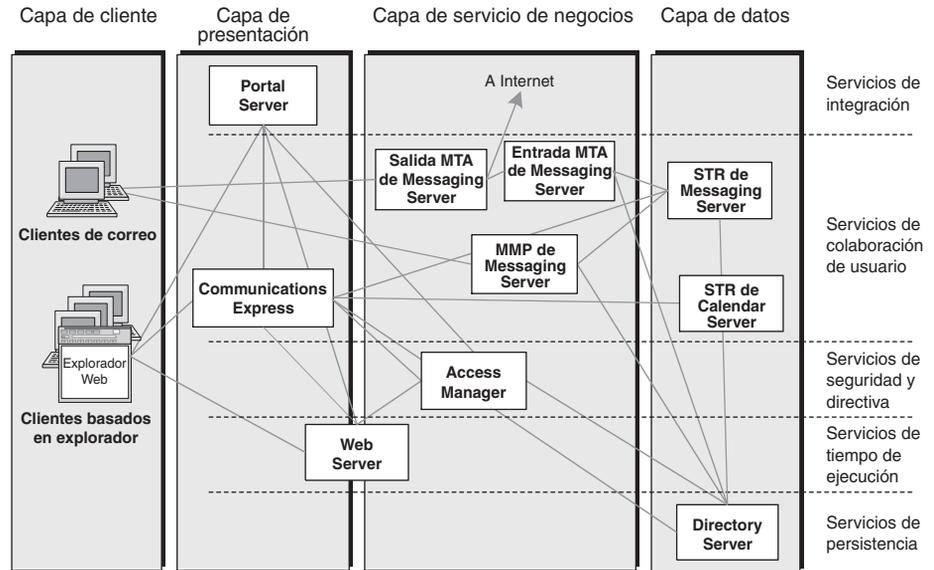
Tabla 2-4 Resumen de requisitos de la empresa: escenario de comunicaciones

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

Una arquitectura lógica para ofrecer los servicios de portal, comunicación e identidad especificados en la [Tabla 2-4](#) utilizando los componentes de Java ES que se muestran en la siguiente figura. 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 almacenada en Directory Server, es el árbitro de la autenticación y autorización de inicio de sesión único para Portal Server y 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, componentes de envío y recuperación en la capa de servicios de negocios y un componente de acceso HTTP y Communications Express en la capa de presentación.

La arquitectura lógica también muestra las dependencias de servicios de infraestructura entre los distintos componentes de Java ES. Portal Server, por ejemplo, depende de Communications Express para sus canales de mensajería y de calendario, y 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. Una serie de componentes requieren los servicios de contenedores Web proporcionados por Web Server.

Si desea obtener más información acerca del diseño lógico de soluciones de Java ES, consulte *Guía de planificación de implementación de Java Enterprise System*.

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 de varios maestros se puede utilizar para Directory Server.

Si desea obtener más información acerca del diseño de implementación de soluciones de Java ES, consulte *Guía de planificación de implementación de Java Enterprise System*.

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.

componente de aplicación Un *componente* de software personalizado que realiza alguna función informática específica que ofrece funciones de *servicio de negocios* a un *usuario final* o a otro componente de aplicación. Normalmente, un componente de aplicación cumple con un modelo de componentes distribuido (como CORBA y la plataforma J2EE™). Estos componentes, individuales o combinados, se pueden encapsular en forma de *servicio Web*.

arquitectura Un diseño que muestra los bloques de construcción física y lógica de una aplicación distribuida (u otro sistema de software) y las relaciones que se establecen entre sí. En el caso de una *aplicación de empresa distribuida*, el diseño arquitectónico normalmente incluye la *arquitectura lógica* y la *arquitectura de implementación* de la aplicación.

servicio de negocios Un *componente de aplicación* o montaje de componentes que realiza la lógica de negocios en nombre de varios clientes (y, por lo tanto, es un multiproceso). Un servicio de negocios también puede ser un montaje de componentes distribuidos encapsulados como *servicio Web* o puede ser un *servidor* independiente.

cliente Software que solicita un *servicio* de software. (Nota: no es una persona, consulte *usuario final*.) Un cliente puede ser un servicio que solicita otro servicio o un componente de GUI al que obtiene acceso un usuario final.

arquitectura de implementación Diseño de alto nivel que describe la asignación de una *arquitectura lógica* 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.

arquitectura lógica Diseño que describe los bloques de construcción lógicos de una aplicación distribuida y las relaciones (o interfaces) que se establecen entre ellos. La arquitectura lógica incluye funciones de *componente de aplicación* distribuidas y los componentes de los servicios de infraestructura necesarios para su compatibilidad.

servidor Proceso de software con varios subprocessos (en oposición a un servidor de hardware) que proporciona un *servicio* distribuido o un conjunto unido de servicios para cada *cliente* que accede al servicio mediante una interfaz externa.

servicio Web Servicio que se ajusta a los protocolos de Internet estandarizados para la accesibilidad, la encapsulación de servicio y el descubrimiento. Los estándares incluyen el protocolo de mensajería SOAP (Simple Object Access Protocol), la definición de interfaz WSDL (Web Service Definition Language) y el estándar de registro UDDI (Universal Discovery, Description, and Integration).

Términos clave de este capítulo

Funciones de integración de Java Enterprise System

Este capítulo proporciona una base conceptual y técnica para comprender las funciones que desempeñan un papel 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:

- “Programa de instalación integrado de Java Enterprise System” en la página 60
- “Identidad integrada y servicios de seguridad” en la página 62
- “Términos clave de este capítulo” en la página 67

Programa de instalación integrado de Java Enterprise System

Todos los componentes de Java ES se instalan usando un único programa de instalación. Este programa de instalación proporciona procedimientos de instalación y desinstalación y comportamientos coherentes en todos los componentes.

El programa de instalación de Java ES es un marco de trabajo integrado que transfiere el software de Java ES a un sistema host. El programa de instalación le permite seleccionar e instalar cualquier número de componentes de Java ES en cualquier equipo del entorno informático. El programa de instalación también proporciona una configuración de instalación en el tiempo, en función de los componentes de Java ES que se estén instalando.

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

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

Esta sección describe los siguientes aspectos del programa de instalación integrado de Java ES (para obtener información detallada consulte la *Java Enterprise System Installation Guide*):

- [Comprobación del software previo](#)
- [Comprobación de relaciones de dependencia](#)
- [Configuración inicial](#)
- [Desinstalación](#)

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.

Igualmente, el programa de instalación comprueba los componentes compartidos de Java ES (consulte [“Componentes compartidos” en la página 26](#)), como J2SE o NSS, que ya están instalados. 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 programa de instalación realiza una comprobación exhaustiva de los componentes para verificar que los componentes seleccionados funcionarán correctamente.

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 realizar 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 ofrece 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 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 identidades de usuario y su marco de autenticación y autorización integrado.

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

- [Identidad única](#)
- [Autenticación e inicio de sesión único](#)

Identidad única

Dentro de un entorno de Java ES, cada usuario final tiene su identidad única integrada. En función de esta *identidad única*, un usuario puede obtener acceso a varios recursos, como un portal, páginas Web y a servicios como mensajería, calendario y mensajería instantánea.

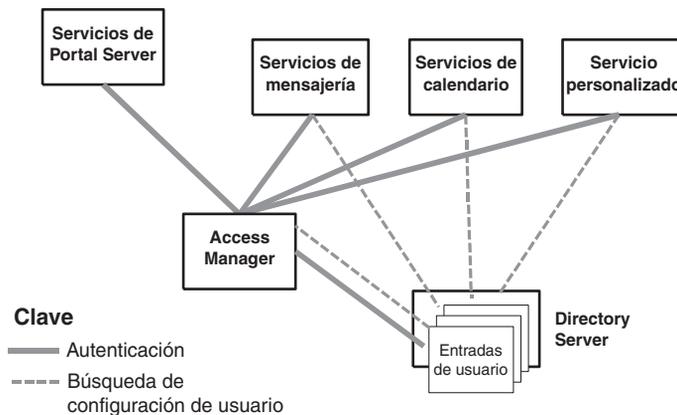
Esta función de identidad y seguridad integradas se basa en una estrecha colaboración entre Directory Server, Access Manager y otros componentes de Java ES.

El acceso del usuario a un servicio o recurso de Java ES se consigue almacenando información específica del usuario en una única entrada de usuario en un repositorio de usuario o en un *directorio*. 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 usuario se almacenan en un directorio que proporciona 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 autorizarle el 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 capacidad, conocida como *inicio de sesión único*, es una función potente que ofrece Java Enterprise System.

Autenticación e inicio de sesión único

Los servicios de autenticación y autorización de Java ES son ofrecidos por Access Manager. Access Manager usa información de Directory Server para mediar 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 políticas. El agente de políticas se conecta al servidor Web que alberga un servicio o recurso asegurado por Access Manager. El agente de políticas intercede en nombre de Access Manager en solicitudes que realizan los usuarios a los recursos asegurados. Para algunos componentes de Java ES, como Portal Server y Communications Express, la funcionalidad del agente de políticas se proporciona mediante un componente secundario de Access Manager (consulte [“Sun Java System Access Manager 6 2005Q1” en la página 82](#)).

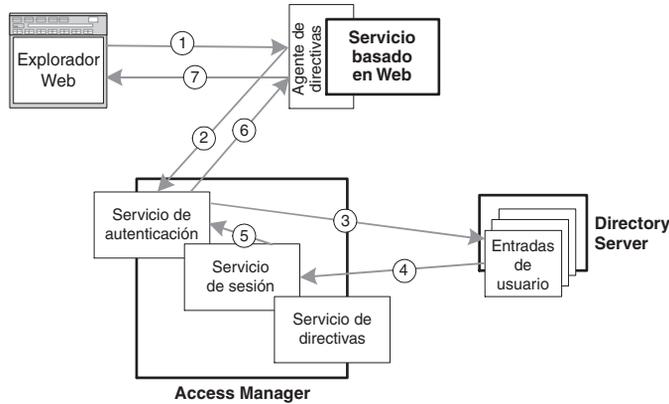
Autenticación

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 aparece en la [Figura 3-2](#). Un agente de políticas 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 verifica la solicitud de autenticación de un usuario, se activa el servicio de sesiones de Access Manager (4), tal y como se muestra en la [Figura 3-2](#). El servicio de sesiones genera un testigo de sesión, que contiene la información de 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.

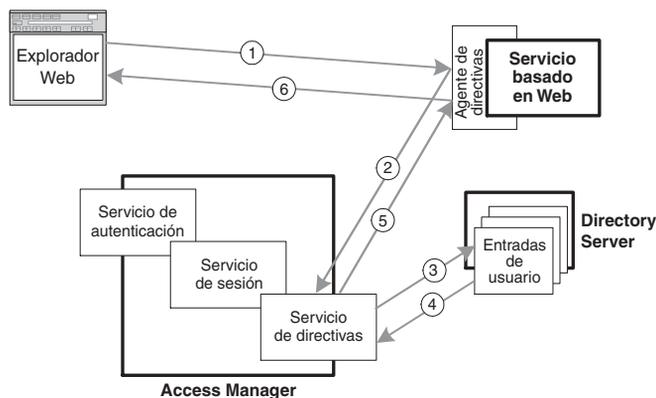
Cuando el usuario autenticado intenta acceder a otro servicio asegurado, el explorador pasa el testigo de sesión al agente de políticas correspondiente. El agente de políticas comprueba con el servicio de sesión que la autenticación anterior del usuario aún es válida y, así, se permite al usuario el acceso 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 políticas que proporciona control de acceso a recursos basados en Web en un entorno de Java ES. Una *política* es una regla que describe quién está autorizado a acceder a un recurso específico y en qué condiciones concretas. 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 de recursos asegurados en Access Manager (1), el agente de políticas lo notifica al servicio de políticas (2), el cual utiliza la información de Directory Server (3) para evaluar la política 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 políticas en una empresa. Las políticas se almacenan en Directory Server y se configuran mediante sus atributos relativos en entradas de organización. También pueden definirse roles para usuarios e incorporarse a definiciones de políticas.

Los agentes de políticas de Access Manager son los agentes que aplican políticas. 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.

directorio Un tipo especial de base de datos optimizada para leer datos en lugar de escribirlos. La mayoría de los directorios se basan en LDAP (Protocolo ligero de acceso a directorios), que es un protocolo estándar de la industria.

política Regla que describe quién está autorizado a obtener acceso a recursos específicos en condiciones concretas. La regla puede basarse en grupos de usuarios o funciones de una organización.

identidad única La identidad que adquiere un usuario mediante 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.

inicio de sesión único Función que permite que la autenticación de un usuario en un servicio de un sistema distribuido se aplique automáticamente a otros servicios del sistema.

Términos clave de este capítulo

Tareas del ciclo de vida de las soluciones de Java Enterprise System

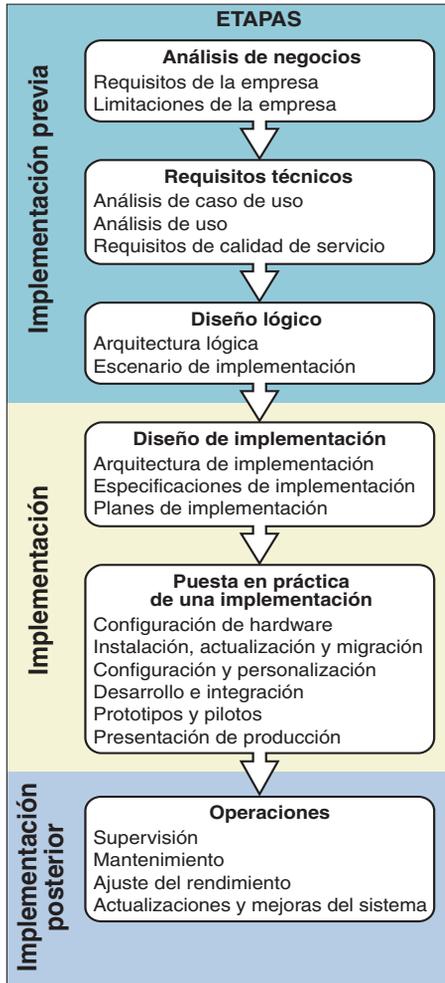
El ciclo de vida de las soluciones de Java ES se presentó en el [Capítulo 1, “Introducción a Java Enterprise System”](#), como un planteamiento estándar para implementar soluciones empresariales utilizando el software de Java ES. Este capítulo describe las tareas involucradas en cada etapa del ciclo de vida. El diagrama del ciclo de vida se repite en la [Figura 4-1 en la página 70](#) para una mejor referencia.

Este capítulo plantea terminología y conceptos relativos a cada etapa. 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 71](#)
- [“Implementación” en la página 72](#)
- [“Implementación posterior” en la página 78](#)
- [“Términos clave de este capítulo” en la página 79](#)

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



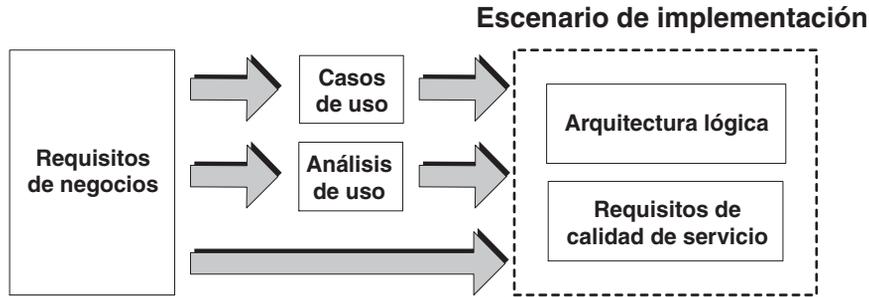
Implementación previa

En las etapas de *implementación previa* del ciclo de vida, se convierte un análisis de necesidades de negocios en un *escenario de implementación*. El escenario de implementación sirve como especificación para el diseño de la implementación.

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

- **Análisis de negocios.** En esta fase, se definen los objetivos empresariales de una implementación propuesta y se especifican las limitaciones y los requisitos empresariales que hay que cumplir para conseguir ese objetivo.
- **Requisitos técnicos.** En esta etapa, se utilizan los resultados del análisis empresarial para crear un *caso de uso* que modele la interacción del usuario con un sistema de software anticipado. También puede determinar patrones de utilización para esos casos de uso. Con el análisis empresarial y de uso, se formulan los requisitos de calidad de servicio (consulte la [Tabla 2-2 en la página 48](#)) que debe satisfacer 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. Utilizando los conceptos planteados en el [Capítulo 2, “Arquitecturas de soluciones de Java Enterprise System”](#), puede 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 engloba en el escenario de implementación, tal y como se muestra en la siguiente figura. Si desea obtener más información acerca de las etapas de implementación previa del ciclo de vida, consulte *Guía de planificación de implementación de Java Enterprise System* .

Figura 4-2 Especificación de un escenario de implementación

Implementación

En las etapas de *implementación* del ciclo de vida, puede convertir un escenario de implementación en un diseño de despliegue que, a continuación, implementará, probará y desplegará 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, tiene que implementar los componentes de aplicación personalizados (componentes J2EE, servicios Web u otros servicios) y los componentes de Java ES necesarios para dar respaldo a la solución.

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

- **Diseño de implementación.** El diseño de implementación depende de la arquitectura lógica de la solución y del rendimiento, de 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.
- **Puesta en práctica de una implementación.** 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

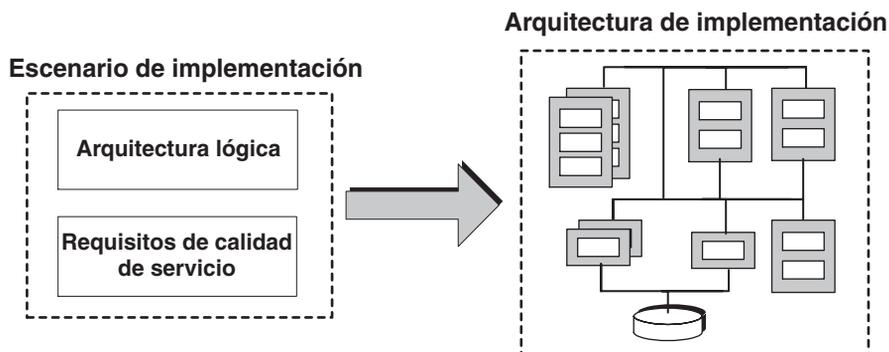
En la fase de diseño de implementación, se crea una arquitectura de implementación de alto nivel seguida de unas especificaciones de implementación de bajo nivel.

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 Java ES y de aplicación 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 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 la implementación, Java Enterprise System está desarrollando distintos modelos 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 de diseño de directorio necesaria para prestar servicio a 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
- Presentación de producción

Puesta en práctica de una implementación

Para llevar a la práctica una implementación, hay que realizar las tareas indicadas 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. Si desea obtener documentación detallada sobre estas tareas, consulte *Java Enterprise System Documentation Roadmap* para obtener más información.

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. Utilice el programa de instalación de Java ES integrado para instalar los componentes de Java ES correspondientes en cada equipo de la arquitectura de implementación (consulte [“Programa de instalación integrado de Java Enterprise System”](#) en la [página 60](#)).

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. Debe realizar la provisión de usuarios para que éstos puedan acceder a diversos servicios y debe configurar las políticas y controles de autenticación y de autorización (consulte [“Identidad integrada y servicios de seguridad” en la página 62](#)).

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 necesita. Por ejemplo, normalmente se personaliza Portal Server para que ofrezca canales de portal, Access Manager para que realice tareas de autorización y Messaging Server para que utilice filtros de comprobación de virus y antispam.

Desarrollo e integración

La arquitectura lógica especificada en el escenario de implementación determina normalmente el ámbito del trabajo de *desarrollo* personalizado necesario para implementar una solución.

Para otras implementaciones, el proceso de desarrollo puede ser bastante amplio, ya que debe 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 etapa de *implementación posterior* del ciclo de vida, se ejecuta una solución que se ha llevado a la práctica en el 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. Estas tareas incluyen la adición de nuevos componentes de Java ES a un sistema para agregar nuevas funciones o sustituir los componentes que no son 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.

implementación Etapa del proceso del ciclo de vida de la Java Enterprise System en la que un escenario de implementación se convierte en un diseño implementado, con prototipo y extraído en un entorno de producción. El producto final de este proceso también se denomina implementación (o solución implementada).

escenario de implementación Una *arquitectura lógica* para una solución de Java Enterprise System y los requisitos de calidad de servicio que debe reunir dicha solución para satisfacer las necesidades del negocio. Entre los requisitos de calidad de servicio, se incluyen los relacionados con: el rendimiento, la disponibilidad, la seguridad, la facilidad de mantenimiento y las funciones de escalabilidad o latencia. Un escenario de implementación es el punto inicial del diseño de implementación.

desarrollo Una tarea del proceso de implementación de la solución de Java Enterprise System en la que se programan y prueban los componentes personalizados de una *arquitectura de implementación*.

implementación previa Etapa del proceso del ciclo de vida de una solución de Java Enterprise System en la que las necesidades de negocios se trasladan a un *escenario de implementación*: una *arquitectura lógica* y un conjunto de requisitos de calidad de servicio que debe cumplir la solución.

implementación posterior Etapa del proceso del ciclo de vida de la solución de Java Enterprise System en la que las aplicaciones distribuidas se inician, se supervisan, se ajustan para optimizar el rendimiento y se actualizan dinámicamente para incluir nuevas funcionalidades.

arquitectura de implementación de referencia Una *arquitectura de implementación* para la que se ha diseñado, implementado y probado su 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 de usuario final específica o un conjunto de tareas que realiza una *aplicación de empresa distribuida* y se utiliza como base para diseñar, probar y medir el rendimiento de la aplicación.

Términos clave de este capítulo

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.** 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 describen en “¿Por qué necesita Java Enterprise System?” en la página 18, incluyen servicios de portal, servicios de comunicación y colaboración, servicios de identidad y seguridad, servicios de aplicaciones y servicios Web y servicios de disponibilidad.
- **Descripciones de componentes de calidad del servicio.** 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.** 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 básica de la documentación de los distintos componentes, consulte *Java Enterprise System Documentation Roadmap* (<http://docs.sun.com/doc/819-0055>).

Descripciones de los componentes de servicios de sistemas

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

- [Sun Java System Access Manager 6 2005Q1](#)
- [Sun Java System Application Server Enterprise Edition 8 2005Q1](#)
- [Sun Java System Calendar Server 6 2005Q1](#)
- [Sun Java System Directory Server 5 2005Q1](#)
- [Sun Java System Instant Messaging 7 2005Q1](#)
- [Sun Java System Message Queue 3 2005Q1](#)
- [Sun Java System Messaging Server 6 2005Q1](#)
- [Sun Java System Portal Server 6 2005Q1](#)
- [Sun Java System Web Server 6.1 2005Q1](#)

Sun Java System Access Manager 6 2005Q1

Sun Java System Access Manager (Access Manager) proporciona una infraestructura que hace posible que las organizaciones administren los procesos usados para gestionar las identidades digitales de los clientes, los empleados y los socios que usan los servicios basados en Web y las 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 incluye Access Manager como un componente instalable independiente. Si es necesario, los siguientes subcomponentes de Access Manager pueden instalarse por separado:

- **Identity Management y Policy Services Core.** Incluye los medios para crear y administrar identidades de usuario y para definir y evaluar directivas que permitan el acceso a recursos de Java ES en función de las identidades de los usuarios. Este subcomponente también incluye los subcomponentes SDK y Delegated Administrator de Access Manager (consulte [“Sun Java System Delegated Administrator 6 2005Q1”](#) en la página 91).

- **Access Manager SDK.** Proporciona una interfaz remota para Access Manager. Este subcomponente tiene que instalarse en un equipo que cuente con 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 gestión de directivas. También proporciona una interfaz única para usuarios con objeto de crear y gestionar cuentas de usuario, atributos de servicio y reglas de acceso en Directory Server.
- **Servicios de dominio común para administración de federación.** 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 2005Q1

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

El programa de instalación de Java ES incluye Application Server como un componente instalable independiente. 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 y despliegue de componentes y aplicaciones J2EE.
- **Application Server Administration Client.** Proporciona clientes de administración gráficos que permiten gestionar y configurar las instalaciones de Application Server y las aplicaciones hospedadas. Administration Client también ayuda a implementar aplicaciones.
- **Herramienta de administración de línea de comandos.** Proporciona clientes de administración de la línea de comandos que permiten gestionar y configurar las instalaciones de Application Server y las aplicaciones hospedadas. La herramienta también ayuda a implementar aplicaciones.
- **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 2005Q1

Sun Java System Calendar Server (Calendar Server) es una solución escalable, basada en Web de servicios de agenda centralizados para empresas y proveedores de servicios. Calendar Server es compatible con los calendarios personales y de grupo, y también con los calendarios para recursos como salas y equipo de conferencia.

El programa de instalación de Java ES incluye Calendar Server como un componente instalable independiente.

Sun Java System Directory Server 5 2005Q1

Sun Java System Directory Server (Directory Server) incluye un servicio de directorio centralizado para la información de intranet, de red y de extranet. Directory Server se integra con sistemas existentes y actúa como depósito centralizado para la consolidación de información de empleados, clientes, proveedores y 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 incluye Directory Server como un componente instalable independiente.

Sun Java System Instant Messaging 7 2005Q1

Sun Java System Instant Messaging (Instant Messaging) permite a los usuarios finales participar en la mensajería instantánea y en sesiones de chat, enviar mensajes de alerta y compartir noticias de grupo al instante. Instant Messaging es adecuado para intranets y para Internet, y admite la interacción con otros proveedores de mensajería instantánea.

El programa de instalación de Java ES incluye Instant Messaging como un componente instalable independiente. Los siguientes subcomponentes de Instant Messaging pueden instalarse por separado:

- **Instant Messaging Server Core.** Incluye software de servidor y de multiplexor.
- **Instant Messaging Recursos.**
- **Access Manager Instant Messaging Service.**

Sun Java System Message Queue 3 2005Q1

Sun Java System Message Queue (Message Queue) es una solución basada en estándares para el problema de comunicación entre aplicaciones y la entrega fiable de mensajes. Message Queue es un sistema de mensajería de empresa que implementa el estándar abierto de Java Message Service (JMS).

Además de ser proveedor de JMS, Message Queue cuenta con funciones que superan los requisitos mínimos de la especificación JMS. Con el software de Message Queue, los procesos que se ejecuten en plataformas y en sistemas operativos diferentes se pueden conectar a un servicio de Message Queue común para enviar y recibir información. Los desarrolladores de aplicaciones se pueden centrar en la lógica de 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** (predeterminado). 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 QueuePlatform Edition como componentes para instalarlos por separado.

Sun Java System Messaging Server 6 2005Q1

Sun Java System Messaging Server (Messaging Server) es un servidor de mensajería de Internet basado en estándares para empresas y proveedores de servicios. Diseñado para gestionar mensajes fiables de gran capacidad, Messaging Server está formado por varios componentes modulares, que se pueden configurar de forma independiente y son compatibles con varios protocolos de correo electrónico.

El programa de instalación de Java ES incluye Messaging Server como un componente instalable independiente. 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:

- **Almacén de mensajes.** Permite almacenar y recuperar mensajes.
- **Message Transfer Agent (MTA).** Permite el envío de correos electrónicos mediante el control de las conexiones SMTP, encaminamiento de correos electrónicos y entrega de mensajes a los almacenes de mensajes correctos. 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, utilizando protocolos IMAP o POP.
- **Message Express Multiplexor (MEM).** Permite recuperar y enviar correos electrónicos mediante clientes de correo electrónico basados en Web (HTML).

Sun Java System Portal Server 6 2005Q1

Sun Java System Portal Server (Portal Server) es una solución de servidor de portales con identidades. Portal Server combina servicios de portales clave, como 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 y asistentes digitales personales.

El programa de instalación de Java ES proporciona Portal Server, incluida la función de acceso móvil, como un componente instalable independiente.

Sun Java System Web Server 6.1 2005Q1

Sun Java System Web Server (Web Server) es un servidor Web seguro, multiproceso y multiproceso, creado con estándares abiertos. Web Server ofrece un alto rendimiento, fiabilidad, escalabilidad y una alta capacidad de gestión para empresas de cualquier tamaño. Web Server es compatible con una amplia gama de estándares de software de Web, como JDK 1.4.1, Java Servlet 2.3, JavaServer Pages™ (JSP™) 1.2, HTTP/1.1, PKCS #11, FIPS-140, certificados escalables de 168 bits y otros estándares basados en la seguridad.

El programa de instalación de Java ES incluye Web Server como un componente instalable independiente.

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 9/04 y Sun Cluster Agents](#)
- [Almacén de sesión de alta disponibilidad 2005Q1](#)

Sun Cluster 3.1 9/04 y Sun Cluster Agents

El software de Sun Cluster incluye servicios de alta disponibilidad y escalabilidad para Java Enterprise System y para aplicaciones basadas en la infraestructura de Java ES.

Un clúster es un conjunto de equipos acoplados libremente (nodos de clúster) que ofrecen de forma conjunta una única visión de los servicios, recursos de sistemas y datos de cliente. Internamente, el clúster utiliza equipos redundantes, interconexiones, almacenamiento de datos e interfaces de red para ofrecer una alta disponibilidad a los servicios y datos basados en el clúster. El software Sun Cluster supervisa continuamente la salud de los nodos miembros y otros recursos del clúster, y utiliza la redundancia interna para proporcionar acceso prácticamente continuo a estos recursos, incluso cuando se produce un fallo.

El programa de instalación de Java ES proporciona Sun Cluster Core y Sun Cluster Agents como componentes instalable independiente. Se encuentran disponibles agentes de Sun Cluster adicionales en varios CD.

Almacén de sesión de alta disponibilidad 2005Q1

El almacén de sesión de alta disponibilidad (HADB) de Sun Java System proporciona un almacén de datos que se puede utilizar para que los datos de las aplicaciones estén disponibles, incluso en caso de 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 incluye HADB como un componente instalable independiente. 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 2005Q1](#)
- [Sun Java System Connector para Microsoft Outlook 6 2005Q1](#)
- [Sun Java System Directory Proxy Server 5 2005Q1](#)
- [Sun Java System Portal Server Secure Remote Access 6 2005Q1](#)

Sun Java System Communications Express 2005Q1

Sun Java System Communications Express (Communications Express) proporciona comunicaciones integradas basadas en Web y un cliente de colaboración que proporciona a los usuarios finales una interfaz Web a los servicios de calendario, correo electrónico y libreta de direcciones. Communications Express consta de tres módulos cliente: libreta de direcciones, calendario y correo. Se puede configurar para proporcionar acceso a Messaging Server o Calendar Server (o ambos). Communications Express funciona con Sun Java System LDAP Schema, Versión 1 (Schema 1) o Schema 2.

El programa de instalación de Java ES incluye Communications Express como un componente instalable independiente.

Sun Java System Connector para Microsoft Outlook 6 2005Q1

Sun Java System Connector para Microsoft Outlook permite utilizar Outlook 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, Connector usa WCAP para efectuar consultas a Calendar Server sobre acontecimientos y tareas que se convierten a continuación 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: correo de Messaging Server e información de calendario 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 2005Q1

Sun Java System Directory Proxy Server (Directory Proxy Server) es un componente primordial de cualquier servicio de directorio vital en soluciones de comercio electrónico. Directory Proxy Server es una puerta de enlace de protocolo de capa de aplicación de LDAP que ofrece mejoras de control de acceso al directorio, compatibilidad de esquemas y alta disponibilidad utilizando un balance de carga de capa de aplicación y conmutación por error.

El programa de instalación de Java ES incluye Directory Proxy Server como un componente instalable independiente.

Sun Java System Portal Server Secure Remote Access 6 2005Q1

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) amplía Portal Server ofreciendo acceso remoto seguro basado en explorador al contenido y los servicios de Portal Server desde cualquier explorador remoto, por lo que se elimina la necesidad de software cliente. La integración con Portal Server garantiza que los usuarios reciben un acceso seguro al contenido y a los servicios a los que tienen permiso.

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

- **Portal Server Secure Remote Access Core.**
- **Gateway.** 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.
- **Netlet Proxy.** 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.
- **Rewriter Proxy.** 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 \(y Console\) 5 2005Q1](#)
- [Herramienta de preparación de Sun Java System Directory 2005Q1](#)
- [Sun Java System Delegated Administrator 6 2005Q1](#)
- [Sun Remote Services Net Connect 2005Q1](#)

Sun Java System Administration Server (y Console) 5 2005Q1

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

Server Console es una aplicación Java independiente que trabaja en la 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 incluye Server Console y Administration Server como un componente único para instalarlo.

Herramienta de preparación de Sun Java System Directory 2005Q1

La secuencia de preparación de Sun Java System Directory se utiliza para configurar Directory Server con el esquema necesario para aportar los datos para los usuarios de Messaging Server y Calendar Server.

Sun Java System Delegated Administrator 6 2005Q1

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

Delegated Administrator se instala automáticamente cuando elige instalar Access Manager.

Sun Remote Services Net Connect 2005Q1

Sun Remote Services Net Connect es una recopilación de servicios de administración del sistema diseñados para ayudarle a controlar mejor su entorno de TI. Estos servicios mediante Web le dan la oportunidad de supervisar los sistemas de forma automática, crear informes de rendimiento y tendencias y recibir notificación automática de acontecimientos del sistema para poder actuar más rápidamente y gestionar problemas potenciales antes de que se agraven.

El programa de instalación de Java ES incluye Sun Remote Services Net Connect como un componente instalable independiente.

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 programa de instalación de Java ES instala automáticamente los componentes compartidos necesarios para la compatibilidad de otros componentes de Java ES instalados en un equipo host.

Java Enterprise System incluye los componentes compartidos que se enumeran a continuación:

- Ant (herramienta basada en Jakarta ANT Java/XML)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley DB
- Contenedor de agentes común
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXM (Java API for XML Messaging) Client Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) 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

- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP, incluido NSPERL
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (sólo Solaris)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (`libxml`)
- WSCL (Web services Common Library)

Componentes compartidos

Índice

A

- Access Manager
 - como componente de los servicios del sistema [22](#)
 - como servicio de infraestructura [41](#)
 - descripción de [82](#)
- actualizaciones, *Consulte* escenarios de adopción
- Administration Server
 - como componente de calidad del servicio [25](#)
 - descripción de [90](#)
- Almacén de sesión de alta disponibilidad
 - descripción de [88](#)
- almacén de sesión de alta disponibilidad
 - como componente de calidad del servicio [24](#)
- ampliaciones, *Consulte* escenarios de adopción
- Ant (herramienta basada en Jakarta ANT Java/XML) [92](#)
- Apache SOAP Runtime [92](#)
- aplicaciones
 - distribuidas, *Consulte* aplicaciones de empresa distribuidas
 - empresa, *Consulte* aplicaciones de empresa distribuidas
- aplicaciones de empresa distribuidas
 - acerca de [18](#)
 - definido [32](#)
 - infraestructura para [18](#)
- Application Server
 - como componente de los servicios del sistema [22](#)
 - como servicio de infraestructura [41](#)
 - descripción de [83](#)
- arquitectura
 - definido [57](#)
 - dimensiones de, *Consulte* dimensiones arquitectónicas
 - implementación [73](#)
 - lógica [71](#)
 - presentación [35](#)
 - referencia [74](#)
 - solución [36](#)
- arquitecturas de implementación
 - definido [57](#)
 - diseño de [73](#)
 - presentación [35](#)
 - relación con arquitecturas de capas [46](#)
- arquitecturas de implementación de referencia
 - definido [79](#)
 - presentación [74](#)
- arquitecturas lógicas
 - acerca de [71](#)
 - definido [57](#)
 - ejemplo [54](#)
 - niveles de servicios de infraestructura, y [38](#)
 - presentación [35](#)
- autenticación [64](#)
- autorización [66](#)

B

Berkeley DB 92

C

Calendar Server

- como componente de los servicios del sistema 22
- como servicio de infraestructura 41
- descripción 84

capas, lógicas

- arquitectura de aplicación, y 43
- capa lógica de cliente 44
- cliente 44
- datos 45
- presentación 45
- servicio de negocios 45

CAPI (Java Calendar API) 92

casos de uso

- definido 79
- presentación 71

categorías de usuarios

- administrador de sistemas 29
- administrador de sistemas especializado 29
- administrador delegado 29
- analista de sistemas 29
- arquitecto 29
- director de TI 29
- ingeniero de campo 29
- integrador de sistemas 29
- organizador de negocios 29

clientes

- componentes de los servicios del sistema, y 22
- definido 57

clústeres, *Consulte* Sun Cluster

Communications Express

- como componente de calidad del servicio 25
- descripción de 88

componentes 18

- acceso 24
- administrativo 25
- calidad de servicio 23, 87
- compartidos 26, 92
- definido 32

dependencias 42

detección de versiones instaladas de 61

disponibilidad 24

distribuidos 18

EJB 45

J2EE 45

JSP 45

servicio de sistemas 22, 82

Servlet 45

sistema, *Consulte* componentes del sistema y servicios de infraestructura 41

componentes administrativos

- descripciones de 90
- presentación 25

componentes compartidos 92

- definido 33
- presentación 26

componentes de acceso

- descripciones de 88
- presentación 24

componentes de aplicación

- definido 56
- en arquitecturas de capas lógicas 44

componentes de calidad del servicio

- definido 32
- descripciones de 87
- presentación 23

componentes de disponibilidad

- descripciones de 87
- presentación 24

componentes de EJB 45

componentes de Java Servlet 45

componentes de JSP 45

componentes de los servicios del sistema

- definido 33
- dependencias 42
- presentación 22

componentes del sistema

- acerca de 21
- componentes compartidos 26, 92
- componentes de calidad del servicio 23
- componentes de los servicios del sistema, *Consulte* componentes de los servicios del sistema definido 33
- servicio de sistemas 82

comprobación de relaciones de dependencia,
 programa de instalación 61

Connector para Microsoft Outlook
 como componente de calidad del servicio 25
 descripción de 89

contenedor de agentes común 92

creación de prototipos 76

D

Delegated Administrator
 como componente de calidad del servicio 25
 descripción de 91

dependencias 42, 61

desarrollo
 como tarea de implementación 76
 definido 79

detección del software instalado 61

dimensiones arquitectónicas
 calidad de servicio 48
 capas lógicas 43
 dependencias de servicio de infraestructura 38
 síntesis 52

directorios
 como almacenes de datos de usuario 63
 definido 67

Directory Proxy Server
 como componente de calidad del servicio 24
 como componente del sistema 50
 descripción de 89

Directory Server
 como componente de los servicios del sistema 22
 como servicio de infraestructura 41
 descripción de 84

disponibilidad
 requisitos 48, 50
 servicios 50, 87

distribuidas
 aplicaciones, *Consulte* aplicaciones de empresa
 distribuidas

distribuidos
 servicios, *Consulte* servicios distribuidos

documentación, Java ES 12

E

entrada de usuario 63

escalabilidad
 requisitos 48, 50
 servicios 50, 87

escenarios de adopción, Java ES
 acerca de 29
 actualización 30
 ampliación 30
 definido 32
 mejora 30
 nuevo sistema 29

escenarios de implementación
 definido 79
 presentación 71

especificaciones de implementación 74, 75

etapas del ciclo de vida
 implementación 27, 72
 implementación posterior 27, 78
 implementación previa 27, 71

F

formación, escenarios de adopción de Java ES, y 31

funciones de integración
 componentes compartidos 21
 identidad y seguridad 21, 62
 presentación 21
 programa de instalación integrado 21, 60

H

hardware, escenarios de adopción de Java ES, y 31

Herramienta de preparación de Directory
 como componente de calidad del servicio 25
 descripción de 91

I

ICU (International Components for Unicode) 92

identidad

administración 62

servicios 19, 62

usuario único 63

identidad única

definido 67

presentación 63

implementación

arquitectura 73

arquitecturas de referencia 74

definido 79

desarrollo y personalización 76

diseño 73

escenarios, *Consulte* escenarios de
implementación

etapa del ciclo de vida 72

fases del ciclo de vida 72

implementación 75

presentación de producción 78

prueba de prototipo 76

implementación posterior

definido 79

fases del ciclo de vida 78

implementación previa

definido 79

fases del ciclo de vida 71

infraestructura

dependencias de servicio, *Consulte* servicios
distribuidos

para aplicaciones de empresa distribuidas 18

inicio de sesión único

definido 67

función de Java ES 22, 64

implementación de 65

niveles de servicios de infraestructura, y 39

Instant Messaging

como componente de los servicios del sistema 23

como servicio de infraestructura 41

descripción de 84

integración

escenarios de adopción de Java ES, y 31

funciones, *Consulte* funciones de integración

servicios 40

J

J2EE

componentes 45

modelo de componentes distribuidos 45

plataforma 23

J2SE (Java 2 Platform, Standard Edition) 92

J2SE™ (Java 2 Platform, Standard Edition) 26

JAF (JavaBeans™ Activation Framework) 92

JATO (Java Studio Enterprise Web Application
Framework) 92

JavaHelp 92

JavaMail 92

JAXB (Java Architecture for XML Binding) 92

JAXM (Java API for XML Messaging) 92

JAXP (Java API for XML Processing) 92

JAXR (Java API for XML Registries) 92

JAX-RPC 92

JDMK (Java Dynamic Management™ Kit) 92

JMS (Java Message Service) 23

JSS (Java Security Services) 26, 92

K

KT Search Engine (KTSE) 92

L

LDAP 45, 67

LDAP C SDK 92

LDAP Java SDK 92

Linux 41

M

- mejoras, *Consulte* escenarios de adopción
- Message Queue
 - como componente de los servicios del sistema 23
 - como servicio de infraestructura 41
 - descripción de 85
- Messaging Server
 - como componente de los servicios del sistema 23
 - como servicio de infraestructura 41
 - descripción de 85
- migración, escenarios de adopción de Java ES, y 30

N

- NSPR (Netscape Portable Runtime) 26, 93
- NSS (Network Security Services) 26, 93

P

- perfiles de usuarios 29
- Perl LDAP 93
- plataforma J2ME 44
- políticas
 - autorización 66
 - definido 67
- Portal Server
 - como componente de los servicios del sistema 23
 - como servicio de infraestructura 41
 - descripción de 86
- Portal Server Secure Remote Access
 - como componente de calidad del servicio 24
 - como componente del sistema 50
 - descripción de 89
- presentación de producción 78
- productos de Sun Java System
 - Access Manager, *Consulte* Access Manager
 - Administration Server, *Consulte* Administration Server
 - almacén de sesión de alta disponibilidad, *Consulte* almacén de sesión de alta disponibilidad

- Application Server, *Consulte* Application Server
- Calendar Server, *Consulte* Calendar Server
- Communications Express, *Consulte* Communications Express
- Connector para Microsoft Outlook, *Consulte* Connector para Microsoft Outlook
- Delegated Administrator, *Consulte* Delegated Administrator
- Directory Proxy Server, *Consulte* Directory Proxy Server
- Directory Server, *Consulte* Directory Server
- Herramienta de preparación de Directory, *Consulte* Herramienta de preparación de Directory
- Instant Messaging *Consulte* Instant Messaging
- Message Queue, *Consulte* Message Queue
- Messaging Server, *Consulte* Messaging Server
- Portal Server, *Consulte* Portal Server
- Portal Server, Secure Remote Access, *Consulte* Portal Server, Secure Remote Access
- Web Server, *Consulte* Web Server
- programa de desinstalación 62
- provisión de usuarios 74

R

- Remote Services Net Connect
 - como componente de calidad del servicio 25
 - descripción de 91
- requisitos de calidad de servicio
 - disponibilidad 48, 50
 - escalabilidad 48, 50
 - facilidad de mantenimiento 49, 50
 - función latente 49
 - rendimiento 48
 - seguridad 48, 50
- requisitos de facilidad de mantenimiento 49, 50
- requisitos de la función latente 49
- requisitos de rendimiento 48

S

- SAAJ (SOAP with Attachments API for Java) 93
- SAML (Security Assertions Markup Language) 93
- SASL (Simple Authentication and Security Layer) 93
- seguridad
 - requisitos 48, 50
 - servicios 19
 - servicios de directivas 39
- servicios
 - alta disponibilidad 50, 87
 - definido 32
 - escalabilidad 50, 87
 - infraestructura 18
 - infraestructura, *Consulte* servicios de infraestructura distribuidos
 - negocios 45
 - Web 45
- servicios administrativos 20
- servicios de acceso 20
- servicios de aplicación 19, 38
- servicios de colaboración 19
- servicios de colaboración de usuarios 40
- servicios de comunicación 19
- servicios de disponibilidad 20
- servicios de mensajería 39
- servicios de negocios
 - capa de presentación, y 45
 - definido 57
- servicios de persistencia 39
- servicios de plataforma 38
- servicios de portal 19
- servicios de sistema operativo 39
- servicios de sistemas
 - acerca de 22
 - definido 33
- servicios de software intermedio 38
- servicios de tiempo de ejecución 39
- servicios de transporte de red 39
- servicios distribuidos
 - acceso 20
 - administrativo 20
 - colaboración de usuarios 40
 - comunicación y colaboración 19
 - disponibilidad 20
 - identidad 19
 - infraestructura 18
 - integración 40
 - mensajería 39
 - nivel de aplicación 38
 - persistencia 39
 - plataforma 38, 39
 - portal 19
 - seguridad 19, 39
 - software intermedio 38
 - tiempo de ejecución 19, 39
 - transporte de red 39
 - visión general 18
 - Web 19
- servicios Web 19
 - componentes J2EE, y 45
 - definido 57
- servidores
 - definido 57
 - independientes 45
- sistema
 - componentes, *Consulte* componentes del sistema
 - configuración 62
 - servicios 18
- SNMP (Simple Network Management Protocol) 93
- SOAP (Apache) 92
- Solaris 41
- soluciones, Java ES
 - arquitectura 35
 - ciclo de vida 27
 - ejemplo 53
 - personalizada y estándar 53
- soporte de idioma 60
- Sun Cluster
 - agentes 51
 - como componente de calidad del servicio 24
 - como servicio de disponibilidad 50
 - descripción de 87
- Sun Explorer Data Collector 93
- Sun Java Monitoring Framework 93
- Sun Java Web Console 93

T

tareas, Java Enterprise System [27, 69](#)
Tomcat Servlet JSP Container [93](#)

U

usuarios finales
 aplicaciones distribuidas, y [18](#)
 definido [32](#)
usuarios, *Consulte* usuarios finales
usuarios, provisión [74](#)

W

Web Server
 como componente de los servicios del sistema [23](#)
 como servicio de infraestructura [41](#)
 descripción de [86](#)
WSCL (Web services Common Library) [93](#)

X

XML C Library [93](#)

Apartado X