



Sun Java™ System

# Sun Java Enterprise System 2005Q1

## Guía de planificación de la implementación

---

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

Nº de pieza: 819-1917

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

<b>Lista de figuras</b> .....	<b>7</b>
<b>Lista de tablas</b> .....	<b>9</b>
<b>Prefacio</b> .....	<b>11</b>
Usuarios a los que está destinada esta guía .....	12
Organización de esta guía .....	12
Documentación relacionada .....	12
Guías de este conjunto de documentación .....	12
Acceso a los recursos de Sun en línea .....	14
Contacto con el servicio técnico de Sun .....	14
Referencias a sitios Web de terceros relacionados .....	15
Sun valora sus comentarios .....	15
<b>Capítulo 1 Introducción a la planificación de la implementación</b> .....	<b>17</b>
Acerca de Java Enterprise System .....	17
Servicios del sistema .....	18
Servicios incorporados y servicios personalizados .....	19
Migración a Java Enterprise System .....	20
Información acerca de la planificación de implementación .....	21
Ciclo de vida de la solución .....	22
Etapa de análisis de negocios .....	23
Etapa de requisitos técnicos .....	23
Etapa de diseño lógico .....	24
Etapa de diseño de implementación .....	25
Etapa de implementación .....	25
Etapa de operaciones .....	26

<b>Capítulo 2 Análisis de negocios</b> .....	<b>27</b>
Acerca del análisis de negocios .....	27
Definición de requisitos empresariales .....	28
Establecimiento de objetivos empresariales .....	28
Alcance .....	28
Prioridades .....	28
Calidades críticas .....	29
Factores de crecimiento .....	29
Margen de seguridad .....	29
Comprensión de las necesidades del usuario .....	29
Desarrollo de requisitos operativos .....	30
Compatibilidad de los patrones de uso existentes .....	31
Comprensión de la cultura corporativa .....	31
Participantes .....	31
Normas y políticas .....	31
Requisitos obligatorios .....	32
Seguridad .....	32
Distribución del sitio .....	32
Utilización de un planteamiento progresivo .....	33
Acuerdos de nivel de servicio .....	33
Definición de limitaciones empresariales .....	34
Aspectos relacionados con la migración .....	34
Obligaciones de programación .....	34
Limitaciones presupuestarias .....	35
Coste de propiedad .....	35
<b>Capítulo 3 Requisitos técnicos</b> .....	<b>37</b>
Información acerca de los requisitos técnicos .....	37
Análisis de uso .....	38
Casos de uso .....	40
Requisitos de calidad de servicio .....	41
Rendimiento .....	42
Disponibilidad .....	43
Sistemas tolerantes a fallos .....	44
Establecimiento de las prioridades en cuanto a disponibilidad del servicio .....	44
Pérdida de servicios .....	45
Escalabilidad .....	45
Cálculo del crecimiento previsto .....	46
Requisitos de seguridad .....	47
Elementos de un plan de seguridad .....	48
Capacidad latente .....	49
Requisitos de mantenimiento .....	49
Requisitos de nivel de servicio .....	50

<b>Capítulo 4 Diseño lógico</b> .....	<b>51</b>
Información acerca de las arquitecturas lógicas .....	51
Diseño de una arquitectura lógica .....	53
Componentes de Java Enterprise System .....	53
Relaciones de dependencia de los componentes .....	54
Compatibilidad con contenedor Web .....	57
Servicios distintos lógicamente prestados por Messaging Server .....	57
Componentes de acceso .....	58
Diseño de arquitectura en varias capas .....	59
Ejemplo de arquitecturas lógicas .....	60
Ejemplo de Messaging Server: .....	61
Casos de uso de Messaging Server .....	62
Ejemplo de comunicaciones basadas en identidades .....	65
Ejemplo de casos de uso para comunicaciones basadas en identidades .....	66
Zonas de acceso .....	69
Escenario de implementación .....	70
<b>Capítulo 5 Diseño de implementación</b> .....	<b>71</b>
Información acerca del diseño de implementación .....	72
Aprobación del proyecto .....	72
Resultados del diseño de implementación .....	73
Factores que afectan al diseño de implementación .....	75
Metodología del diseño de implementación .....	76
Cálculo de los requisitos de procesadores .....	77
Ejemplo de cálculo de los requisitos de procesadores .....	79
Determine el cálculo base de CPU para los puntos de entrada del usuario .....	79
Incluya los cálculos de CPU para las relaciones de dependencia de los servicios .....	81
Estudie los casos de uso para la utilización de carga máxima .....	82
Modifique los cálculos para otros estados de carga .....	83
Actualice los cálculos de CPU .....	83
Cálculo de los requisitos de procesadores para transacciones seguras .....	84
Cálculos de CPU para transacciones seguras .....	85
Hardware especializado para tratar las transacciones SSL .....	87
Determinación de las estrategias de disponibilidad .....	88
Estrategias de disponibilidad .....	88
Sistema de un servidor .....	89
Sistemas redundantes horizontalmente .....	90
Software de Sun Cluster .....	92
Ejemplo de diseño de disponibilidad .....	92
Ejemplo de equilibrado de carga para Messaging Server .....	92
Ejemplo de recuperación tras error utilizando el software de Sun Cluster .....	94
Ejemplo de duplicación de servicios de directorio .....	94
Repetición de una réplica principal .....	95
Repetición de varias réplicas principales .....	96

Determinación de estrategias para la escalabilidad .....	97
Capacidad latente .....	97
Ejemplo de escalabilidad .....	98
Identificación de cuellos de botella de rendimiento .....	99
Optimización de acceso al disco .....	101
Diseño para la óptima utilización de los recursos .....	102
Gestión de riesgos .....	104
Ejemplo de arquitectura de implementación .....	104
<b>Capítulo 6 Implantación de un diseño de implementación .....</b>	<b>107</b>
Información acerca de la implantación de diseños de implementación .....	108
Instalación y configuración del software .....	108
Desarrollo de pilotos y prototipos .....	109
Comprobación de las implementaciones piloto y prototipo .....	110
Despliegue de una implementación de producción .....	111
<b>Índice .....</b>	<b>113</b>

# Lista de figuras

Figura 1-1	Ciclo de vida de la solución	22
Figura 4-1	Tres dimensiones de la arquitectura de la solución Java Enterprise System	52
Figura 4-2	Componentes de Java Enterprise System	54
Figura 4-3	Java Enterprise System Relaciones de dependencia de los componentes	56
Figura 4-4	Modelo de arquitectura en varias capas	59
Figura 4-5	Arquitectura lógica de una implementación de Messaging Server	61
Figura 4-6	Arquitectura lógica de Messaging Server que muestra el caso de uso 1	63
Figura 4-7	Arquitectura lógica de Messaging Server que muestra el caso de uso 2	64
Figura 4-8	Arquitectura lógica de Messaging Server que muestra el caso de uso 3	65
Figura 4-9	Arquitectura lógica para un escenario de comunicaciones basado en identidades	66
Figura 4-10	Arquitectura lógica de un escenario de comunicaciones que muestra el caso de uso 1	67
Figura 4-11	Arquitectura lógica de un escenario de comunicaciones que muestra el caso de uso 2	68
Figura 4-12	Componentes lógicos colocados en zonas de acceso	69
Figura 5-1	Arquitectura lógica para un escenario de comunicaciones basado en identidades	80
Figura 5-2	Sistema de un servidor	89
Figura 5-3	Sistema de recuperación tras error N+1 con dos servidores	90
Figura 5-4	Equilibrado de carga más recuperación tras error en dos servidores	90
Figura 5-5	Distribución de la carga entre $n$ servidores	91
Figura 5-6	Ejemplo de estrategia de disponibilidad para Messaging Server	93
Figura 5-7	Diseño de recuperación tras error utilizando el software de Sun Cluster	94
Figura 5-8	Ejemplo de repetición de una réplica principal	95
Figura 5-9	Ejemplo de repetición de varias réplicas principales	96
Figura 5-10	Ejemplos de escalamiento horizontal y vertical	99
Figura 5-11	Ejemplo de arquitectura de implementación	105



# Lista de tablas

Tabla 1	Java Enterprise System Documentación .....	13
Tabla 1-1	Categorías de servicios de Java Enterprise System .....	19
Tabla 3-1	Factores de análisis de uso .....	38
Tabla 3-2	Calidades de sistema que afectan a los requisitos de QoS .....	41
Tabla 3-3	Tiempo de inactividad no programado para un sistema que está operativo todo el año (8.760 horas) .....	43
Tabla 3-4	Disponibilidad de servicios por prioridad .....	44
Tabla 3-5	Factores de deescalabilidad .....	46
Tabla 3-6	Temas para requisitos de mantenimiento .....	49
Tabla 4-1	Java Enterprise System Relaciones de dependencia de los componentes .....	55
Tabla 4-2	Messaging Server Configuraciones .....	57
Tabla 4-3	Java Enterprise System Componentes que ofrecen acceso remoto .....	58
Tabla 4-4	Capas lógicas en una arquitectura en varias capas .....	59
Tabla 4-5	Componentes en la arquitectura lógica de Messaging Server .....	61
Tabla 4-6	Zonas de acceso seguro y componentes en ellas .....	70
Tabla 5-1	Cálculos de CPU para componentes que contienen acceso a los puntos de entrada de usuario .....	80
Tabla 5-2	Cálculos de CPU para compatibilidad de componentes .....	81
Tabla 5-3	Ajustes en el cálculo de CPU para carga máxima .....	82
Tabla 5-4	Ajustes de cálculos de CPU para compatibilidad de componentes .....	84
Tabla 5-5	modificación de los cálculos de CPU para transacciones seguras .....	86
Tabla 5-6	Ajustes de cálculos de CPU para transacciones seguras en Portal Server .....	87
Tabla 5-7	Ajustes de cálculos de CPU para compatibilidad de componentes .....	92
Tabla 5-8	Puntos de acceso a los datos .....	100
Tabla 5-9	Consideraciones para la gestión de recursos .....	103



# Prefacio

*Guía de planificación de la implementación de Java Enterprise System* proporciona una introducción a la planificación y diseño de soluciones de implementación empresarial basadas en Sun Java™ Enterprise System. Esta guía presenta conceptos y principios básicos de la planificación y diseño de implementación, plantea el ciclo de vida de la solución, que engloba las fases y tareas de un proyecto de diseño de implementación, y proporciona ejemplos y estrategias globales que se pueden emplear a la hora de planificar soluciones de implementación en toda la empresa con Java Enterprise System.

Este prefacio incluye los siguientes apartados:

- “Usuarios a los que está destinada esta guía” en la página 12
- “Documentación relacionada” en la página 12
- “Acceso a los recursos de Sun en línea” en la página 14
- “Contacto con el servicio técnico de Sun” en la página 14
- “Referencias a sitios Web de terceros relacionados” en la página 15
- “Sun valora sus comentarios” en la página 15

Antes de realizar cualquiera de las tareas descritas en esta guía, lea *Notas de la versión de Java Enterprise System* (<http://docs.sun.com/doc/819-0813>).

# Usuarios a los que está destinada esta guía

Esta guía está principalmente pensada para los arquitectos de implementación y planificadores empresariales responsables del análisis y diseño de las implementaciones empresariales. Esta guía también es útil para los integradores de sistemas y otros responsables para el diseño e implementación de distintos aspectos de una aplicación empresarial.

Esta guía asume que está familiarizado con el diseño e instalación de aplicaciones en la empresa y que ha leído *Visión general técnica de Java Enterprise System*.

## Organización de esta guía

Esta guía está basada en un ciclo de vida de la solución que describe las distintas etapas de la planificación de implementación. El [Capítulo 1, “Introducción a la planificación de la implementación”](#), proporciona una descripción del ciclo de vida de la solución.

## Documentación relacionada

El sitio web <http://docs.sun.com><sup>SM</sup> permite acceder en línea a la documentación técnica de Sun. Puede buscar el archivo o buscar un título de guía o asunto específico.

## Guías de este conjunto de documentación

Los manuales de Java ES 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<sup>TM</sup> en:

<http://docs.sun.com>

La documentación de Java ES incluye información acerca del sistema como una unidad completa y también de sus componentes. Se puede acceder a la documentación 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 ES. 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** Java Enterprise System Documentación

Documento	Contenido
<p><i>Notas de la versión de Java Enterprise System</i>  <a href="http://docs.sun.com/doc/819-0813">http://docs.sun.com/doc/819-0813</a></p>	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.
<p><i>Guía de documentación de Java Enterprise System</i>  <a href="http://docs.sun.com/doc/819-1910">http://docs.sun.com/doc/819-1910</a></p>	Proporciona una descripción de la documentación relacionada con Java Enterprise System. Incluye enlaces a la documentación asociada a los componentes.
<p><i>Visión general técnica de Java Enterprise System</i>  <a href="http://docs.sun.com/doc/819-1924">http://docs.sun.com/doc/819-1924</a></p>	Introduce los conceptos básicos y fundamentos conceptuales de Java Enterprise System. Describe los componentes, arquitectura, procesos y funciones.
<p><i>Guía de planificación de la implementación de Java Enterprise System</i>  <a href="http://docs.sun.com/doc/819-1917">http://docs.sun.com/doc/819-1917</a></p>	Proporciona una introducción a la planificación y diseño de soluciones de implementación empresarial basadas en Java Enterprise System. Presenta los conceptos básicos y los principios de la planificación y diseño de implementación, plantea el ciclo de vida de la solución y proporciona ejemplos y estrategias generales que utilizar cuando se planifiquen soluciones basadas en Java Enterprise System.
<p><i>Guía de administración de usuarios de Sun Java Enterprise System</i>  <a href="http://docs.sun.com/doc/817-5761">http://docs.sun.com/doc/817-5761</a></p>	Ayuda a planificar, implementar y gestionar información acerca de los usuarios de su solución Java Enterprise System. Complementa <i>Guía de planificación de la implementación de Java Enterprise System</i> describiendo los asuntos de gestión del usuario en cada etapa del ciclo de vida de la solución.
<p><i>Serie de ejemplos de implementación de Java Enterprise System: Escenario de evaluación</i>  <a href="http://docs.sun.com/doc/819-0059">http://docs.sun.com/doc/819-0059</a></p>	Describe cómo instalar Java Enterprise System en un equipo, establecer una serie de servicios de red básicos compartidos y configurar cuentas de usuario que puedan acceder a los servicios configurados.
<p><i>Guía de instalación de Java Enterprise System</i>  <a href="http://docs.sun.com/doc/819-0806">http://docs.sun.com/doc/819-0806</a></p>	Le guía en el proceso de instalación de Java Enterprise System para el sistema operativo Solaris™ o el sistema operativo Linux. Muestra cómo seleccionar los componentes que se deben instalar, cómo configurar dichos componentes tras la instalación y cómo comprobar que los componentes configurados funcionan adecuadamente.

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

Documento	Contenido
<i>Guía de migración y actualización de Java Enterprise System</i> <a href="http://docs.sun.com/doc/819-2233">http://docs.sun.com/doc/819-2233</a>	Proporciona la información e instrucciones para actualizar Java Enterprise System para el sistema operativo Solaris™ o el sistema operativo Linux.
<i>Glosario de Java Enterprise System</i> <a href="http://docs.sun.com/doc/816-1931">http://docs.sun.com/doc/816-1931</a>	Define los términos utilizados en la documentación de Java Enterprise System.

## Acceso a los recursos de Sun en línea

Para obtener descargas de productos, servicios profesionales, revisiones y servicio técnico, así como información adicional para los desarrolladores, acuda a:

- Centro de descargas  
<http://www.sun.com/software/download/>
- Servicios profesionales  
<http://www.sun.com/service/sunps/sunone/index.html>
- Servicios empresariales de Sun, revisiones de Solaris y servicio técnico  
<http://sunsolve.sun.com/>
- Información para programadores  
<http://developers.sun.com/prodtech/index.html>

La siguiente ubicación incluye información acerca de Java ES y sus componentes:

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

## Contacto con el servicio técnico de Sun

Si tiene preguntas técnicas acerca de este producto que no obtengan respuestas en la documentación del producto, vaya a <http://www.sun.com/service/contacting>.

## Referencias a sitios Web de terceros relacionados

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

## Sun valora sus comentarios

Sun tiene interés en mejorar su documentación y valora sus comentarios y sugerencias.

Para publicar sus comentarios, acceda a <http://docs.sun.com> y haga clic en "Send Comments" (Enviar comentarios). Se mostrará un formulario en línea en el que deberá indicar el título 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.

Sun valora sus comentarios

# Introducción a la planificación de la implementación

Este capítulo proporciona una breve descripción general de Sun Java™ Enterprise System (Java ES), plantea los conceptos de planificación de la implementación y presenta el ciclo de vida de la solución, que define los distintos pasos para la planificación y diseño de sistemas de software empresariales. Este capítulo incluye los siguientes apartados:

- [“Acerca de Java Enterprise System” en la página 17](#)
- [“Información acerca de la planificación de implementación” en la página 21](#)

## Acerca de Java Enterprise System

Java Enterprise System es una infraestructura de software que proporciona un conjunto completo de servicios de software intermedio para brindar soporte a las aplicaciones empresariales distribuidas en una red o entorno de Internet. Los componentes de Java Enterprise System que proporcionan los servicios se instalan utilizando un programa de instalación común, se sincronizan en un conjunto común de bibliotecas compartidas y comparten una identidad de usuario y un sistema de gestión de la seguridad integrados.

## Servicios del sistema

Los servicios principales de infraestructura que los componentes de Java Enterprise System proporcionan se pueden clasificar de la siguiente manera:

- **Servicios de portal.** Gracias a estos servicios, los trabajadores móviles, los teletrabajadores, los gestores de conocimiento, los socios de negocios, los proveedores y los clientes pueden acceder de forma segura y personalizada a sus portales empresariales desde cualquier lugar externo a la red corporativa sirviéndose de Internet. 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, personalización, seguridad, acceso móvil y búsqueda.
- **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 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 web.** Estos servicios permiten que los componentes distribuidos se comuniquen entre sí y que admitan el desarrollo, la implementación y la gestión de aplicaciones para una amplia gama de servidores, clientes y dispositivos. Estos servicios están basados en la tecnología Java 2 Platform, Enterprise Edition (J2EE™).
- **Servicios de disponibilidad.** Estos servicios proporcionan disponibilidad prácticamente continua y escalabilidad para las aplicaciones y servicios web.

La siguiente tabla muestra las categorías de servicio anteriores y especifica los componentes de Java Enterprise System que prestan los servicios para cada categoría.

**Tabla 1-1** Categorías de servicios de Java Enterprise System

<b>Categoría de servicio</b>	<b>Componentes de Java Enterprise System</b>
Servicios de portal	Portal Server Portal Server Secure Remote Access Access Manager Directory Server Application Server o Web Server
Servicios de colaboración y comunicación	Messaging Server Calendar Server Instant Messaging Access Manager Directory Server Application Server o Web Server
Servicios de identidad de red	Access Manager Directory Server Web Server
Servicios web y de aplicaciones	Application Server Message Queue Web Server
Servicios de disponibilidad	Sun Cluster Sun Cluster Agents

Si desea obtener más información acerca de los servicios, componentes de Java Enterprise System y los conceptos arquitectónicos de Java Enterprise System, consulte *Visión general técnica de Java Enterprise System*, <http://docs.sun.com/doc/819-1924>.

## Servicios incorporados y servicios personalizados

Normalmente, las soluciones de implementación basadas en Java Enterprise System se pueden clasificar en dos categorías generales:

- **Implementaciones 80:20.** Estas soluciones consisten básicamente en servicios proporcionados por Java Enterprise System. Java Enterprise System proporciona alrededor del 80% o más de los servicios.
- **Implementaciones 20:80.** Estas soluciones consisten en un número importante de servicios personalizados y aplicaciones de terceros.

Las categorías 80:20 y 20:80 son generalizaciones. El porcentaje exacto del tipo de servicios ofertado no es importante. Sin embargo, el porcentaje indica la cantidad de personalización que contiene una solución.

Java Enterprise System es adecuado para las implementaciones 80:20 ya que Java ES proporciona un amplio conjunto de servicios. Por ejemplo, es relativamente sencillo implementar un sistema de comunicaciones en la empresa o un sistema de portal en la empresa utilizando los servicios proporcionados por Java Enterprise System.

Para las implementaciones que requieren un desarrollo personalizado, Java Enterprise System admite la creación e integración de servicios y aplicaciones personalizados.

La mayoría de las categorías de servicio que se indican en la [Tabla 1-1 en la página 19](#) se pueden utilizar para ofrecer las implementaciones 80:20. Por ejemplo, los servicios de comunicaciones y colaboración proporcionan servicios de correo electrónico, calendario y mensajería instantánea a los usuarios finales, lo que permite agregar y personalizar el contenido. Igualmente, las categorías de portal empresarial e identidad de red de los servicios permiten instalar y configurar aplicaciones en toda la empresa sin desarrollar o integrar servicios personalizados.

Las soluciones empresariales que requieren el desarrollo personalizado de servicios de plataforma J2EE pueden aprovechar Application Server, Message Queue o Web Server que se proporcionan con los servicios de aplicaciones y web de Java Enterprise System.

Las implementaciones empresariales pueden variar en gran medida en cuanto al número de servicios personalizados que requieren. Debido a la interoperabilidad entre los servicios de Java Enterprise System, puede crear su propio conjunto de servicios personalizados para las necesidades particulares de su empresa.

## Migración a Java Enterprise System

La planificación, diseño e implementación de una solución empresarial que utiliza Java Enterprise System depende en gran medida de su estrategia de implementación actual. Para las empresas que están planificando una solución de implementación por primera vez, la planificación, diseño e implementación están motivados en gran medida por las necesidades específicas de su empresa. Sin embargo, las soluciones de implementación por primera vez no son habituales. Normalmente, se trata de soluciones que utilizan Java Enterprise System para mejorar las soluciones empresariales existentes de versiones anteriores de componentes de Java Enterprise System.

Al sustituir o actualizar las soluciones existentes, se deben tomar medidas adicionales de planificación, diseño e implementación para asegurar que se mantienen los datos existentes y que el software se actualiza adecuadamente a las versiones actuales. A medida que realice el análisis y el diseño definidos en esta guía, tenga en cuenta la preparación y planificación necesarios para sustituir y actualizar los sistemas de software existentes.

Si desea obtener más información acerca de la actualización a la versión actual de Java Enterprise System y las estrategias para la migración desde otras aplicaciones, consulte la *Guía de migración y actualización de Java Enterprise System*.

## Información acerca de la planificación de implementación

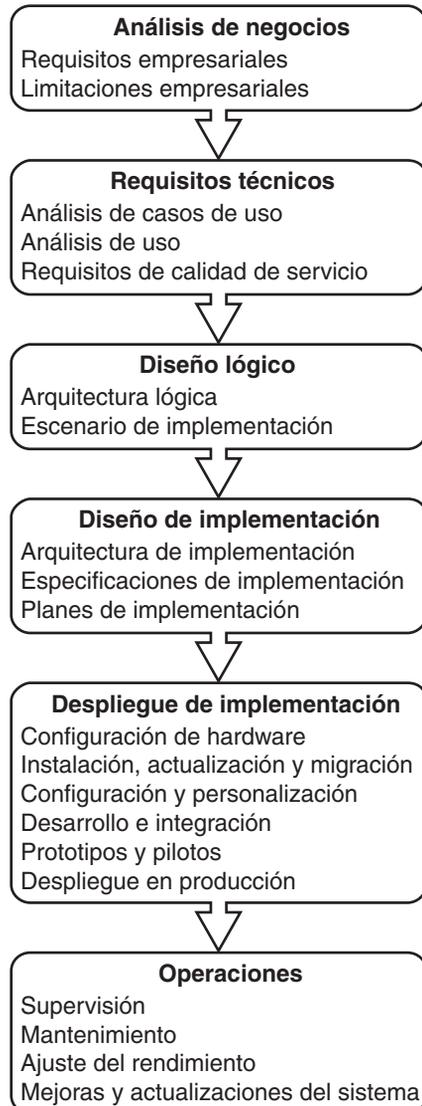
La planificación de la implementación es una etapa crítica para la implementación con éxito de una solución de Java Enterprise System. Cada empresa debe considerar su propio conjunto de objetivos, requisitos y prioridades. Una buena planificación comienza con el análisis de los objetivos de una empresa y el establecimiento de los requisitos empresariales necesarios para cumplir con dichos objetivos. Los requisitos empresariales deben convertirse en requisitos técnicos que se pueden utilizar como una base para diseñar e implementar un sistema que puede satisfacer los objetivos de la empresa.

Una planificación de la implementación de éxito es el resultado de una preparación, análisis y diseño realizados de forma exhaustiva. Los errores que se producen durante el proceso de planificación pueden dar lugar a un sistema que pueda fallar de varias maneras. Pueden surgir problemas importantes derivados de un sistema mal planificado. Por ejemplo, el sistema podría tener un rendimiento inferior al previsto, ser difícil de mantener, demasiado caro de utilizar, utilizar demasiados recursos o podría no cumplir los requisitos de escalamiento para satisfacer el aumento de la demanda.

## Ciclo de vida de la solución

El ciclo de vida de la solución que se muestra en la siguiente figura muestra los pasos en la planificación, diseño e implementación de una solución de software empresarial basada en Java Enterprise System. El ciclo de vida es una herramienta útil para realizar un seguimiento de un proyecto de implementación.

**Figura 1-1** Ciclo de vida de la solución



El ciclo de vida está compuesto por etapas ordenadas. Cada etapa consta de tareas relacionadas que se traducen en resultados que se utilizan en las etapas siguientes. Las tareas de cada etapa son iterativas, y requieren un análisis y diseño completo antes de generar los resultados de la etapa. Las etapas tempranas también pueden ser iterativas. Por ejemplo, durante la etapa de diseño de la implementación, puede descubrir que el análisis en las etapas anteriores es insuficiente y necesita presentarse con más detalle.

Las siguientes secciones de este capítulo describen brevemente cada etapa del ciclo de vida.

## Etapa de análisis de negocios

En la etapa de análisis de negocios, se definen los objetivos empresariales de un proyecto de implementación y se indican los requisitos empresariales que deben satisfacerse para alcanzar dichos objetivos. Cuando se indican los requisitos empresariales, se deben considerar las limitaciones que puedan afectar a la capacidad para alcanzar el objetivo empresarial. A lo largo del ciclo de vida, se medirá el éxito de la planificación de la implementación y en última instancia, la solución de acuerdo con el análisis realizado en la etapa de análisis empresarial.

Durante la etapa de análisis de negocios, se crean documentos de requisitos empresariales que se utilizarán posteriormente como información para la etapa de requisitos técnicos.

Para obtener más información acerca de la etapa de análisis de negocios, consulte [Capítulo 2, “Análisis de negocios” en la página 27](#).

## Etapa de requisitos técnicos

La etapa de requisitos técnicos comienza con los requisitos y limitaciones empresariales definidas durante la etapa de análisis de negocios y las traslada a las especificaciones técnicas que se pueden utilizar para diseñar prácticamente la arquitectura de la implementación. Los requisitos técnicos especifican las funciones de calidad de servicio (QoS), como rendimiento, disponibilidad, seguridad y otras.

Durante la etapa de requisitos técnicos, se crean documentos que contienen la siguiente información:

- Análisis de tareas de usuario y patrones de uso
- Casos de uso que recrean la interacción del usuario con el sistema planificado

- Los requisitos de calidad de servicio obtenidos de la etapa de requisitos empresariales, posiblemente teniendo en cuenta el análisis de las tareas del usuario y los patrones de uso

Los documentos de análisis de uso, casos de uso y requisitos de calidad de servicio resultantes actúan como información para la etapa de diseño lógico del ciclo de vida de la solución. El análisis de uso también desempeña un importante papel en la fase de diseño de la implementación.

Durante la etapa de requisitos técnicos, también puede especificar los requisitos de nivel de servicio que son la base para crear los acuerdos de nivel de servicio (SLA). Los acuerdos de nivel de servicio especifican los términos en los que se debe prestar servicio técnico a los clientes para mantener el sistema y normalmente, se firman como parte de la aprobación del proyecto en la etapa de diseño de implementación.

Si desea más información acerca de los requisitos técnicos, consulte [Capítulo 3, “Requisitos técnicos” en la página 37](#).

## Etapa de diseño lógico

Durante el diseño lógico, con los casos de uso procedentes de la etapa de requisitos técnicos como información, se identifican los componentes de Java Enterprise System necesarios para implementar una solución. Se identifican los componentes que proporcionan compatibilidad a dichos componentes de Java ES y también se identifican algunos componentes personalizados necesarios para satisfacer los requisitos empresariales. A continuación, se asignan los componentes en una arquitectura lógica que muestra las relaciones entre los componentes. La arquitectura lógica no especifica ningún hardware necesario para implementar la solución.

El resultado de la etapa de diseño lógico es la arquitectura lógica. En sí misma, la arquitectura lógica no es suficiente para comenzar el diseño de implementación. También se necesitarán los requisitos de calidad de servicio procedentes de la etapa de requisitos técnicos. La arquitectura lógica y los requisitos de calidad de servicio procedentes de la etapa de requisitos técnicos forman un escenario de implementación. Este escenario es la información que se utilizará para la etapa de diseño de la implementación.

Si desea más información acerca del diseño lógico, consulte [Capítulo 4, “Diseño lógico” en la página 51](#).

## Etapa de diseño de implementación

Durante el diseño de implementación, se asignan los componentes especificados en la arquitectura lógica a un entorno físico, produciendo una arquitectura de implementación global. También se crea una especificación de la implementación, que proporciona detalles que especifican cómo crear la arquitectura de implementación. Además, se crea una serie de planes y especificaciones que detallan distintos aspectos de la implementación de la solución de software.

La aprobación del proyecto se produce en la fase de diseño de implementación. Durante la aprobación del proyecto, se evalúa el coste de la implementación. Si se aprueba, se firman los contratos para el despliegue de la implementación y se adquieren los recursos para crear el proyecto. A menudo, la aprobación del proyecto se produce tras haber detallado la especificación de la implementación. Sin embargo, la aprobación también se puede realizar tras la finalización de la arquitectura de implementación.

Los resultados de la etapa de diseño de la implementación incluyen:

- **Arquitectura de implementación.** Un documento general de diseño que representa la asignación de componentes al hardware y software de red.
- **Especificaciones de implementación.** Especificaciones detalladas que se utilizan como referencia para crear la implementación.
- **Planes de implementación.** Un grupo de planes y especificaciones que cubre distintos aspectos de la implementación de una solución de software empresarial. Los planes de implementación incluyen los planes de migración, de instalación, de administración de usuarios y otros.

Si desea más información acerca del diseño de la implementación, consulte [Capítulo 5, “Diseño de implementación” en la página 71.](#)

## Etapa de implementación

Durante la etapa de implementación, se trabaja a partir de las especificaciones y planes creados durante el diseño de implementación para crear la arquitectura de implementación e implementar la solución. En función de la naturaleza del proyecto de implementación, esta etapa incluye alguna (o todas) de las siguientes tareas:

- Instalación y configuración de la infraestructura de hardware
- Instalación y configuración de software
- Modelado de usuarios y recursos en un diseño de directorios LDAP

- Migración de datos desde los directorios y bases de datos existentes de acuerdo con un plan de administración de usuarios
- Creación y despliegue de implementaciones piloto y prototipo en un entorno de prueba
- Diseño y ejecución de pruebas funcionales para comprobar el cumplimiento de los requisitos del sistema
- Diseño y ejecución de pruebas de estrés para medir el rendimiento en caso de cargas máximas
- Desarrollo e integración de aplicaciones empresariales personalizadas
- Creación de una implementación de producción, que puede introducirse progresivamente (en etapas) en producción

Una vez que una implementación está en el entorno de producción, se iniciará la etapa de operaciones del ciclo de vida de la solución.

Para obtener más información acerca de la etapa de implementación, consulte [Capítulo 6, “Implantación de un diseño de implementación” en la página 107.](#)

## Etapa de operaciones

La etapa de operaciones cubre las tareas necesarias para mantener la implementación ejecutándose y sin problemas. Esta etapa incluye:

- Supervisión de la implementación para asegurar que el sistema funciona de acuerdo con la planificación
- Ajustes del rendimiento para asegurar que el software implementado se ejecuta en los niveles óptimos
- Mantenimiento programado para funcionamiento sin problemas y mantenimiento ocasional según sea necesario
- Actualización de software y hardware según sea necesario

La información acerca de la etapa de operaciones no entra dentro del ámbito de acción de esta guía.

# Análisis de negocios

Durante la etapa de análisis de negocios del ciclo de vida de la solución, se definen los objetivos empresariales mediante el análisis de un problema empresarial y la identificación de los requisitos y limitaciones empresariales para conseguir dicho objetivo.

Este capítulo incluye los siguientes apartados:

- “Acerca del análisis de negocios” en la página 27
- “Definición de requisitos empresariales” en la página 28
- “Definición de limitaciones empresariales” en la página 34

## Acerca del análisis de negocios

El análisis de negocios comienza con la identificación de los objetivos empresariales. A continuación, se analizan los problemas empresariales que se deben resolver y se identifican los requisitos que se deben satisfacer para obtener los objetivos empresariales. Tenga en cuenta también las restricciones empresariales que limitan su capacidad para obtener los objetivos. El análisis de los requisitos y limitaciones empresariales suele dar lugar a un conjunto de documentos en los que se identifican requisitos empresariales.

Utilice este conjunto de documentos como base para obtener los requisitos técnicos que se utilizarán en la etapa de requisitos técnicos. A lo largo del ciclo de vida de la solución, medirá el éxito de la planificación de la implementación y, en última instancia, el éxito de la solución de acuerdo con el análisis realizado en la etapa de análisis de negocio.

# Definición de requisitos empresariales

No existe una fórmula única que pueda identificar todos los requisitos empresariales. Determine los requisitos basándose en la colaboración con los participantes que requieren una solución de software, su conocimiento del ámbito empresarial y el pensamiento creativo aplicado.

Esta sección proporciona algunos factores que se deben tener en cuenta al definir los requisitos empresariales.

## Establecimiento de objetivos empresariales

El análisis de negocios debería establecer los objetivos de un proyecto de implementación. Unos objetivos claros ayudan a la toma de decisiones en cuanto al diseño y evitan que el proyecto se pierda. La comparación de los objetivos empresariales con las operaciones actuales también es un buen indicador para determinar las decisiones sobre el diseño.

### Alcance

Los requisitos empresariales deberían indicar el alcance del proyecto de implementación. Asegúrese de que identifica áreas que se pueden resolver y evite los requisitos abiertos (que no están definidos) que hagan que el objetivo no sea claro o que no se pueda conseguir. Un alcance que no esté claramente definido puede traducirse en un diseño de implementación que no satisfaga las necesidades empresariales o que utilice excesivos recursos.

### Prioridades

Priorice sus objetivos con el fin de asegurar que los aspectos más importantes de la implementación se pueden obtener en primer lugar. Si se cuenta con recursos limitados, es posible que algunos de los objetivos deban ser modificados o pospuestos. Por ejemplo, las implementaciones grandes y complejas requieren normalmente una implementación por etapas de la solución. Mediante el establecimiento de prioridades, se proporciona información valiosa para la toma de decisiones necesarias para el diseño de implementación que los participantes deberán aceptar.

## Calidades críticas

Identifique las áreas que son vitales para el éxito de la implementación para que los participantes y diseñadores se concentren en los criterios más importantes.

## Factores de crecimiento

A medida que establece objetivos empresariales, no tenga en cuenta sólo las necesidades actuales de la organización, sino que deberá anticipar cómo pueden cambiar estas necesidades y aumentar en períodos de tiempo ampliados. El objetivo es que la solución no quede obsoleta de forma prematura.

## Margen de seguridad

El diseño de la solución está basado en los supuestos realizados durante la etapa de análisis de negocios. Estos supuestos pueden ser imprecisos por varios motivos, como datos insuficientes, errores de juicio o eventos externos imprevistos. Asegúrese de que planifica un margen de seguridad no sólo para los objetivos empresariales sino en toda la planificación, de forma que la solución que diseñe pueda afrontar los sucesos imprevistos.

# Comprensión de las necesidades del usuario

Lleve a cabo todos los estudios necesarios para conocer los tipos de usuarios a los que está destinada la solución, sus necesidades y las ventajas previstas que la solución aportará. Por ejemplo, la siguiente lista proporciona una manera para clasificar los usuarios:

- Sólo empleados actuales
- Empleados actuales y anteriores
- Administradores
- Clientes activos
- Todos los clientes
- Sitio privado
- Público en general
- Acceso restringido

La definición clara de las ventajas previstas para los usuarios ayuda a tomar las decisiones de diseño. Por ejemplo, a continuación se muestran algunas de las ventajas que una solución puede ofrecer a sus usuarios:

- Acceso remoto a recursos de la empresa
- Colaboración empresarial
- Simplificación de las tareas diarias
- Uso compartido de recursos para los equipos remotos
- Mayor productividad
- Administración realizada por los usuarios finales

## Desarrollo de requisitos operativos

Expresé los requisitos operativos como un conjunto de requisitos funcionales con objetivos claros. Normalmente, se crean especificaciones operativas para áreas como:

- Funciones de usuario final
- Tiempo de respuesta reducido
- Disponibilidad y tiempo de actividad
- Menor tasa de errores
- Archivo y retención de información

Expresé los requisitos operativos en términos que se puedan cuantificar y que sean fáciles de comprender para todos los participantes. Evite utilizar un lenguaje ambiguo, como “tiempo de respuesta del usuario final adecuado.” A continuación se muestran unos ejemplos de requisitos operativos:

- Capacidad para restaurar los servicios en menos de 10 minutos tras una caída de tensión
- Capacidad para reproducir las últimas 48 horas de entrega de mensajes entrantes
- Transacciones en línea completadas en 60 segundos en los períodos de mayor actividad
- Autenticación de usuario final completada en menos de 4 segundos durante los períodos de mayor actividad

## Compatibilidad de los patrones de uso existentes

Expresé los patrones de uso existentes como objetivos que se puedan medir de forma clara. Las siguientes preguntas pueden determinar dichos objetivos.

- ¿Cómo se utilizan los servicios actuales?
- ¿Cuáles son los patrones de uso (por ejemplo, utilización esporádica, periódica o frecuente)?
- ¿Cuáles son los sitios a los que se conectan normalmente los usuarios?
- ¿Cuál es el tamaño medio de los mensajes enviados por los usuarios?
- ¿Cuál es el promedio de transacciones que los usuarios realizan por día u hora?

Estudie los usuarios que acceden a los servicios. Factores como cuándo acceden los usuarios a los servicios existentes y su duración son elementos clave para identificar los objetivos. Si la trayectoria de su organización no puede ofrecer estos patrones, estudie los datos de organizaciones parecidas.

## Comprensión de la cultura corporativa

Los análisis de requisitos deberían tomar en cuenta varios aspectos de la cultura y política corporativas. Si no se presta atención a la cultura corporativa, se puede obtener una solución que no sea bien recibida o que sea difícil de implementar.

### Participantes

Identifique las personas y organizaciones que tengan interés en el éxito de la solución propuesta. Todos estos participantes deben involucrarse de forma activa en la definición de los objetivos y requisitos empresariales. Si uno de estos participantes no se involucra o no está informado de los cambios previstos, los planes podrían tener defectos importantes. Dicho participante podría incluso bloquear la implementación de la solución.

### Normas y políticas

Asegúrese de que comprende las normas y políticas de la organización que solicita la solución. Estas normas y políticas pueden tener repercusiones sobre los aspectos técnicos del diseño, la selección del producto y la metodología de la implementación.

Un ejemplo es la confidencialidad de los datos del personal que poseen la organización de recursos humanos o los directores de la unidad. Otro ejemplo serían los procedimientos de la empresa para la gestión del cambio. Las políticas de gestión del cambio podrían afectar de forma drástica la aceptación de una solución e influir en la metodología de implementación y en la programación.

## Requisitos obligatorios

Los requisitos obligatorios varían en gran medida en función de la naturaleza de la actividad comercial. Estudie y comprenda todos los requisitos obligatorios que puedan afectar a la implementación. Muchas empresas y organismos gubernamentales requieren que se cumplan las normas de accesibilidad. Al implementar soluciones globales, tenga en cuenta las leyes y normativas extranjeras. Por ejemplo, muchos países europeos cuentan con estrictos controles sobre el almacenamiento de información personal.

## Seguridad

Algunos de los objetivos que identifique pueden derivar en problemas de seguridad que deberían tenerse en cuenta. Destaque los objetivos de seguridad específicos que sean básicos para la solución. Por ejemplo:

- Acceso autorizado a información privada
- Acceso a información confidencial según el puesto
- Comunicación segura entre ubicaciones remotas
- Invocación de aplicaciones remotas en sistemas locales
- Transacciones seguras con otras empresas y organizaciones
- Cumplimiento de las políticas de seguridad

## Distribución del sitio

La distribución geográfica de los sitios y el ancho de banda entre los sitios puede afectar a las decisiones de diseño. Además, es posible que algunos sitios requieran una gestión local.

Dichas consideraciones geográficas pueden aumentar los costes de formación del proyecto, su complejidad y otra serie de factores. Indique claramente los requisitos que se deriven de la distribución geográfica de los sitios. Destaque los sitios que son vitales para el éxito del diseño.

## Utilización de un planteamiento progresivo

A menudo, se considera una solución de software como un sistema completo y global. Sin embargo, frecuentemente se obtiene la implementación de todo el sistema de forma progresiva mediante pasos controlados.

Al adoptar un planteamiento progresivo, normalmente se diseña una programación que ofrece distintos hitos que en última instancia se traducen en la solución completa. Además, deberá considerar los planes a corto plazo para aspectos de la solución completa que se retrasan para una implementación posterior.

El planteamiento progresivo proporciona estas ventajas:

- Puede adaptar los cambios de requisitos debidos al crecimiento de la empresa.
- Puede aprovechar la infraestructura existente a medida que realiza la transición a la implementación de despliegue final.
- Puede adaptar los requisitos de gastos de capital.
- Puede aprovechar una pequeña aportación de recursos humanos.
- Permite la posibilidad de crear alianzas.

## Acuerdos de nivel de servicio

Un acuerdo de nivel de servicio (SLA) especifica los requisitos mínimos de rendimiento y, en caso de que no se cumplan dichos requisitos, el nivel y el alcance del servicio al cliente que se debe proporcionar. Un acuerdo de nivel de servicio está basado en los requisitos empresariales definidos durante el análisis de negocios, que posteriormente se especifican como requisitos de nivel de servicio durante la etapa de requisitos técnicos. El SLA se firma durante la aprobación del proyecto, que se produce en la etapa de diseño de implementación.

Debería desarrollar un SLA en áreas como tiempo de actividad, tiempo de respuesta, tiempo de entrega de mensajes y recuperación tras desastres. Un SLA debería incorporar elementos como la descripción general del sistema, las funciones y responsabilidades de las organizaciones de servicio técnico, el método de medición de los niveles de servicio, las solicitudes de cambio, etc. La identificación de las expectativas de la empresa con respecto a la disponibilidad del sistema es un elemento clave para determinar el alcance de un SLA.

## Definición de limitaciones empresariales

Las limitaciones empresariales desempeñan un importante papel a la hora de determinar la naturaleza de un proyecto de implementación. Una de las claves para realizar un diseño de implementación con éxito es encontrar la forma óptima para satisfacer los requisitos empresariales a pesar de las limitaciones empresariales conocidas. Las limitaciones empresariales pueden ser limitaciones fiscales, físicas (por ejemplo, capacidad de la red), de tiempo (por ejemplo, finalización antes de eventos importantes como la siguiente junta anual), o cualquier otra limitación que pueda considerarse como un factor que influya en la consecución del objetivo empresarial.

Esta sección proporciona algunos factores que se deben considerar al definir las limitaciones empresariales.

## Aspectos relacionados con la migración

Normalmente, un proyecto de implementación sustituye o complementa la infraestructura de software y los datos. Una solución nueva deberá migrar los datos y procedimientos de la infraestructura existente a la nueva solución, a menudo manteniendo la interoperabilidad con las aplicaciones existentes. Es necesario realizar un análisis de la infraestructura actual para determinar la importancia de los aspectos relacionados con la migración en la solución propuesta.

## Obligaciones de programación

La programación de la implementación de una solución puede afectar a las decisiones de diseño. Una programación agresiva puede dar lugar a una reducción de los objetivos, a un cambio de las prioridades o a la adopción de un planteamiento progresivo para la solución. En una programación, deben existir hitos importantes que también se deben tener en cuenta. Los hitos se pueden definir mediante eventos internos, como despliegues de servicio programados o eventos externos, como la fecha de apertura de un semestre escolar.

## Limitaciones presupuestarias

La mayoría de los proyectos de implementación deben cumplir con un presupuesto. A la hora de considerar el coste de la creación de la solución propuesta y de los recursos necesarios para mantener la solución en un tiempo de vida específico, se deben tener en cuenta los siguientes aspectos:

- **Hardware e infraestructura de red existente.** La infraestructura existente puede afectar al diseño del sistema.
- **Recursos de desarrollo necesarios para implementar la solución.** Unos recursos de desarrollo limitados, que incluyen hardware, software y recursos humanos, puede ser un buen indicador para una implementación gradual. Tendrá que utilizar varias veces los mismos recursos o equipos de desarrollo en cada etapa de la implementación.
- **Mantenimiento, administración y servicio técnico.** Analice los recursos disponibles para administrar, mantener y prestar servicio técnico a los usuarios del sistema. Unos recursos limitados pueden tener un efecto en las decisiones de diseño.

## Coste de propiedad

Además del mantenimiento, administración y servicio técnico, analice otros factores que afectan al coste de propiedad. Las actualizaciones de hardware y software pueden ser necesarias, la repercusión de la solución en los costes de telecomunicaciones o eléctricos y otros factores influyen en los gastos. Los acuerdos de nivel de servicio que especifican los niveles de disponibilidad de la solución también afectan al coste de propiedad, ya que requieren una mayor redundancia.

La implementación de una solución debería proporcionar una rentabilidad de la inversión en la solución. Normalmente, los análisis de la rentabilidad de la inversión implican la medición de ventajas financieras obtenidas a partir de un gasto de capital.

La estimación de las ventajas financieras de una solución implica un análisis detallado de los objetivos que se van a conseguir comparándolos con las formas alternativas de obtener dichos objetivos y con el coste de no realizar ningún tipo de acción.

Definición de limitaciones empresariales

# Requisitos técnicos

Durante la etapa de requisitos técnicos del ciclo de vida de la solución se realiza un análisis de uso, se identifican los casos de uso y se determinan los requisitos de calidad de servicio para la solución de implementación propuesta.

Este capítulo incluye los siguientes apartados:

- “Información acerca de los requisitos técnicos” en la página 37
- “Análisis de uso” en la página 38
- “Casos de uso” en la página 40
- “Requisitos de calidad de servicio” en la página 41
- “Requisitos de nivel de servicio” en la página 50

## Información acerca de los requisitos técnicos

El análisis de los requisitos técnicos comienza con los documentos de requisitos empresariales creados durante la etapa de análisis empresarial del ciclo de vida de la solución. Con el análisis empresarial como base, puede realizar las siguientes opciones:

- Realizar un análisis de uso para ayudar a determinar las condiciones de carga previstas.
- Crear los casos de uso que modelan la interacción de usuario típica con el sistema.
- Crear un conjunto de requisitos de calidad de servicio (QoS) que definen cómo debe comportarse una solución implementada en áreas como tiempo de respuesta, disponibilidad, seguridad y otras.

Los requisitos de calidad de servicio se obtienen a partir del análisis de uso y los casos de uso, teniendo en cuenta los requisitos empresariales y las limitaciones identificadas anteriormente.

Los requisitos de calidad de servicio se comparan posteriormente con las arquitecturas lógicas en la etapa de diseño lógico para formar un escenario de implementación. El escenario de implementación es la información principal para la etapa de diseño de implementación del ciclo de vida de la solución.

En cuanto al análisis empresarial, no existe una fórmula sencilla para el análisis de requisitos técnicos que genere los análisis de uso, casos de uso y requisitos del sistema. El análisis de requisitos técnicos requieren una comprensión del ámbito empresarial, los objetivos empresariales y la tecnología del sistema subyacente.

## Análisis de uso

El análisis de uso implica la identificación de los distintos usuarios de la solución que está diseñando y la determinación de los patrones de uso de dichos usuarios. La información que recopile proporciona una base para estimar las condiciones de carga del sistema. La información de análisis de uso también es útil al definir la importancia de los distintos casos de uso, como se describe en “Casos de uso” en la página 40.

Durante el análisis de uso, debería realizar entrevistas a usuarios siempre que sea posible, buscar los datos existentes sobre patrones de uso y realizar entrevistas con los administradores y constructores de los sistemas anteriores. La siguiente tabla muestra los factores que se deben considerar al realizar un análisis de uso.

**Tabla 3-1** Factores de análisis de uso

Tema	Descripción
Número y tipo de usuarios	<p data-bbox="575 1168 1243 1220">Identifique el número de usuarios que debe admitir la solución y clasifique dichos usuarios en categorías según sea necesario.</p> <p data-bbox="575 1237 1243 1263">Por ejemplo:</p> <ul data-bbox="575 1272 1243 1480" style="list-style-type: none"> <li data-bbox="575 1272 1243 1359">• Una solución de empresa a cliente (B2C) puede tener un gran número de visitantes, pero sólo un pequeño número de usuarios se registra y realiza transacciones comerciales.</li> <li data-bbox="575 1367 1243 1480">• Una solución de empresa a empleado (B2E) admite normalmente a todos los empleados, aunque algunos empleados pueden necesitar acceder desde fuera de la red corporativa.</li> </ul> <p data-bbox="575 1506 1243 1555">En una solución B2E, los directores necesitarán autorización en áreas a las que los empleados normales no podrán acceder.</p>

**Tabla 3-1** Factores de análisis de uso (*continúa*)

<b>Tema</b>	<b>Descripción</b>
Usuarios activos e inactivos	<p>Identifique los patrones de uso y las relaciones de usuarios activos e inactivos.</p> <p>Los usuarios activos son los usuarios que han iniciado sesión en el sistema e interactúan con los servicios del sistema. Los usuarios inactivos pueden ser usuarios que no se han registrado, usuarios que se registran pero que no interactúan con los componentes del sistema o usuarios que se encuentran en la base de datos pero nunca se registran.</p>
Usuarios administrativos	<p>Identifique los usuarios que acceden al sistema implementado para supervisar, actualizar y admitir la implementación.</p> <p>Determine todos los patrones de uso administrativo específicos que pueden afectar a los requisitos técnicos (por ejemplo, administración de la implementación desde fuera del servidor de seguridad).</p>
Patrones de uso	<p>Identifique cómo los distintos tipos de usuarios acceden al sistema y proporcionan objetivos de uso previsto.</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> <li>• ¿Existen horas pico cuando cae el rendimiento?</li> <li>• ¿Cuáles son las horas de negocio habituales?</li> <li>• ¿Se distribuyen globalmente los usuarios?</li> <li>• ¿Cuál es la duración prevista de la conectividad del usuario?</li> </ul>
Crecimiento de usuarios	<p>Determine si el tamaño de la base de usuarios es fija o si la implementación prevé el crecimiento en el número de usuarios.</p> <p>Si se prevé que la base de usuarios aumente, intente crear provisiones razonables del crecimiento.</p>
Transacciones de usuario	<p>Identifique el tipo de transacciones de usuario que se deben admitir. Estas transacciones de usuario se pueden traducir en casos de uso.</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> <li>• ¿Qué tareas realizan los usuarios?</li> <li>• Cuando los usuarios inician sesión, ¿se mantienen en el sistema? Normalmente, ¿realizan varias tareas y terminan la sesión?</li> <li>• ¿Es necesario realizar calendarios comunes, conferencias vía web e implementación de páginas web internas para poder realizar una importante colaboración entre los usuarios?</li> </ul>

**Tabla 3-1** Factores de análisis de uso (*continúa*)

Tema	Descripción
Estudios de usuario y datos estadísticos	<p>Use los estudios de usuario previos y otras fuentes para determinar los patrones de comportamiento de usuario.</p> <p>A menudo, las empresas u organizaciones del sector cuentan con estudios de investigación de usuarios que puede utilizar para obtener importante información acerca de los usuarios. Los archivos de registro para las aplicaciones existentes pueden contener datos estadísticos útiles para realizar los cálculos de un sistema.</p>

## Casos de uso

Los casos de uso modelan la interacción de los usuarios con la interacción que está diseñando y describen el flujo completo de una operación desde la perspectiva de un usuario final. El establecimiento de prioridades en el diseño basándose en un conjunto completo de casos de uso asegura un planteamiento continuo en la prestación de las funciones previstas. Los casos de uso son la información principal del diseño lógico.

Asigne importancia relativa a los casos de uso, siendo los casos de uso más importante los que representan las tareas más comunes. La importancia de los casos de uso permite centrarse en las decisiones de diseño de los servicios del sistema que se utilizan con mayor frecuencia.

Los casos de uso se pueden describir en dos niveles.

- **Informes de caso de uso.** Descripciones de casos de uso individuales, que incluyen los flujos principales y alternativos de eventos.
- **Diagramas de caso de uso.** Diagramas que muestran las relaciones entre los actores y los casos de uso, presentando una organización más formal del flujo de eventos. Los diagramas de caso de uso son útiles para modelar casos de uso largos o complejos. Normalmente, se utilizan los estándares de idioma de modelos unificado (UML) para crear los diagramas de caso de uso.

# Requisitos de calidad de servicio

Requisitos de calidad de servicio (QoS) son especificaciones técnicas que especifican la calidad del sistema de las funciones como rendimiento, disponibilidad, escalabilidad y facilidad de mantenimiento. Los requisitos de QoS están impulsados por las necesidades empresariales especificadas en los requisitos empresariales. Por ejemplo, si los servicios deben estar disponibles las 24 horas del día durante todo el año, el requisito de disponibilidad debe cumplir con este requisito empresarial.

La siguiente tabla muestra las calidades de sistema que normalmente forman la base de los requisitos de QoS.

**Tabla 3-2** Calidades de sistema que afectan a los requisitos de QoS

Calidad del sistema	Descripción
Rendimiento	La medición del tiempo de respuesta y de rendimiento 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, a menudo expresado como el <i>tiempo de actividad</i> de un sistema.
Escalabilidad	La capacidad de agregar funciones (y usuarios) a un sistema implementado a lo largo del tiempo. La escalabilidad normalmente implica agregar recursos al sistema, pero no debería requerir cambios en la arquitectura de implementación.
Security (Seguridad)	Combinación compleja de factores que describe la integridad de un sistema y sus usuarios. La seguridad incluye la autenticación y autorización de usuarios, seguridad de los datos y acceso seguro a un sistema implementado.
Capacidad latente	La capacidad de un sistema de gestionar el uso de cargas máximas inusual sin recursos adicionales. La capacidad latente es un factor en las calidades de disponibilidad, rendimiento y escalabilidad.
Facilidad de mantenimiento	La facilidad con que un sistema implementado puede mantenerse, incluidas tareas tales como la supervisión del sistema, la reparación de los problemas que surjan y la actualización de los componentes de hardware y software.

Las calidades del sistema están estrechamente relacionadas entre sí. Los requisitos de una calidad del sistema pueden afectar a los requisitos y al diseño de otras calidades del sistema. Por ejemplo, unos mayores niveles de seguridad podrían afectar al rendimiento, que a su vez podría afectar a la disponibilidad. La integración de servidores adicionales para solucionar problemas de disponibilidad podría afectar a los costes de mantenimiento (facilidad de mantenimiento).

El conocimiento de las relaciones entre las calidades del sistema y los cambios que se deben realizar es un elemento clave para diseñar un sistema que satisfaga con éxito los requisitos empresariales y se adecue a las limitaciones empresariales impuestas.

Las siguientes secciones describen con más detalle las calidades de sistema que afectan al diseño de implementación, proporcionando directrices sobre los factores que se deben considerar al formular los requisitos de QoS. También se incluye una sección relativa a los requisitos de nivel de servicio, que forman la base de los acuerdos de nivel de servicio.

## Rendimiento

Normalmente, los requisitos empresariales expresan el rendimiento en términos no técnicos que especifican el tiempo de respuesta. Por ejemplo, un requisito empresarial para el acceso basado en web puede indicar lo siguiente:

Los usuarios esperan un tiempo de respuesta razonable al iniciar sesión, normalmente, no superior a cuatro segundos.

A partir de este requisito empresarial, examine todos los casos de uso para determinar cómo expresar este requisito a un nivel de sistema. Es posible que en algunos casos desee incluir las condiciones de carga de usuarios determinadas durante el análisis de uso. Exprese el requisito de rendimiento para cada caso de uso en términos de tiempo de respuesta de acuerdo con las condiciones de carga especificadas o de tiempo de respuesta más rendimiento. También debe especificar el número permitido de errores.

A continuación se muestran dos ejemplos de cómo especificar los requisitos del sistema en cuanto a rendimiento:

La respuesta de actualización de la página web no debe ser superior a cuatro segundos en todo el día, con un muestreo de intervalo de 15 minutos, con menos de 3,4 errores por millón de transacciones.

Durante los períodos pico definidos, el sistema debe permitir 25 inicios de sesión seguros por segundo con un tiempo de respuesta no superior a 12 segundos para cualquier usuario y menos de 3,4 errores por millón de transacciones.

Los requisitos de rendimiento están estrechamente relacionados con los requisitos de disponibilidad (cómo afecta la recuperación tras error al rendimiento) y la capacidad latente (la capacidad disponible para tratar cargas pico inusuales).

## Disponibilidad

La disponibilidad es una forma de especificar el tiempo de actividad de un sistema y normalmente se mide como el porcentaje de tiempo que el sistema está accesible para los usuarios. El tiempo que el sistema no está accesible (tiempo de inactividad) puede ser la consecuencia de un error de hardware, de software, de la red o de cualquier otro factor (como una caída de tensión) que produce que el sistema no esté operativo. El tiempo de inactividad programado para la realización de tareas de servicio (mantenimiento y actualizaciones) no se considera tiempo de inactividad. Una ecuación básica para calcular la disponibilidad del sistema en términos de porcentaje de tiempo de actividad es:

$$\text{Disponibilidad} = \text{tiempo de actividad} / (\text{tiempo de actividad} + \text{tiempo de inactividad}) * 100\%$$

Normalmente, la disponibilidad se mide por el número de “nueves” que se pueden obtener. Por ejemplo, una disponibilidad del 99% tiene dos nueves. La especificación de nueves adicionales afecta en gran medida al diseño de implementación. La siguiente tabla cuantifica el tiempo de inactividad no programado para los nueves adicionales de disponibilidad en un sistema que está operativo 24x7 todos los días del año (un total de 8.760 horas).

**Tabla 3-3** Tiempo de inactividad no programado para un sistema que está operativo todo el año (8.760 horas)

Número de nueves	Porcentaje disponible	Tiempo de inactividad no programado
2	99%	88 horas
3	99.9%	9 horas
4	99.99%	45 minutos
5	99.999%	5 minutos

## Sistemas tolerantes a fallos

Los requisitos de disponibilidad de cuatro o cinco nueves requieren normalmente un sistema que sea tolerante a los fallos. Este tipo de sistemas debe ser capaz de continuar prestando el servicio incluso durante un fallo de hardware o software. Normalmente, esto se consigue mediante la redundancia en hardware (como CPU, memoria y dispositivos de red) y en el software que proporciona los servicios clave.

Un único punto de fallo es un componente de hardware o software que es parte de una ruta crítica pero que no tiene una copia de seguridad mediante componentes redundantes. Si este componente falla, el sistema no podrá prestar el servicio. Al diseñar un sistema tolerante a fallos, se debe identificar y eliminar los puntos de fallo potenciales.

Los sistemas tolerantes a fallos pueden ser caros de implementar y mantener. Asegúrese de que comprende la naturaleza de los requisitos empresariales en cuanto a disponibilidad y considere las estrategias y costes de las soluciones de disponibilidad que satisfacen dichos requisitos.

## Establecimiento de las prioridades en cuanto a disponibilidad del servicio

Desde la perspectiva del usuario, la disponibilidad a menudo se aplica más servicio a servicio que en cuanto a la disponibilidad general de todo el sistema. Por ejemplo, normalmente el hecho de que los servicios de mensajería instantánea no estén disponibles tiene una repercusión pequeña (o prácticamente nula) en la disponibilidad del resto de servicios. Sin embargo, el hecho de que otros servicios de los que dependen muchos otros servicios (como Directory Server) no estén disponibles, tiene una repercusión mucho mayor. Unas especificaciones de mayor disponibilidad deberían hacer referencia a casos de uso específicos y análisis de uso que requieren la mayor disponibilidad.

Es útil mostrar las necesidades de disponibilidad de acuerdo con un conjunto de prioridades ordenadas. La siguiente tabla establece prioridades sobre la disponibilidad de los distintos tipos de servicios.

**Tabla 3-4** Disponibilidad de servicios por prioridad

Prioridad	Tipo de servicio	Descripción
1	Vital	Servicios que deben estar disponibles en todo momento. Por ejemplo, servicios de bases de datos (como los directorios LDAP) para las aplicaciones.
2	Debe estar disponible	Servicios que deben estar disponibles, pero que pueden estar disponibles con un rendimiento inferior. Por ejemplo, la disponibilidad del servicio de mensajería puede no ser vital en algunos entornos empresariales.

**Tabla 3-4** Disponibilidad de servicios por prioridad (*continúa*)

Prioridad	Tipo de servicio	Descripción
3	Se puede posponer	Servicios que deben estar disponibles en un período de tiempo determinado. Por ejemplo, la disponibilidad de los servicios de calendario puede no ser vital en algunos entornos empresariales.
4	Opcional	Servicios que se pueden posponer indefinidamente. Por ejemplo, en algunos entornos, los servicios de mensajería instantánea se pueden considerar útiles pero no necesarios.

## Pérdida de servicios

El diseño de disponibilidad incluye la consideración de lo que sucede cuando la disponibilidad se ve comprometida o cuando se pierde un componente. Esto incluye la consideración de si los usuarios conectados deben reiniciar la sesión y cómo un fallo en un área afecta al resto de áreas del sistema. Los requisitos de QoS deberían tener en cuenta estos escenarios y especificar cómo la implementación reacciona antes estas situaciones.

## Escalabilidad

La escalabilidad es la capacidad para agregar capacidad a un sistema, de forma que pueda admitir carga adicional de los usuarios existentes o de una base de usuarios mayor. Normalmente, la escalabilidad requiere agregar recursos, pero no requiere cambios en el diseño de la arquitectura de la implementación ni produce una pérdida de servicio debido al tiempo necesario para agregar recursos adicionales.

Al igual que con la disponibilidad, la escalabilidad hace más referencia a los servicios individuales proporcionados por un sistema en vez de a todo un sistema. Sin embargo, para los servicios en los que dependen otros servicios, como Directory Server, la escalabilidad puede tener un impacto en todo el sistema.

No es necesario especificar los requisitos de escalabilidad con los requisitos de QoS a menos que el crecimiento previsto de la implementación se indique claramente en los requisitos empresariales. Sin embargo, durante la etapa de diseño empresarial del ciclo de vida de la solución, la arquitectura de implementación siempre debe agregar más tolerancia para escalar los sistemas, incluso si no se ha especificado los requisitos de QoS de escalabilidad.

## Cálculo del crecimiento previsto

El cálculo del crecimiento previsto de un sistema para determinar los requisitos de escalabilidad implica el trabajo con previsiones, estimaciones y predicciones que es posible que no se cumplan. A continuación se muestran tres claves para desarrollar los requisitos para un sistema escalable.

- **Estrategia de diseño de alto rendimiento.** Durante la especificación de requisitos de rendimiento, incluya la capacidad latente para gestionar las cargas que pueden aumentar a lo largo del tiempo. Asimismo, maximice la disponibilidad dentro de las restricciones presupuestarias. Esta estrategia permite absorber el crecimiento y programar mejor los hitos para escalar el sistema.
- **Implementación gradual.** La implementación gradual permite programar la adición de recursos. Especifique hitos claros para escalar el sistema. Normalmente, los hitos son requisitos basados en la carga, coordinados con fechas específicas para evaluar la escalabilidad.
- **Continua supervisión de rendimiento.** La supervisión del rendimiento ayuda a determinar cuándo agregar recursos al sistema. Los requisitos para supervisar el rendimiento pueden proporcionar unas directrices a los operadores y administradores responsables de realizar el mantenimiento y las actualizaciones.

La siguiente tabla muestra los factores que se deben tener en cuenta para determinar los requisitos de escalabilidad.

**Tabla 3-5** Factores de escalabilidad

Tema	Descripción
Analice los patrones de uso	Comprenda los patrones de uso de la base de usuarios actual (o prevista) mediante el análisis de los datos existentes. En ausencia de datos actuales, analice los datos del sector o las previsiones del mercado.
Realice un diseño para una escala máxima razonable	Realice el diseño teniendo en cuenta los objetivos de escala máxima necesaria para la demanda conocida, así como para la demanda posible.  A menudo, se trata de un cálculo de 24 meses basado en la evaluación de rendimiento de la carga de usuarios existentes y unas previsiones razonables de la carga futura. El período de tiempo de este cálculo depende en gran medida de la fiabilidad de las proyecciones.

**Tabla 3-5** Factores de escalabilidad (*continúa*)

Tema	Descripción
Defina hitos adecuados	<p>Despliegue el diseño de implementación de forma gradual para satisfacer los requisitos a corto plazo con un colchón que permita el crecimiento imprevisto. Defina hitos para agregar recursos de sistema.</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> <li>• Adquisición de capital (como trimestral o anualmente)</li> <li>• Tiempo medio para adquirir hardware y software (como de una a seis semanas)</li> <li>• Búfer (10% a 100%, en función de las expectativas de crecimiento)</li> </ul>
Incorpore la tecnología emergente	<p>Conozca la nueva tecnología, como procesadores más rápidos y servidores Web, y cómo puede afectar al rendimiento de la arquitectura subyacente.</p>

## Requisitos de seguridad

La seguridad es un tema complejo que involucra a todos los niveles de un sistema implementado. El desarrollo de los requisitos de seguridad se centra en la identificación de las amenazas de seguridad y el desarrollo de una estrategia para combatirlas. Este análisis de seguridad incluye los siguientes pasos:

1. Identificación de los activos críticos
2. Identificación de las amenazas para dichos activos
3. Identificación de las vulnerabilidades que crean amenazas que pueden crear riesgos para la organización
4. Desarrollo de un plan de seguridad que mitiga el riesgo para la organización

El análisis de los requisitos de seguridad debería incluir una representación de los participantes de la organización, incluidos los directivos, analistas de negocios y personal de tecnología de la información. A menudo, una organización nombra a un arquitecto de seguridad para que dirija el proceso de diseño e implementación de las medidas de seguridad.

La siguiente sección describe algunas de las áreas cubiertas en el plan de seguridad.

## Elementos de un plan de seguridad

La planificación de la seguridad de un sistema es parte del diseño de implementación que es vital para llevar a cabo la implementación con éxito. Cuando planifique la seguridad, tenga en cuenta lo siguiente:

- **Seguridad física.** La seguridad física es el acceso físico a los enrutadores, servidores, salas de servidores, centros de datos y otras partes de la infraestructura. Otras medidas de seguridad pueden verse en peligro si una persona no autorizada puede entrar en una sala de servidores y desconectar los enrutadores.
- **Seguridad de red.** La seguridad de red consiste en el acceso a la red mediante servidores de seguridad, zonas de acceso seguro, listas de control de acceso y puertos de acceso. Para la seguridad de la red se desarrollan estrategias para acceso no autorizado, falsificación y ataques de denegación de servicio (DoS).
- **Seguridad de las aplicaciones y de datos de aplicación.** La seguridad de los datos de aplicaciones y de las aplicaciones cubre el acceso a las cuentas de usuario, datos corporativos y aplicaciones empresariales mediante políticas y procedimientos de autenticación y autorización. Esta área incluye la definición de las siguientes políticas:
  - Política de contraseña
  - Derechos de acceso, como administración delegada a los usuarios frente al acceso de administrador
  - Desactivación de una cuenta
  - Control de acceso
  - Políticas de cifrado, como el transporte seguro de datos y la utilización de certificados para firmar los datos
- **Prácticas de seguridad personal.** Una política de seguridad de la empresa define el entorno de trabajo y las prácticas que deben cumplir todos los usuarios para asegurar que otras medidas de seguridad tienen un rendimiento como el diseñado. Normalmente, se desarrolla una guía o manual sobre la seguridad y se ofrece formación a los usuarios acerca de prácticas de seguridad. Para obtener una política de seguridad general eficaz, las prácticas de seguridad deben ser parte de la cultura de la empresa.

## Capacidad latente

La capacidad latente es la capacidad de que una implementación gestione una utilización de carga máxima inusual sin añadir recursos. Normalmente, no se especifican los requisitos de QoS directamente basándose en la capacidad latente, sino que este parámetro de calidad del sistema es un factor que afecta a la disponibilidad, el rendimiento y la escalabilidad del sistema.

## Requisitos de mantenimiento

Esta característica es la facilidad con que un sistema implementado puede mantenerse, incluidas tareas tales como la supervisión del sistema, la reparación de los problemas que surjan, la adición y eliminación de usuarios del sistema, y la actualización de los componentes de hardware y software.

Al planificar los requisitos de planificación en cuanto facilidad de mantenimiento, se deben tener en cuenta los temas que se muestran en la siguiente tabla.

**Tabla 3-6** Temas para requisitos de mantenimiento

Tema	Descripción
Planificación del tiempo de inactividad	<p>Identifique las tareas de mantenimiento que requieren que determinados servicios específicos no estén disponibles o que sólo lo estén de manera parcial.</p> <p>Algunas operaciones de mantenimiento y actualizaciones se pueden realizar sin que los usuarios se vean afectados, mientras que otras requieren la interrupción del servicio. Siempre que sea posible, programe con los usuarios las actividades de mantenimiento que requieren un tiempo de inactividad, dejando que los usuarios planifiquen el tiempo de inactividad.</p>
Patrones de uso	<p>Identifique los patrones de uso para determinar el mejor momento para programar el mantenimiento.</p> <p>Por ejemplo, en los sistemas en los que la utilización pico se realiza durante el horario comercial normal, programe el mantenimiento durante la tarde o el fin de semana. Para los sistemas distribuidos geográficamente, la identificación de estos momentos puede ser más complicado.</p>

**Tabla 3-6** Temas para requisitos de mantenimiento (*continúa*)

Tema	Descripción
Disponibilidad	<p>La facilidad de mantenimiento es a menudo un reflejo del diseño de disponibilidad. Las estrategias para minimizar el tiempo de inactividad para el mantenimiento y actualizaciones se centran en la estrategia de disponibilidad. Los sistemas que requieren un alto grado de disponibilidad tienen oportunidades limitadas para las operaciones de mantenimiento, actualización y reparación.</p> <p>Las estrategias para tratar los requisitos de disponibilidad afectan a la forma en que se tratan las operaciones de mantenimiento y de actualización. Por ejemplo, en los sistemas que están distribuidos geográficamente, el servicio puede depender de la capacidad de dirigir las cargas de trabajo a servidores remotos durante los períodos en los que se realiza el mantenimiento.</p> <p>Asimismo, los sistemas que requieren un alto grado de disponibilidad pueden requerir soluciones más sofisticadas que automatizan el reinicio de los sistemas con una escasa intervención humana.</p>
Diagnóstico y supervisión	<p>Puede mejorar la estabilidad de un sistema mediante la ejecución periódica de herramientas de diagnóstico y supervisión para identificar las áreas con problemas.</p> <p>La supervisión periódica de un sistema puede evitar los problemas antes de que se produzcan, ayudar a equilibrar las cargas de trabajo de acuerdo con las estrategias de disponibilidad y mejorar la planificación del mantenimiento y el tiempo de inactividad.</p>

## Requisitos de nivel de servicio

Un acuerdo de nivel de servicio (SLA) especifica los requisitos mínimos de rendimiento y, en caso de que no se cumplan dichos requisitos, el nivel y el alcance de servicio al cliente que se debe proporcionar. Los requisitos de nivel de servicio son requisitos de sistema que especifican las condiciones en las que está basado el SLA.

Al igual que con los requisitos de QoS, los requisitos de nivel de servicio se obtienen a partir de los requisitos empresariales y representan una garantía acerca de la calidad general del sistema que debe satisfacer el sistema implementado. Debido a que el acuerdo de nivel de servicio se considera como un contrato, la especificación de los requisitos de nivel de servicio deberían ser muy específicos y no presentar ningún tipo de ambigüedad. Los requisitos de nivel de servicio definen exactamente en qué condiciones se prueban los requisitos y definen de forma precisa las situaciones que constituyen un incumplimiento de dichos requisitos.

# Diseño lógico

Durante la etapa de diseño lógico del ciclo de vida de la solución, se diseña una arquitectura lógica que muestra las relaciones entre los componentes lógicos de la solución. La arquitectura lógica y el análisis de uso de la etapa de requisitos técnicos forman un escenario de implementación, que es la información utilizada en la etapa de diseño de la implementación.

Este capítulo incluye los siguientes apartados:

- [“Información acerca de las arquitecturas lógicas” en la página 51](#)
- [“Diseño de una arquitectura lógica” en la página 53](#)
- [“Ejemplo de arquitecturas lógicas” en la página 60](#)
- [“Zonas de acceso” en la página 69](#)
- [“Escenario de implementación” en la página 70](#)

## Información acerca de las arquitecturas lógicas

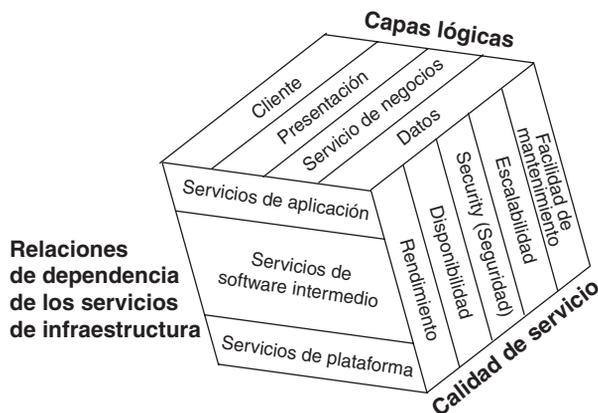
Una arquitectura lógica identifica los componentes de software necesarios para implementar una solución, mostrando las relaciones entre sí. La arquitectura lógica y los requisitos de calidad de servicio procedentes de la etapa de requisitos técnicos forman un escenario de implementación. El escenario de implementación es la base para diseñar la arquitectura de implementación, que se produce en la siguiente fase, el diseño de implementación.

Al desarrollar una arquitectura lógica, deberá identificar no sólo los componentes que proporcionan los servicios a los usuarios, sino también el resto de componentes que proporcionan el software intermedio y los servicios de plataforma. Las relaciones de dependencia de los servicios de infraestructura y las capas lógicas proporcionan dos formas complementarias de realizar este análisis.

Las relaciones de dependencia del servicio de infraestructura y las capas lógicas son dos de las tres dimensiones de la arquitectura de solución en la que está basada Sun Java™ Enterprise System. A continuación se muestran las tres dimensiones y se representan en la [Figura 4-1 en la página 52](#).

- **Relaciones de dependencia de la infraestructura.** Componentes de software que interactúan para proporcionar los servicios empresariales. Los componentes de software requieren un conjunto subyacente de servicios de infraestructura que permiten que los componentes distribuidos se comuniquen entre sí y que interactúen.
- **Capas lógicas.** Una organización lógica de componentes de software en capas que representan la independencia lógica y física de los componentes de software basada en la naturaleza de los servicios que proporcionan.
- **Calidad de servicio.** Las calidades de servicio del sistema, como rendimiento, disponibilidad, escalabilidad, etc, que representan aspectos particulares de un diseño y funcionamiento de una solución de software.

**Figura 4-1** Tres dimensiones de la arquitectura de la solución Java Enterprise System



---

**NOTA** Para obtener más información acerca de los conceptos de arquitectura de Java Enterprise System, consulte el capítulo “Arquitectura de Java Enterprise System” en *Visión general técnica de Java Enterprise System*, <http://docs.sun.com/doc/819-1924>.

---

Una arquitectura lógica muestra los niveles de servicio de la arquitectura mostrando los componentes necesarios y sus relaciones de dependencia. Una arquitectura lógica también distribuye los componentes en capas lógicas que representan los servicios de presentación, empresariales y de datos a los que una capa de cliente puede acceder. Los requisitos de calidad de servicio no se modelan en la arquitectura lógica sino que se emparejan con la arquitectura lógica en un escenario de implementación.

## Diseño de una arquitectura lógica

Cuando se diseña una arquitectura lógica, se utilizan los casos de uso identificados durante la etapa de requisitos técnicos para determinar los componentes de Java Enterprise System que proporcionan los servicios necesarios para la solución. Debe identificar los componentes que proporcionan servicios a los servicios identificados inicialmente.

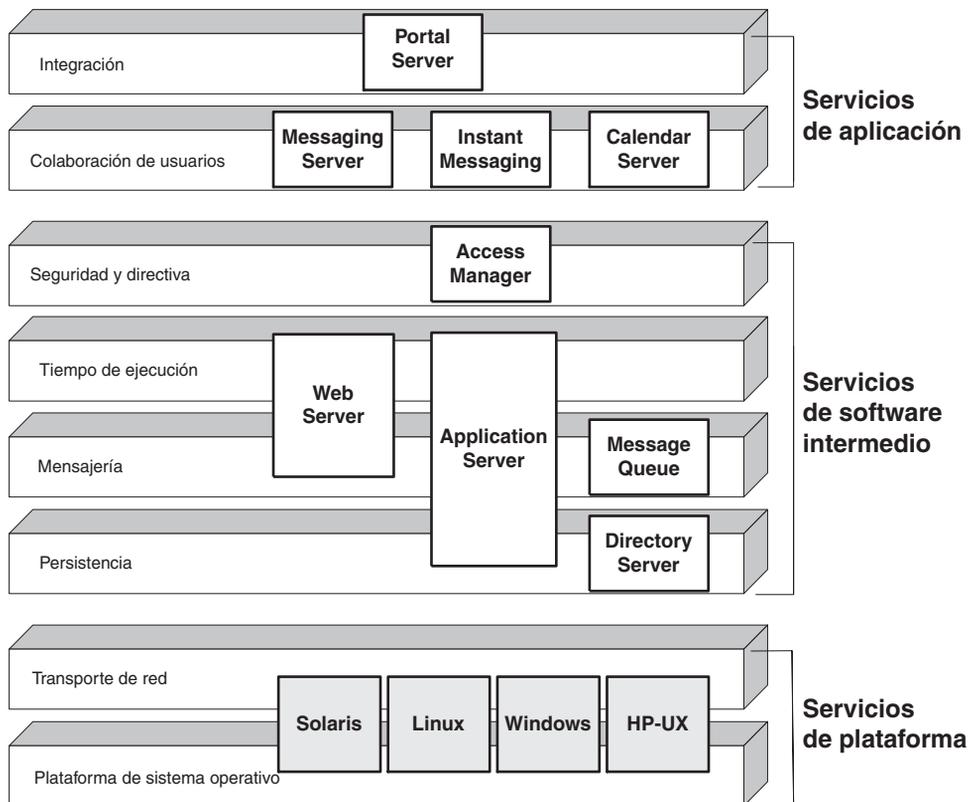
Se colocan los componentes de Java Enterprise System en un contexto de una arquitectura de varias capas de acuerdo con el tipo de servicios que proporcionan. La consideración de los componentes como parte de una arquitectura en varias capas ayuda a determinar posteriormente una estrategia para implementar los requisitos de calidad de servicio (como escalabilidad, disponibilidad, etc.)

Además, puede proporcionar otra visión de los componentes lógicos que los coloca en zonas de acceso seguro. La sección *“Zonas de acceso”* en la página 69 proporciona un ejemplo de las zonas de acceso seguras.

## Componentes de Java Enterprise System

Java Enterprise System consta de componentes de software que interactúan y que proporcionan los servicios empresariales que puede utilizar para crear la solución empresarial. La siguiente figura muestra los principales componentes de software que se suministran con Java Enterprise System. *Visión general técnica de Java Enterprise System*, <http://docs.sun.com/doc/819-1924> proporciona información adicional acerca de los componentes de Java Enterprise System y los servicios que proporcionan.

**Figura 4-2** Componentes de Java Enterprise System



### Relaciones de dependencia de los componentes

Cuando se identifican los componentes de Java Enterprise System para una arquitectura lógica, deberá identificar los componentes compatibles. Por ejemplo, si identifica Messaging Server como un componente necesario para una arquitectura lógica, la arquitectura lógica debe incluir Directory Server y posiblemente Access Manager. Messaging Server depende de Directory Server para los servicios de directorio y Access Manager para las soluciones que requieren un inicio de sesión único.

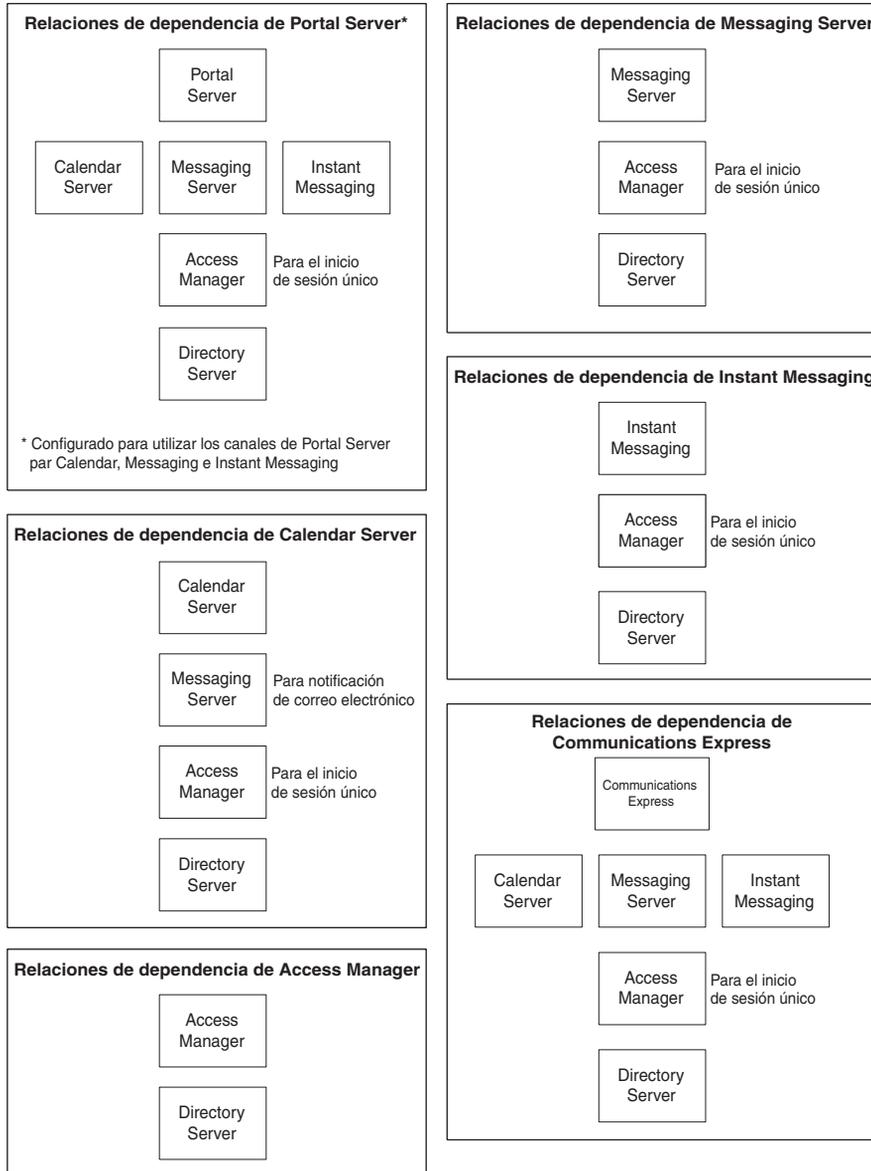
La siguiente tabla muestra las relaciones de dependencia de los componentes de Java Enterprise System. Consulte la [Figura 4-3 en la página 56](#) para ver una representación gráfica de las relaciones de dependencia entre los componentes principales. Al diseñar una arquitectura lógica, use esta tabla y la figura que la acompaña para determinar los componentes dependientes del diseño.

**Tabla 4-1** Java Enterprise System Relaciones de dependencia de los componentes

Componente de Java Enterprise System	Depende de
Application Server	Message Queue Directory Server (opcional)
Calendar Server	Messaging Server (para el servicio de notificación de correo electrónico) Access Manager (para el inicio de sesión único) Web Server (para interfaz Web) Directory Server
Communications Express	Access Manager (para el inicio de sesión único) Calendar Server Messaging Server Instant Messaging Web Server (para interfaz Web) Directory Server
Directory Proxy Server	Directory Server
Directory Server	Ninguno
Access Manager	Application Server o Web Server Directory Server
Instant Messaging	Access Manager (para el inicio de sesión único) Directory Server
Message Queue	Directory Server (opcional)
Messaging Server	Access Manager (para el inicio de sesión único) Web Server (para interfaz Web) Directory Server
Portal Server	Si se configura para usar los canales de Portal Server: Calendar Server Messaging Server Instant Messaging Access Manager (para el inicio de sesión único) Application Server o Web Server Directory Server
Portal Server Secure Remote Access	Portal Server
Web Server	Access Manager (opcional, para control de acceso)

**NOTA** En la [Tabla 4-1](#) no se muestran todas las relaciones de dependencia entre los componentes de Java Enterprise System. La [Tabla 4-1](#) no muestra las relaciones de dependencia que se deben considerar al planificar la instalación. Para obtener una lista completa de las relaciones de dependencia de Java Enterprise System, consulte *Guía de instalación de Java Enterprise System.*, <http://docs.sun.com/doc/819-0806>.

**Figura 4-3** Java Enterprise System Relaciones de dependencia de los componentes



## Compatibilidad con contenedor Web

La sección anterior, “[Relaciones de dependencia de los componentes](#),” no considera el contenedor web en el que se ejecutan Portal Server y Access Manager. Este contenedor web puede obtenerse mediante Application Server, Web Server o un producto de terceros. Cuando diseñe una arquitectura lógica que incluya Portal Server o Access Manager, asegúrese de tener en cuenta el contenedor web necesario para estos componentes.

## Servicios distintos lógicamente prestados por Messaging Server

Messaging Server de Java Enterprise System se puede configurar para proporcionar instancias separadas que ofrecen los siguientes servicios distintos lógicamente:

- Message Transfer Agent
- Message Multiplexor
- Message Express Multiplexor
- Message Store

Estas configuraciones de Messaging Server ofrecen funciones que pueden implementarse en servidores físicos separados y se puede representar en distintas capas de una arquitectura lógica. Dado que estas configuraciones de Messaging Server representan servicios distintos lógicamente en capas separadas, considérelas como componentes lógicamente distintos al diseñar la arquitectura lógica. La sección “[Ejemplo de arquitecturas lógicas](#)” en la página 60 proporciona un ejemplo de los componentes distintos lógicamente.

La siguiente tabla describe las configuraciones distintas lógicamente de Messaging Server.

**Tabla 4-2** Messaging Server Configuraciones

Subcomponente	Descripción
Message Transfer Agent (MTA)	Admite el envío de correo electrónico manejando las conexiones SMTP, encaminando los correos electrónicos y entregando los mensajes a los almacenes de mensajes adecuados. Los componentes de MTA se pueden configurar para que admitan el correo electrónico enviado desde fuera de la empresa (entrada) o enviado desde dentro de la empresa (salida).
Message Store (STR)	Permite la recuperación y almacenamiento de mensajes de correo electrónico.

**Tabla 4-2** Messaging Server Configuraciones (*continúa*)

Subcomponente	Descripción
Message Multiplexor (MMP)	Admite la recuperación de correo electrónico accediendo al almacén de mensaje de clientes de correo electrónico, utilizando los protocolos IMAP o POP.
Messenger Express Multiplexor (MEM)	Admite la recuperación de correo electrónico accediendo al almacén de mensajes en nombre de clientes basados en web (HTTP).

## Componentes de acceso

Java Enterprise System también contiene componentes que permiten acceder a los servicios del sistema, a menudo desde fuera de un servidor de seguridad empresarial. Algunas configuraciones de Messaging Server también proporcionan acceso a la red, como Messaging Server configurado para multiplexor de mensajes. La siguiente tabla describe los componentes de Java Enterprise System que ofrecen acceso remoto a servicios del sistema.

**Tabla 4-3** Java Enterprise System Componentes que ofrecen acceso remoto

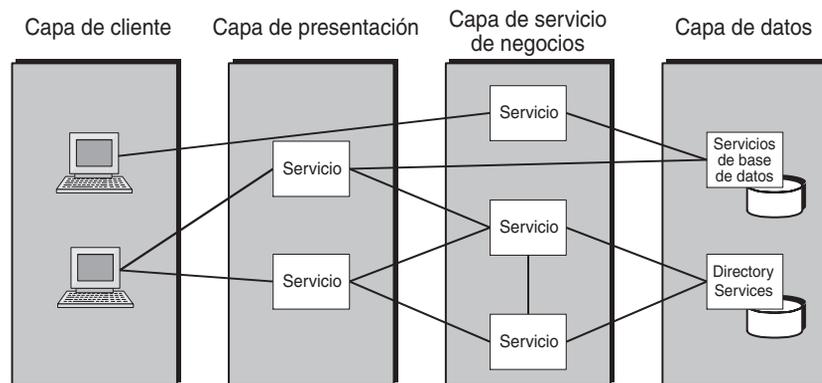
Componente	Descripción
Directory Proxy Server	Incluye mejoras en el control de acceso a directorios, la compatibilidad de esquemas, el enrutamiento y el equilibrado de cargas para varias instancias de Directory Server.
Portal Server, Portal Server Secure Remote Access	Proporciona acceso seguro a Internet desde fuera de un servidor de seguridad empresarial a servicios y contenido de Portal Server, incluidos portales internos y aplicaciones de Internet.
Portal Server, Portal Server Mobile Access	Ofrece acceso inalámbrico desde dispositivos móviles y acceso de voz a Portal Server.
Messaging Server Message Multiplexor (MMP)	Admite la recuperación de correo electrónico accediendo al almacén de mensajes en nombre de clientes basados en web (HTTP).

Los componentes que proporcionan un acceso remoto se despliegan normalmente en zonas de acceso seguro, como se muestra en el ejemplo de la sección [“Zonas de acceso” en la página 69](#).

## Diseño de arquitectura en varias capas

Java Enterprise System es adecuado para el diseño de arquitectura en varias capas, donde los servicios se colocan en capas de acuerdo con las funciones que proporcionan. Cada servicio es independiente lógicamente y se puede acceder mediante servicios que se encuentran en la misma capa o en otra. La siguiente figura muestra un modelo de arquitectura en varias capas para las aplicaciones empresariales, con las capas de clientes, presentación, servicio empresarial y de datos.

**Figura 4-4** Modelo de arquitectura en varias capas



La siguiente tabla describe las capas lógicas que se muestran en la [Figura 4-4](#).

**Tabla 4-4** Capas lógicas en una arquitectura en varias capas

Capa	Descripción
Capa de cliente	Contiene aplicaciones cliente que presentan la información a los usuarios finales. Para Java Enterprise System, estas aplicaciones son normalmente clientes de correo, exploradores de Internet o clientes de acceso móvil.
Capa de presentación	Ofrece servicios que muestran los datos a los usuarios finales, permitiendo que los usuarios procesen y manipulen su presentación. Por ejemplo, un cliente de correo web o un componente de Portal Server permite que los usuarios modifiquen la presentación de la información que reciben.

**Tabla 4-4** Capas lógicas en una arquitectura en varias capas (*continúa*)

Capa	Descripción
Capa de servicios de negocios	Proporciona los servicios de servidor que normalmente recuperan los datos de la capa de datos para ofrecer otros servicios dentro de las capas de presentación o servicios de negocios o directamente a los clientes en la capa de clientes. Por ejemplo, Access Manager proporciona servicios de identidad a otros componentes de Java Enterprise System.
Capa de datos	Proporciona servicios de bases de datos a los que los servicios de las capas de presentación o de servicios de negocios pueden acceder. Por ejemplo, Directory Server proporciona acceso al directorio LDAP a otros servicios.

El diseño de arquitectura en varias capas proporciona varias ventajas. Durante la etapa de diseño de implementación, la colocación de los servicios de acuerdo con sus funciones en una arquitectura de varias capas ayuda a determinar cómo distribuir los servicios en la red. También puede ver cómo los componentes de la arquitectura acceden a los servicios de otros componentes. Esta visualización ayuda a planificar la disponibilidad, escalabilidad, seguridad y otras soluciones de calidad de servicio.

## Ejemplo de arquitecturas lógicas

Esta sección proporciona algunos ejemplos de arquitecturas lógicas para las soluciones de Java Enterprise System. Estos ejemplos muestran cómo se colocan los componentes lógicos en las capas adecuadas de una arquitectura en varias capas y luego, analizan las relaciones entre los componentes estudiando los casos de uso. Use los ejemplos de arquitecturas lógicas de esta sección como la base para comprender el diseño de arquitectura lógica en las soluciones de Java Enterprise System.

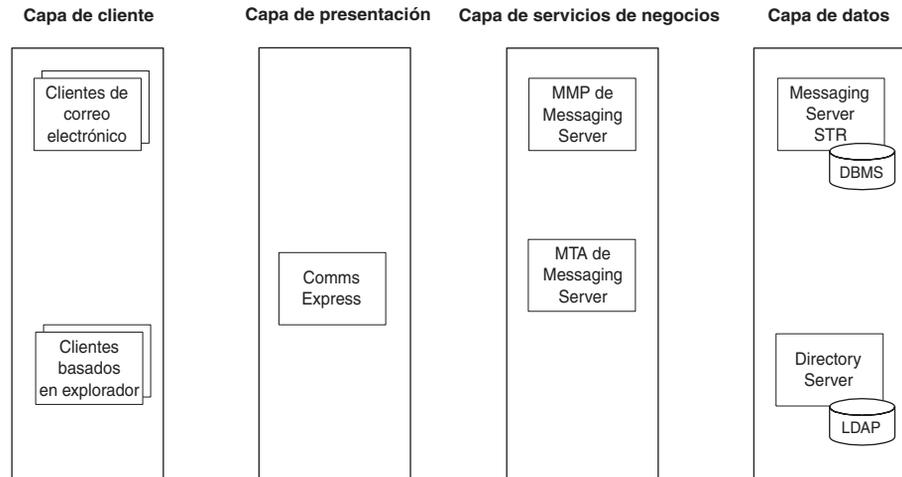
El primer ejemplo es una solución de Messaging Server básica que muestra cómo los componentes lógicamente distintos de Messaging Server interactúan con otros componentes. El segundo ejemplo muestra una arquitectura lógica para una solución de implementación basada en identidades que puede ser adecuada para una empresa de tamaño medio de entre 1.000 y 5.000 empleados.

## Ejemplo de Messaging Server:

La siguiente figura muestra una arquitectura lógica básica para una implementación de Messaging Server. Esta arquitectura lógica muestra únicamente los componentes lógicamente distintos necesarios para Messaging Server. Las figuras posteriores ilustran las relaciones entre estos componentes.

**NOTA** Normalmente, una implementación de Messaging Server es parte de una solución empresarial que incluye otros componentes de Java Enterprise System, como se muestra en [“Ejemplo de comunicaciones basadas en identidades”](#) en la página 65.

**Figura 4-5** Arquitectura lógica de una implementación de Messaging Server



La tabla que figura a continuación describe los componentes que se muestran en la [Figura 4-5](#).

**Tabla 4-5** Componentes en la arquitectura lógica de Messaging Server

Componente	Descripción
Cientes de correo electrónico	Aplicaciones cliente para leer y enviar correo electrónico.
Messaging Server MTA	Messaging Server configurado como agente de transferencia de mensajes (MTA) para recibir, enrutar, transportar y entregar mensajes de correo electrónico.

**Tabla 4-5** Componentes en la arquitectura lógica de Messaging Server *(continúa)*

Componente	Descripción
Messaging Server MMP	Messaging Server configurado como un multiplexor de mensajes (MMP) para enrutar las conexiones a los almacenes de mensajes adecuados para su recuperación y almacenamiento. MMP accede a Directory Server para buscar información de directorios para determinar el almacén de mensajes adecuado.
Messaging Server STR	Messaging Server configurado como un almacén de mensajes para la recuperación y almacenamiento de mensajes de correo electrónico.
Directory Server	Proporciona acceso a los datos del directorio LDAP.

La arquitectura lógica no especifica la duplicación de los servicios para los componentes de Messaging Server. Por ejemplo, normalmente las implementaciones empresariales crean instancias MTA de entrada y salida separadas pero la [Figura 4-5 en la página 61](#) muestra únicamente un componente MTA. La duplicación de los componentes lógicos en varias instancias es una decisión de diseño que se realiza durante la fase de diseño de implementación.

## Casos de uso de Messaging Server

Los casos de uso ayudan a identificar las relaciones entre los componentes lógicos de una arquitectura. Al asignar las interacciones entre los componentes según los casos de uso, se obtiene una imagen de la interacción entre los componentes que es útil en el diseño de la implementación.

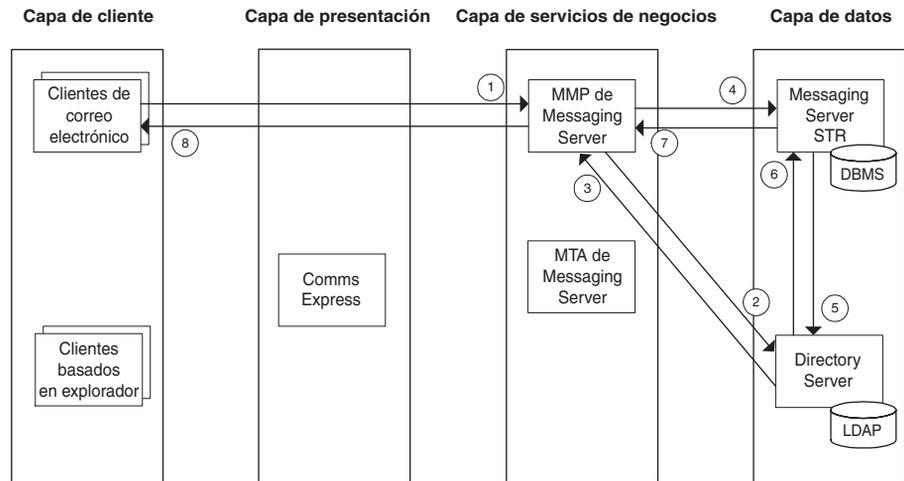
Normalmente, se analiza el caso de uso para determinar la interacción de los componentes antes de un diseño de implementación. Los siguientes tres casos de uso son habituales para Messaging Server y muestran las interacciones entre los componentes lógicos.

### *Caso de uso 1: El usuario inicia sesión con éxito en Messaging Server*

1. El cliente de correo electrónico envía la información de inicio de sesión a Messaging Server Multiplexor (MMP)
2. MMP solicita una verificación del ID de usuario y la contraseña a Directory Server.
3. Directory Server devuelve la verificación a MMP.
4. MMP solicita la lista de mensajes a Messaging Server Message Store (STR).
5. STR solicita el registro LDAP del usuario a Directory Server.

6. Directory Server envía el registro LDAP del usuario a STR.
7. STR envía la lista de mensajes a MMP.
8. MMP reenvía la lista de mensajes al cliente de correo electrónico.

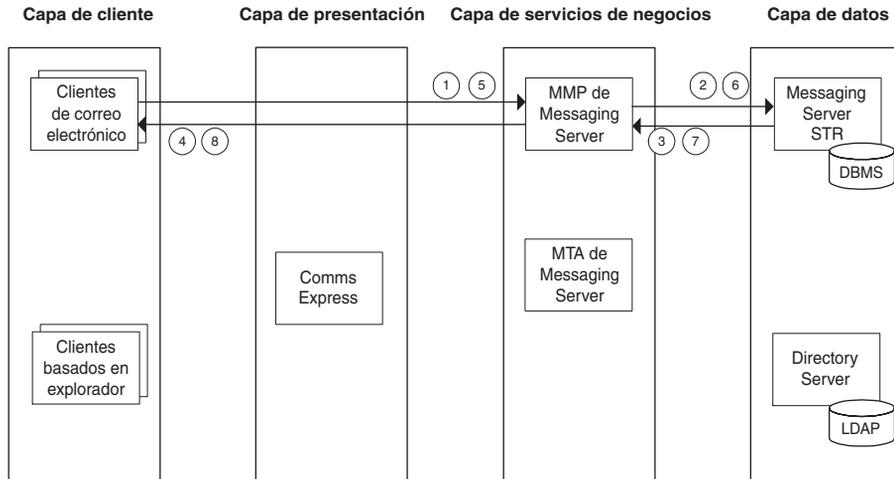
**Figura 4-6** Arquitectura lógica de Messaging Server que muestra el caso de uso 1



*Caso de uso 2: El usuario que ha iniciado la sesión lee y elimina correo*

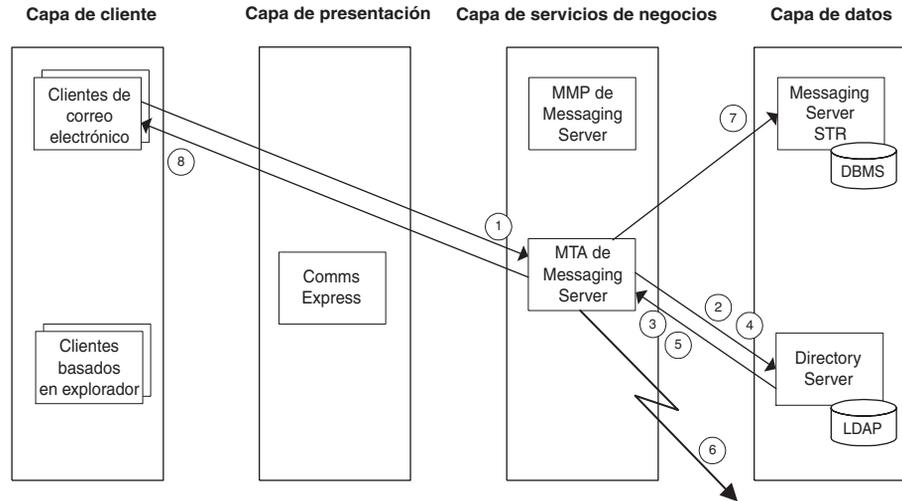
1. El cliente de correo electrónico solicita leer un mensaje desde Messaging Server Multiplexor (MMP).
2. MMP solicita un mensaje a Messaging Server Message Store (STR).
3. STR envía el mensaje a MMP.
4. MMP reenvía el mensaje al cliente de correo electrónico.
5. El cliente de correo electrónico envía la acción de eliminación de mensaje a MMP.
6. MMP envía la acción de eliminación de mensaje a STR.
7. STR elimina el mensaje de la base de datos y envía la confirmación a MMP.
8. MMP reenvía la confirmación de eliminación al cliente de correo electrónico.

**Figura 4-7** Arquitectura lógica de Messaging Server que muestra el caso de uso 2



**Caso de uso 3: El usuario que ha iniciado sesión envía el mensaje de correo electrónico**

1. El cliente de correo electrónico envía el mensaje creado en el cliente a Messaging Server Message Transfer Agent (MTA).
2. MTA solicita una verificación del ID de usuario y la contraseña a Directory Server.
3. Directory Server devuelve la verificación a MTA.
4. MTA comprueba el dominio de destino de cada destinatario en Directory Server.
5. Directory Server envía a MTA el dominio de destino de cada destinatario.
6. MTA envía el mensaje a cada destinatario.
7. MTA envía el mensaje a Messaging Server Message Store (STR) para guardar el mensaje en el buzón de salida.
8. MTA envía la confirmación al cliente de correo electrónico.

**Figura 4-8** Arquitectura lógica de Messaging Server que muestra el caso de uso 3

## Ejemplo de comunicaciones basadas en identidades

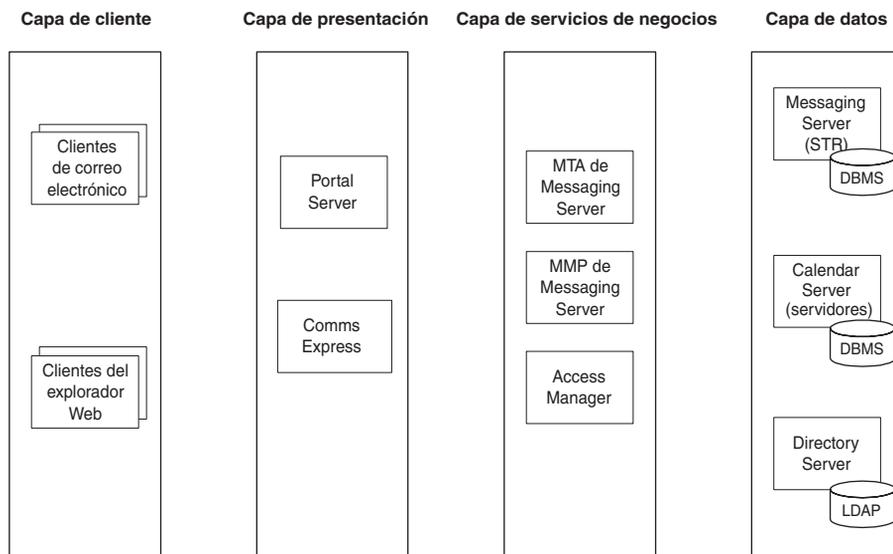
Este ejemplo ilustra una solución de comunicaciones basadas en identidades para una empresa de tamaño medio de entre 1.000 y 5.000 empleados. Normalmente, es necesario un análisis empresarial exhaustivo junto con unos requisitos técnicos detallados para diseñar la arquitectura lógica. Sin embargo, debido a que esto es un ejemplo teórico, se asume que se han determinado los siguientes requisitos empresariales:

- Los empleados de la empresa requieren acceso personalizado a los sitios web internos, a los servicios de comunicaciones, de calendario y a otros recursos.
- La autenticación y autorización en la empresa permiten acceder a los sitios web internos y otros servicios.
- Una única identidad se controla en todos los servicios empresariales, lo que permite un único inicio de sesión (SSO) que permita acceder a los sitios web internos y otros servicios.

Los casos de uso para este ejemplo detallarían los procedimientos de inicio de sesión, de lectura y envío de correo electrónico, de personalización del portal, la sincronización de los calendarios y otras actividades de usuario parecidas.

La siguiente figura muestra una arquitectura lógica para este tipo de solución de comunicaciones basadas en identidades.

**Figura 4-9** Arquitectura lógica para un escenario de comunicaciones basado en identidades



## Ejemplo de casos de uso para comunicaciones basadas en identidades

Para una solución de implementación de esta naturaleza, normalmente se cuenta con numerosos casos de uso que definen la interacción de los usuarios con los servicios que la solución ofrece. Este ejemplo se centra en la interacción entre los componentes cuando un usuario se registra en un portal desde un cliente de explorador web. El ejemplo divide este escenario de inicio de sesión en dos casos de uso:

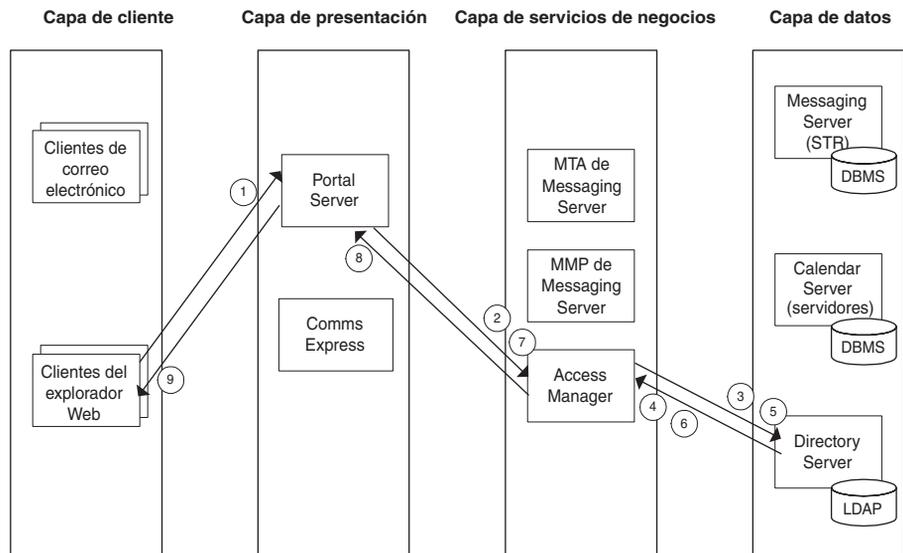
- El usuario inicia sesión, se autentica y Portal Server recupera la configuración del portal del usuario.
- Portal Server recupera el correo electrónico y la información de calendario que se mostrará en el cliente web.

Los dos casos se pueden considerar como un caso de uso extendido. Sin embargo, para este ejemplo los casos de uso se muestran separados para una mayor claridad.

**Caso de uso 1: El usuario inicia sesión con éxito y Portal recupera la configuración del usuario**

1. El cliente del explorador Web envía el ID de usuario y la contraseña a Portal Server.
2. Portal Server solicita la autenticación a Access Manager.
3. Access Manager solicita una verificación del ID de usuario y la contraseña a Directory Server.
4. Directory Server comprueba el ID de usuario y la contraseña.
5. Access Manager solicita el perfil del usuario a Directory Server.
6. Directory Server envía el perfil del usuario.
7. Portal Server solicita el perfil de visualización del usuario a Access Manager.
8. Access Manager envía la configuración de portal.
9. La configuración de Portal se muestra en un cliente explorador web.

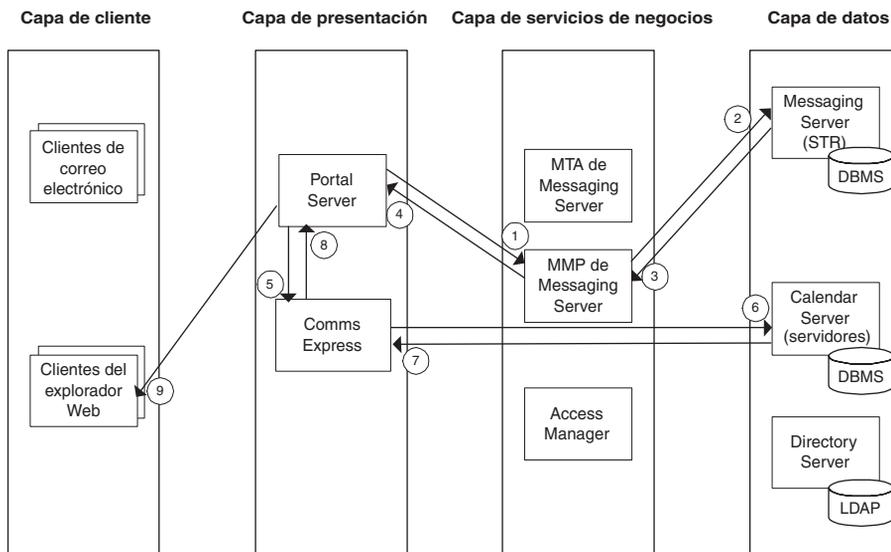
**Figura 4-10** Arquitectura lógica de un escenario de comunicaciones que muestra el caso de uso 1



**Caso de uso 2: Portal Server muestra la información de calendario y de correo electrónico**

1. Después de iniciar sesión con éxito, la autenticación y recuperar la configuración de portal, Portal Server solicita los mensajes de correo electrónico a Messaging Server MMP.
2. MMP solicita la lista de mensajes a Messaging Server STR.
3. STR envía la lista de mensajes a MMP.
4. MMP envía los encabezados del mensaje a Portal Server.
5. Portal Server solicita la información de calendario a Communications Express.
6. Communications Express solicita la información de calendario al servidor de Calendar Server.
7. El servidor de Calendar Server solicita la información de calendario a Communications Express.
8. Communications Express envía la información de calendario a Portal Server.
9. Portal Server envía toda la información del canal al cliente del explorador web.

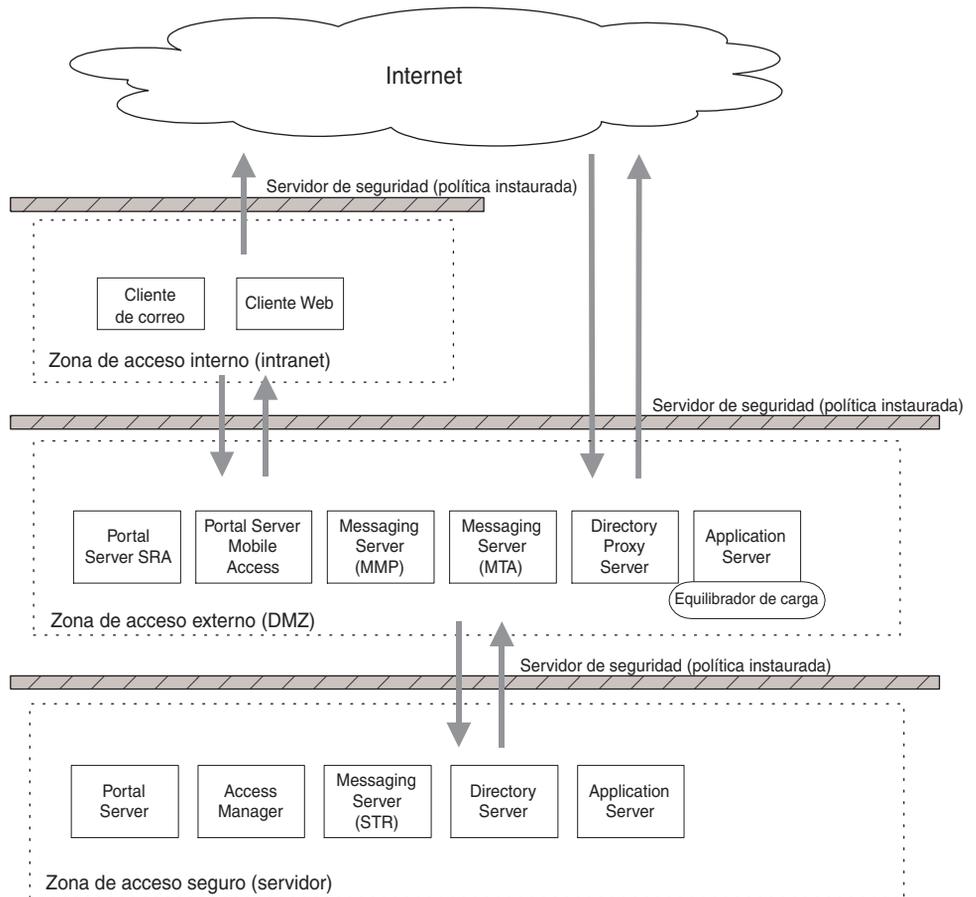
**Figura 4-11** Arquitectura lógica de un escenario de comunicaciones que muestra el caso de uso 2



# Zonas de acceso

Otra manera de representar los componentes de una arquitectura lógica es colocarlos en zonas de acceso que muestran cómo la arquitectura proporciona el acceso seguro. La siguiente figura muestra las zonas de acceso para implementar los componentes de Java Enterprise System. Cada zona de acceso muestra cómo los componentes ofrecen acceso remoto seguro desde y a Internet e intranet.

**Figura 4-12** Componentes lógicos colocados en zonas de acceso



La tabla que figura a continuación describe las zonas de acceso que se muestran en la [Figura 4-12](#).

**Tabla 4-6** Zonas de acceso seguro y componentes en ellas

Zona de acceso	Descripción
Zona de acceso interno (Intranet)	Acceso a Internet mediante políticas puestas en vigor gracias a un servidor de seguridad entre la intranet e Internet. Normalmente, los usuarios finales utilizan la zona de acceso interno para navegar en la web y para enviar correos electrónicos.  En algunos casos, se permite el acceso directo para navegar por Internet. Sin embargo, normalmente se proporciona un acceso seguro a y desde Internet mediante la zona de acceso externo.
Zona de acceso externo (DMZ)	Proporciona un acceso seguro a y desde Internet, actuando como un búfer de seguridad a servicios de servidor vitales.
Zona de acceso seguro (servidor)	Ofrece un acceso restringido a servicios de servidor vitales, a los que sólo se puede acceder desde la zona de acceso externo.

La [Figura 4-12](#) no ilustra las capas lógicas indicadas en los ejemplos anteriores, sino que se centra en los componentes que proporcionan acceso interno y remoto, las relaciones de estos componentes con las medidas de seguridad como servidores de seguridad y muestra gráficamente las normas de acceso que se deben implantar. Utilice el diseño de arquitectura de varias capas junto con el diseño que muestra las zonas de acceso para proporcionar un modelo lógico de la implementación planificada.

## Escenario de implementación

El diseño de arquitectura lógica completado no es suficiente por sí mismo para pasar a la etapa de diseño de implementación del ciclo de vida de la solución. Deberá comparar la arquitectura lógica con los requisitos de calidad de servicio (QoS) determinados durante la etapa de requisitos técnicos. Esta comparación permite crear un escenario de implementación. El escenario de implementación es el punto de inicio para diseñar la arquitectura de implementación, como se explica en el [Capítulo 5, “Diseño de implementación”](#).

# Diseño de implementación

Durante la etapa de diseño de implementación del ciclo de vida de la solución, se diseña una arquitectura de implementación general y una especificación de implementación detallada y se prepara una serie de planes y especificaciones necesarias para implementar la solución. La aprobación del proyecto se produce en la fase de diseño de implementación.

Este capítulo incluye los siguientes apartados:

- “Información acerca del diseño de implementación” en la página 72
- “Metodología del diseño de implementación” en la página 76
- “Cálculo de los requisitos de procesadores” en la página 77
- “Cálculo de los requisitos de procesadores para transacciones seguras” en la página 84
- “Determinación de las estrategias de disponibilidad” en la página 88
- “Determinación de estrategias para la escalabilidad” en la página 97
- “Diseño para la óptima utilización de los recursos” en la página 102
- “Ejemplo de arquitectura de implementación” en la página 104

# Información acerca del diseño de implementación

El diseño de implementación se inicia con el escenario de implementación que se crea durante las etapas de diseño lógico y requisitos técnicos del ciclo de vida de la solución. El escenario de implementación contiene una arquitectura lógica y los requisitos de calidad del servicio (QoS) de la solución. Los componentes identificados en la arquitectura lógica se asignan en servidores físicos y otros dispositivos de red para crear una arquitectura de implementación. Los requisitos de QoS proporcionan directrices sobre las configuraciones de hardware para obtener especificaciones relativas a la calidad del servicio como el rendimiento, la disponibilidad o la escalabilidad.

El diseño de la arquitectura de implementación es un proceso iterativo. Normalmente, se revisan los requisitos de QoS y se vuelven a examinar los diseños preliminares. Se tienen en cuenta las relaciones entre los distintos requisitos de calidad de servicio, equilibrando los cambios y el coste de propiedad para obtener una solución óptima que en última instancia satisfaga los objetivos del proyecto.

## Aprobación del proyecto

La aprobación del proyecto se produce en la etapa de diseño del mismo, normalmente después de haber creado la arquitectura de implementación. Con la arquitectura de implementación y las especificaciones que se describen a continuación, se calcula el coste real de la implementación y se envía a los participantes para su aprobación. Una vez que se aprueba el proyecto, se firman los contratos para la realización de la implementación y se asignan los recursos necesarios para implementar el proyecto.

## Resultados del diseño de implementación

Durante la etapa de diseño de implementación, deberá preparar alguno de los siguientes planes y especificaciones:

- **Arquitectura de implementación.** Una arquitectura general que muestra la asignación de una arquitectura lógica en un entorno físico. El entorno físico incluye los nodos informáticos de un entorno de intranet o Internet, procesadores, memoria, dispositivos de almacenamiento y otros dispositivos de red y hardware.
- **Especificaciones de implementación.** Especificaciones detalladas que se utilizan como referencia para crear la implementación. Estas especificaciones proporcionan detalles específicos acerca del hardware de red e informático para obtener y describir el diseño de la red para la implementación. Las especificaciones de implementación también incluyen especificaciones para los servicios de directorios, incluidos los detalles sobre el árbol de información de directorios (DIT) y los grupos y funciones definidos para el acceso a los directorios.
- **Planes de implementación.** Un grupo de planes que cubre distintos aspectos de la implementación de una solución de software empresarial. Los planes de implementación incluyen:
  - **Plan de migración.** Describe las estrategias para migrar datos de la empresa y actualizar el software empresarial. Los datos migrados deben ser compatibles con los formatos y estándares de las aplicaciones empresariales recién instaladas. Todo el software empresarial debe encontrarse en los niveles de versión correctos para que se pueda relacionar entre sí.
  - **Plan de instalación.** Obtenido a partir de la arquitectura de implementación, especifica los nombres del servidor de hardware, los directorios de instalación, la secuencia de instalación, los tipos de instalación para cada nodo y la información de configuración necesaria para instalar y configurar una implementación distribuida.
  - **Plan de gestión de usuarios.** Incluye estrategias de migración para los datos en los directorios y bases de datos existentes, especificaciones de diseño de directorio que tienen en cuenta el diseño de duplicación especificado en la arquitectura de implementación y procedimientos para proporcionar nuevo contenido a los directorios.

- **Plan de prueba.** Describe los procedimientos para probar el software implementado, incluidos planes específicos para desarrollar implementaciones prototipo y piloto, pruebas de estrés que determinan la capacidad para tratar las cargas previstas y pruebas funcionales que determinan si la función planificada actúa de la forma prevista.
- **Plan de despliegue.** Describe los procedimientos y programación para cambiar la implementación de un entorno de planificación y prueba a uno de producción. Normalmente, el cambio de la implementación a la producción se produce en varias etapas. Por ejemplo, la primera etapa puede ser la implementación del software para un grupo limitado de usuarios y aumentar la base de usuarios con cada etapa hasta que se complete la implementación. La implementación en fases también puede incluir una implementación programada de paquetes de software específicos hasta que se complete toda la implementación.
- **Plan de recuperación tras desastre.** Describe los procedimientos que se deben llevar a cabo para restaurar el sistema cuando se producen fallos imprevistos en el mismo. El plan de recuperación incluye procedimientos para los fallos a pequeña y gran escala.
- **Plan de operaciones (Manual de ejecución).** Un manual de operaciones que describe los procedimientos de supervisión, mantenimiento, instalación y actualización.
- **Plan de formación.** Contiene procesos y procedimientos para formar a los operadores, administradores y usuarios finales acerca del software empresarial recién instalado.

## Factores que afectan al diseño de implementación

Varios factores influyen en la toma de decisiones durante el diseño de la implementación. Tenga en cuenta los siguientes factores clave:

- **Arquitectura lógica.** La arquitectura lógica detalla los servicios funcionales en una solución propuesta y las relaciones entre los componentes que proporcionan dichos servicios. Utilice la arquitectura lógica como el factor clave para determinar la mejor manera de distribuir los servicios. Un escenario de implementación contiene la arquitectura lógica junto con los requisitos de calidad de servicio (se describe a continuación).
- **Requisitos de calidad de servicio.** Los requisitos de calidad de servicio (QoS) especifican varios aspectos del funcionamiento de una solución. Utilice los requisitos de calidad de servicio para ayudar a desarrollar estrategias que permitan obtener los objetivos de rendimiento, disponibilidad, escalabilidad, facilidad de mantenimiento, etc. definidos.. Un escenario de implementación contiene la arquitectura lógica junto con los requisitos de calidad de servicio (como se ha descrito anteriormente).
- **Análisis de uso.** El análisis de uso, desarrollado durante la etapa de requisitos técnicos del ciclo de vida de la solución, proporciona información acerca de los patrones de uso que pueden ayudar a calcular la carga y el estrés de un sistema implementado. Utilice el análisis de uso para ayudar a aislar los atascos de rendimiento y desarrollar estrategias para satisfacer los requisitos de QoS.
- **Casos de uso.** Los casos de uso, desarrollados durante la etapa de requisitos técnicos del ciclo de vida de la solución, muestran varias interacciones de usuario identificadas para una implementación, a menudo identificando los casos de uso más comunes. Aunque los casos de uso están englobados en el análisis de uso, al evaluar un diseño de implementación se debe hacer referencia a los casos de uso para asegurar que se tratan adecuadamente.
- **Acuerdos de nivel de servicio.** Un acuerdo de nivel de servicio (SLA) especifica los requisitos de rendimiento mínimos y, cuando no se cumplen dichos niveles, el nivel y alcance del servicio técnico al cliente que se debe prestar. Un diseño de implementación debería satisfacer fácilmente los requisitos de rendimiento especificados en un acuerdo de nivel de servicio.

- **Coste total de propiedad.** En el diseño de implementación, se analizan las soluciones potenciales que se encargan de los requisitos de QoS para la disponibilidad, rendimiento, escalabilidad, etc. Sin embargo, para cada solución que se considera, también se debe tener en cuenta el coste de dicha solución y cómo repercute en el coste total de propiedad. Asegúrese de que tiene presente los cambios asociados con las decisiones y que optimiza los recursos para obtener los requisitos empresariales dentro de las limitaciones existentes.
- **Objetivos empresariales.** Los objetivos empresariales se establecen durante la etapa de análisis empresarial del ciclo de vida de la solución e incluyen los requisitos y limitaciones empresariales para satisfacer dichos objetivos. En última instancia, el diseño de implementación se juzga por su capacidad para cumplir los objetivos empresariales.

## Metodología del diseño de implementación

Al igual que otros aspectos de la planificación de la implementación, el diseño es tanto un arte como una ciencia y no se puede detallar con procedimientos y procesos específicos. Algunos de los factores que contribuyen a un diseño de implementación realizado con éxito son la experiencia previa en diseño, el conocimiento de la arquitectura de sistemas, el conocimiento del campo y el pensamiento creativo aplicado.

Normalmente, el diseño de implementación se centra en la obtención de los requisitos de rendimiento a la vez que se cumplen los requisitos de calidad de servicio. Las estrategias que se utilicen deben equilibrar los cambios implicados con las decisiones del diseño para optimizar la solución. Normalmente, la metodología que utilice implica las siguientes tareas:

- **Cálculo de los requisitos del procesador.** A menudo, el diseño de implementación comienza con el cálculo del número de CPU necesarias para cada componente de la arquitectura lógica. Comience con los casos de uso que representen la mayor carga y continúe con cada caso de uso. Tenga en cuenta la carga en todos los componentes que forman los casos de uso y modifique los cálculos consecuentemente. Considere también la experiencia previa que tenga con el diseño de sistemas empresariales.
- **Cálculo de los requisitos de procesadores para un transporte seguro.** Estudie los casos de uso que requieren un transporte seguro y modifique los cálculos de CPU de manera correspondiente.

- **Duplicación de los servicios para disponibilidad y escalabilidad.** Una vez que haya realizado los cálculos de procesadores, cambie el diseño para tener en cuenta los requisitos de calidad de servicio en cuanto a disponibilidad y escalabilidad. Considere soluciones de equilibrado de carga que traten las cuestiones de disponibilidad y recuperación tras error.

En el análisis, considere los cambios derivados de las decisiones del diseño. Por ejemplo, ¿cuáles son los efectos de la estrategia de disponibilidad y escalabilidad sobre la facilidad de mantenimiento del sistema? ¿Cuáles son los otros costes de las estrategias?

- **Identificación de cuellos de botella.** A medida que continúa con su análisis, examine el diseño de implementación para identificar cualquier cuello de botella que produzca que la transmisión de los datos esté por debajo de los requisitos y realice los ajustes pertinentes.
- **Optimización de los recursos.** Revise el diseño de implementación en cuanto a gestión de los recursos y considere las opciones que minimizan los costes a la vez que satisfacen los requisitos.
- **Gestión de los riesgos.** Vuelva a estudiar los análisis técnicos y empresariales con respecto al diseño, realizando las modificaciones necesarias para tener en cuenta los eventos o situaciones que no se hubieran previsto en la planificación anterior.

## Cálculo de los requisitos de procesadores

Esta sección plantea un proceso para estimar el número de CPU y la memoria correspondiente que son necesarios para poder prestar los servicios en un diseño de implementación. La sección incluye un ensayo de un proceso de cálculo para un ejemplo de escenario de implementación de comunicaciones.

La estimación de la potencia de cálculo de la CPU es un proceso iterativo que considera las siguientes opciones:

- Componentes lógicos y sus interacciones (como se indica en las relaciones de dependencia entre los componentes en la arquitectura lógica)
- Análisis de uso para los casos de uso identificados
- Requisitos de calidad de servicio
- Experiencia anterior con diseño de implementación y con Java Enterprise System
- Consulta con los servicios profesionales de Sun que tienen experiencia en el diseño e implementación de distintos tipos de escenarios de implementación

El proceso de cálculo incluye los siguientes pasos. El orden de estos pasos no es vital, pero proporciona una forma de considerar los factores que afectan al resultado final.

1. Determine un cálculo base de CPU para los componentes identificados como puntos de entrada del usuario al sistema.

Una decisión de diseño es si se deben cargar total o parcialmente las CPU. Las CPU completamente cargadas maximizan la capacidad del sistema. Para aumentar la capacidad, se aumentarán los costes de mantenimiento y el posible tiempo de inactividad al añadir CPU adicionales. En algunos casos, puede elegir añadir equipos adicionales para satisfacer el aumento de los requisitos de rendimiento.

Las CPU cargadas parcialmente permiten tratar el aumento de requisitos de rendimiento sin tener que incurrir inmediatamente en costes de mantenimiento. Sin embargo, hay un gasto adicional asociado a los sistemas infrautilizados.

2. Ajuste los cálculos de CPU para tener en cuenta las interacciones entre los componentes.

Estudie las interacciones entre los componentes en la arquitectura lógica para determinar la carga extra necesaria debido a los componentes dependientes.

3. Estudie el análisis de uso de los casos de uso específicos para determinar las cargas máximas del sistema, y realice los ajustes necesarios en los componentes que tratan estas cargas máximas.

Comience con los casos de uso que más ponderan (aquellos que requieren la mayor carga) y continúe con cada caso de uso para asegurarse de que tiene en cuenta todos los escenarios de uso proyectados.

4. Ajuste los cálculos de CPU para que reflejen los requisitos de seguridad, disponibilidad y escalabilidad.

Este proceso de cálculo proporciona puntos de inicio para determinar la potencia de procesamiento real que necesita. Normalmente, se crean implementaciones prototipo basadas en dichos cálculos y luego se realizan comprobaciones estrictas con respecto a los casos de uso previstos. Sólo después de comprobaciones iterativas puede determinar los requisitos de procesamiento reales para un diseño de implementación.

## Ejemplo de cálculo de los requisitos de procesadores

Esta sección ilustra una metodología para estimar la potencia de procesamiento necesaria para una implementación de ejemplo. La implementación de ejemplo está basada en la arquitectura lógica para la solución de comunicaciones basada en identidades para una empresa de tamaño medio de entre 1.000 y 5.000 empleados, como se describe en la sección [“Ejemplo de comunicaciones basadas en identidades” en la página 65](#).

Las cifras de CPU y memoria que se utilizan en el ejemplo son cálculos arbitrarios utilizados sólo a modo de referencia. Estas cifras están basadas en datos arbitrarios en los que se basa el ejemplo teórico. Es necesario un análisis exhaustivo de los distintos factores para calcular los requisitos de procesadores. Este análisis debería incluir, aunque no debería limitarse a ello, la siguiente información:

- Casos de uso detallados y análisis de uso basados en un análisis exhaustivo de negocio
- Requisitos de calidad de servicio determinados por el análisis de los requisitos empresariales
- Costes y especificaciones específicos del hardware de procesamiento y de red
- Experiencia previa en implementaciones de despliegues parecidos

---

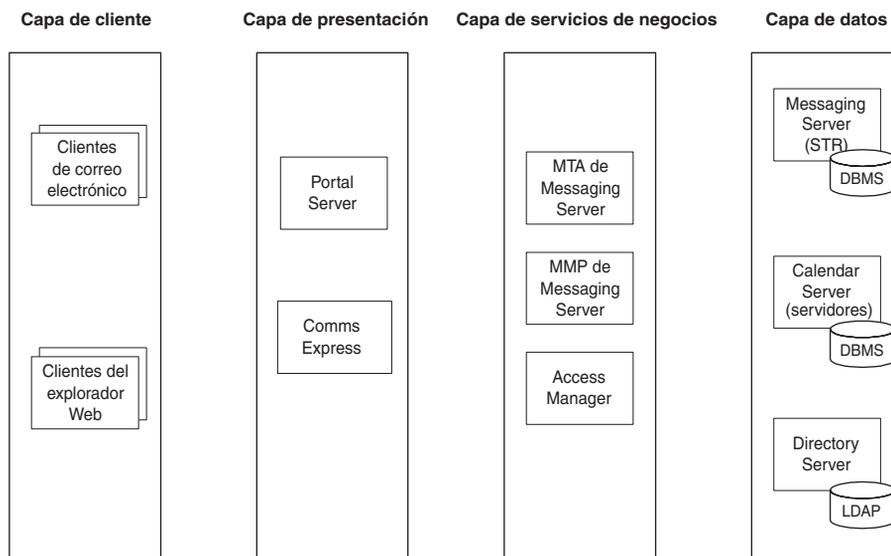
**PRECAUCIÓN** La información que se presenta en estos ejemplos no representa ningún consejo de implementación y sólo se deberá considerar como un ejemplo de un proceso que puede utilizarse al diseñar un sistema.

---

### Determine el cálculo base de CPU para los puntos de entrada del usuario

Comience calculando el número de CPU necesarias para tratar la carga prevista de cada componente que actúa como punto de entrada de usuario. La siguiente figura muestra la arquitectura lógica para un escenario de comunicación basado en identidades descrito anteriormente en el [Capítulo 4, “Diseño lógico” en la página 65](#).

**Figura 5-1** Arquitectura lógica para un escenario de comunicaciones basado en identidades



La siguiente tabla muestra los componentes en la capa de presentación de la arquitectura lógica que trabaja directamente con los usuarios finales de la implementación. La tabla incluye cálculos de CPU base obtenidos a partir del análisis de requisitos técnicos, casos de uso, análisis de uso específico y experiencia previa con este tipo de implementación.

**Tabla 5-1** Cálculos de CPU para componentes que contienen acceso a los puntos de entrada de usuario

Componente	Número de CPU	Descripción
Portal Server	4	Componente que es un punto de entrada de usuario.
Communications Express	2	Dirige los datos a los canales de calendario y mensajería de Portal Server.

## Incluya los cálculos de CPU para las relaciones de dependencia de los servicios

Los componentes que proporcionan puntos de entrada de usuario requieren compatibilidad con otros componentes de Java Enterprise System. Para continuar especificando requisitos de rendimiento, incluya los cálculos de rendimiento para que tengan en cuenta la compatibilidad requerida por otros componentes. El tipo de interacciones entre los componentes debería detallarse al diseñar la arquitectura lógica, tal y como se describe en los ejemplos de arquitectura lógica de la sección “Ejemplo de arquitecturas lógicas” en la página 60.

**Tabla 5-2** Cálculos de CPU para compatibilidad de componentes

Componente	CPU	Descripción
Messaging Server MTA (entrada)	1	Dirige los mensajes de correo entrantes de clientes de Communications Express y de correo electrónico.
Messaging Server MTA (salida)	1	Dirige los mensajes de correo salientes a los destinatarios.
Messaging Server MMP	1	Acceso al almacén de mensajes de Messaging Server para clientes de correo electrónico.
Messaging Server STR (almacén de mensajes)	1	Recupera y almacena mensajes de correo electrónico.
Access Manager	2	Proporciona servicios de autorización y de autenticación.
Calendar Server (servidor)	2	Recupera y guarda datos de calendario para Communications Express, aplicación de usuario de Calendar Server.
Directory Server	2	Proporciona servicios de directorio LDAP.
Web Server	0	Proporciona compatibilidad con contenedor web para Portal Server y Access Manager. (No se precisa de ciclos de CPU adicionales.)

## Estudie los casos de uso para la utilización de carga máxima

Vuelva a los casos de uso y análisis de uso para identificar las áreas de utilización de carga máxima y realice los ajustes oportunos a los cálculos de CPU.

Por ejemplo, suponga que para este ejemplo identifica las siguientes condiciones de carga máxima:

- Aumento inicial de usuarios a medida que se registran simultáneamente
- Intercambios de correo electrónico durante franjas horarias específicas

Para tener en cuenta esta utilización de carga máxima, realice ajustes en los componentes que prestan estos servicios. La siguiente tabla describe los ajustes que puede tener en cuenta a la hora de evaluar esta utilización de carga máxima.

**Tabla 5-3** Ajustes en el cálculo de CPU para carga máxima

Componente	CPU (ajustadas)	Descripción
Messaging Server MTA entrada	2	Añada 1 CPU por correo electrónico entrante pico
Messaging Server MTA salida	2	Añada 1 CPU por correo electrónico saliente pico
Messaging Server MMP	2	Añada 1 CPU por carga adicional
Messaging Server STR (almacén de mensajes)	2	Añada 1 CPU por carga adicional
Directory Server	3	Añada 1 CPU por búsquedas LDAP adicionales

## Modifique los cálculos para otros estados de carga

Continúe con los cálculos de CPU para tener en cuenta otros requisitos de calidad de servicio que puedan afectar a la carga:

- **Seguridad.** A partir de la etapa de requisitos técnicos, determine cómo puede afectar el transporte seguro de los datos a los requisitos de carga y realice las modificaciones correspondientes en sus cálculos. La siguiente sección, [“Cálculo de los requisitos de procesadores para transacciones seguras” en la página 84](#), describe un proceso para realizar los ajustes.
- **Duplicación de los servicios.** Ajuste los cálculos de CPU para tener en cuenta la duplicación de los servicios para incluir las consideraciones de disponibilidad, equilibrado de carga y escalabilidad. La siguiente sección, [“Determinación de las estrategias de disponibilidad” en la página 88](#), debate el tamaño de las soluciones de disponibilidad. La sección [“Determinación de estrategias para la escalabilidad” en la página 97](#) plantea las soluciones que implican contar con acceso a los servicios de directorios.
- **Capacidad latente y escalabilidad.** Modifique los cálculos de CPU según sea necesario para permitir una capacidad latente para cargas grandes imprevistas en la implementación. Compruebe los hitos previstos para el escalamiento y el aumento de carga previsto en el tiempo para asegurarse de que puede alcanzar cualquier hito previsto para escalar el sistema, ya sea horizontal o verticalmente.

## Actualice los cálculos de CPU

Normalmente, se redondea el número de CPU a un número par. Esta operación permite dividir de forma igual los cálculos de CPU entre dos servidores físicos y también añade un pequeño factor para la capacidad latente. Sin embargo, el redondeo de acuerdo con las necesidades específicas necesita una duplicación de servicios.

Como norma general, permita 2 gigabytes de memoria para cada CPU. La memoria real necesaria depende de la utilización específica y se puede determinar en la comprobación.

La siguiente tabla muestra los cálculos finales para el ejemplo de comunicaciones basado en identidades. Estos cálculos no incluyen la potencia de cálculo adicional que se podría añadir para la seguridad y disponibilidad. Los totales para seguridad y disponibilidad se agregarán en las siguientes secciones.

**Tabla 5-4** Ajustes de cálculos de CPU para compatibilidad de componentes

Componente	CPU	Memoria
Portal Server	4	8 GB
Communications Express	2	4 GB
Messaging Server (MTA; entrada)	2	4 GB
Messaging Server (MTA; salida)	2	4 GB
Messaging Server (MMP)	2	4 GB
Messaging Server (almacén de mensajes)	2	4 GB
Access Manager	2	4 GB
Calendar Server	2	4 GB
Directory Server	4	8 GB (redondeado desde 3 CPU/6 GB de memoria)
Web Server	0	0

## Cálculo de los requisitos de procesadores para transacciones seguras

El transporte seguro de los datos implica el tratamiento de transacciones en un protocolo de transporte seguro como nivel de sockets seguro (SSL) o seguridad de nivel de transporte (TLS). Normalmente, las transacciones tratadas en un transporte seguro requieren una potencia de cálculo adicional para en primer lugar establecer una sesión segura (conocida como protocolo de enlace) y luego para cifrar y descifrar los datos transportados. En función del algoritmo de cifrado utilizado (por ejemplo, algoritmos de cifrado de 40 bits o 128 bits), la potencia de cálculo adicional puede ser importante.

Para que las transacciones seguras se realicen al mismo nivel que las transacciones no seguras, debe planificar la potencia de cálculo adicional. En función de la naturaleza de la transacción y los servicios de Sun Java™ Enterprise System que las tratan, las transacciones seguras pueden requerir hasta cuatro veces más de potencia de cálculo que las transacciones no seguras.

Al calcular la potencia de cálculo necesaria para tratar las transacciones seguras, analice los casos de uso para determinar el porcentaje de transacciones que requieren un transporte seguro. Si los requisitos de rendimiento para las transacciones seguras son iguales que para las transacciones no seguras, modifique los cálculos de CPU para tener en cuenta la potencia de cálculo adicional necesaria para las transacciones seguras.

En algunos escenarios de uso, el transporte seguro sólo puede ser necesario para la autenticación. Una vez que un usuario se autentica en el sistema, no son necesarias más medidas de seguridad para transportar los datos. En otros escenarios, el transporte seguro puede ser necesario para todas las transacciones.

Por ejemplo, al examinar un catálogo de productos en un sitio de comercio electrónico, todas las transacciones pueden ser no seguras hasta que el cliente realice todas las selecciones y esté listo para realizar la compra. Sin embargo, algunos escenarios de uso, como las implementaciones para los bancos o agentes de bolsa, requieren que la mayoría o todas las transacciones sean seguras y aplican el mismo estándar de rendimiento para las transacciones seguras y no seguras.

## Cálculos de CPU para transacciones seguras

Esta sección continúa la implementación de ejemplo para mostrar cómo calcular los requisitos de CPU para un caso de uso teórico que incluye transacciones seguras y no seguras.

Para calcular los requisitos de CPU para las transacciones seguras, lleve a cabo los siguientes cálculos:

1. Comience con una cifra de base para los cálculos de CPU (como se mostraba en la sección anterior, [“Ejemplo de cálculo de los requisitos de procesadores” en la página 79](#)).
2. Calcule el porcentaje de transacciones que requieren un transporte seguro y calcule las CPU previstas para las transacciones seguras.
3. Calcule el número de CPU reducido para las transacciones no seguras.
4. Anote el número para las transacciones seguras y para las no seguras para calcular el número total de CPU.
5. Redondee el número de CPU total a un número par.

La [Tabla 5-5](#) muestra un cálculo de ejemplo basado en casos de uso y análisis de uso para Portal Server que suponen lo siguiente:

- Todos los inicios de sesión requieren autenticación segura.
- Todos los inicios de sesión suponen un 10% de la carga total de Portal Server.
- El requisito de rendimiento para las transacciones seguras es el mismo que el requisito de rendimiento para transacciones no seguras.

Para contar con la potencia de cálculo extra necesaria para tratar las transacciones seguras, el número de CPU necesarias para tratar estas transacciones se aumentará por cuatro. Al igual que otras cifras de CPU del ejemplo, este factor es arbitrario y simplemente tiene una finalidad ilustrativa.

**Tabla 5-5** modificación de los cálculos de CPU para transacciones seguras

Paso	Descripción	Cálculo	Resultado
1	Comience con un cálculo de base para todas las transacciones de Portal Server.	El cálculo de base a partir de la <a href="#">Tabla 5-3 en la página 82</a> es 4 CPU.	-----
2	Calcule el número de CPU adicional para las transacciones seguras. Suponga que las transacciones seguras requieren cinco veces más de potencia de CPU que las transacciones no seguras.	El diez por ciento del cálculo de base requiere un transporte seguro: $0,10 \times 4 \text{ CPU} = 0,4 \text{ CPU}$ Aumente la potencia de CPU para transacciones seguras en un factor de cuatro: $4 \times 0,4 = 1,6 \text{ CPU}$	1,6 CPU
3	Calcule el número de CPU reducido para las transacciones no seguras.	El noventa por ciento del cálculo de base son transacciones no seguras: $0,9 \times 4 \text{ CPU} = 3,6 \text{ CPU}$	3,6 CPU
4	Calcule el número total de CPU ajustado para las transacciones seguras y no seguras.	Cálculo para transacciones seguras + no seguras = total: $1,6 \text{ CPU} + 3,6 \text{ CPU} = 5,2 \text{ CPU}$	5,2 CPU
5	Redondee a un número par.	$5,2 \text{ CPU} \Rightarrow 6 \text{ CPU}$	6 CPU

A partir de los cálculos para transacciones seguras de este ejemplo, debería modificar el número total de CPU en la [Tabla 5-5 en la página 86](#) añadiendo dos CPU y cuatro gigabytes de memoria para obtener el siguiente total para Portal Server.

**Tabla 5-6** Ajustes de cálculos de CPU para transacciones seguras en Portal Server

Componente	CPU	Memoria
Portal Server	6	12 GB

## Hardware especializado para tratar las transacciones SSL

Los dispositivos de hardware especializados, como las tarjetas aceleradoras SSL y otros dispositivos, están disponibles para ofrecer potencia de cálculo y establecer sesiones seguras y el cifrado y descifrado de los datos. Al utilizar hardware especializado para las operaciones SSL, se dedica una potencia de cálculo para parte de los cálculos SSL, normalmente la operación de protocolo de enlace que establece una sesión segura.

Este hardware puede ser beneficioso para la arquitectura de implementación final. Sin embargo, debido a la naturaleza especializada del hardware, calcule en primer lugar los requisitos de rendimiento para transacciones seguras en términos de potencia de CPU y luego considere las ventajas de utilizar hardware especializado para tratar la carga adicional.

Algunos factores que se deben considerar al utilizar hardware especializado es si los casos de uso admiten la utilización de hardware (por ejemplo, casos de uso que requieren un gran número de operaciones de protocolo de enlace SSL) y la capa añadida de complejidad que este hardware aporta al diseño. Esta complejidad incluye la instalación, configuración, pruebas y administración de estos dispositivos.

# Determinación de las estrategias de disponibilidad

Al desarrollar una estrategia para requisitos de disponibilidad, estudie las interacciones entre componentes y el análisis de uso para determinar las soluciones de disponibilidad que debe considerar. Realice el análisis componente por componente, determinando la solución que mejor se ajuste a los requisitos de disponibilidad y recuperación tras error.

Los siguientes elementos son ejemplos del tipo de información que recopila para determinar las estrategias de disponibilidad:

- ¿Cuál es el porcentaje de disponibilidad especificado?
- ¿Cuáles son las especificaciones de rendimiento con respecto a las situaciones de recuperación tras error (por ejemplo, al menos un 50% de rendimiento durante la recuperación)?
- ¿El análisis de uso identifica los tiempos de uso máximo?
- ¿Cuáles son las consideraciones geográficas?

La estrategia de disponibilidad que elija también debe tener en cuenta los requisitos de facilidad de mantenimiento, como se plantea en [“Determinación de estrategias para la escalabilidad” en la página 97](#). Evite las soluciones complejas que requieren un gran esfuerzo de mantenimiento y administración.

## Estrategias de disponibilidad

Las estrategias de disponibilidad para las implementaciones de Java Enterprise System incluyen:

- **Equilibrado de carga.** Utiliza los componentes de hardware y software redundantes para compartir una carga de procesamiento. Un equilibrador de carga dirige todas las solicitudes de un servicio a uno de las varias instancias simétricas del servicio. Si cualquiera de estas instancias fallase, hay otras instancias disponibles para asumir una mayor carga.
- **Recuperación tras error.** Implica la gestión del hardware y software redundante para proporcionar un acceso continuo a los servicios y seguridad para los datos críticos en caso de que alguno de los componentes falle.

El software Sun Cluster proporciona una solución de recuperación tras error para los datos críticos gestionados por los componentes de servidores como el almacenamiento de mensajes de Messaging Server y los datos de calendario de Calendar Server.

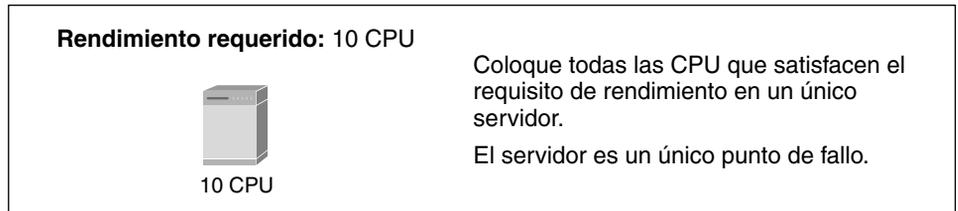
- **Duplicación de servicios.** La duplicación de servicios proporciona varias fuentes para acceder a los mismos datos. Directory Server proporciona varias estrategias de duplicación y sincronización para el acceso al directorio LDAP.

Las siguientes secciones proporcionan algunos ejemplos de soluciones de disponibilidad que proporcionan varios niveles de equilibrado de carga, recuperación tras error y duplicación de servicios.

## Sistema de un servidor

Coloque todos los recursos de cálculo de un servicio en un único servidor. Si el servidor falla, todo el servicio falla.

**Figura 5-2** Sistema de un servidor



Sun ofrece servidores de gama alta que proporcionan las siguientes ventajas:

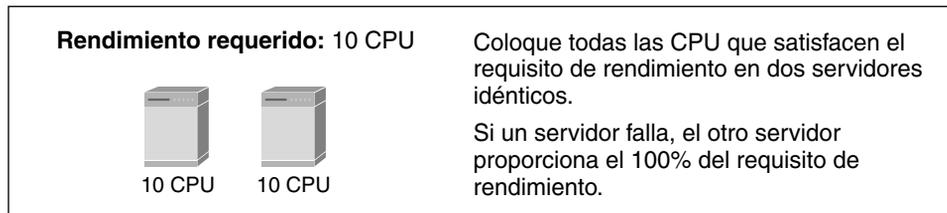
- Sustitución y reconfiguración de componentes de hardware mientras el sistema se está ejecutando
- Capacidad para ejecutar varias aplicaciones en dominios a prueba de fallos en un servidor
- Posibilidad de actualizar la capacidad, la velocidad de rendimiento y la configuración de E/S sin reiniciar el sistema

Normalmente, un servidor de gama alta cuesta más que un sistema multiservidor comparable. Sin embargo, un único servidor proporciona ahorros en cuanto a los costes de administración, supervisión y hospedaje con respecto a los servidores en un centro de datos. Las funciones de equilibrador de carga, recuperación tras error y eliminación de puntos de fallo son más flexibles en los sistemas multiservidor.

## Sistemas redundantes horizontalmente

Existen varias maneras de aumentar la disponibilidad con servidores redundantes paralelos que proporcionan equilibrado de carga y recuperación tras errores. La siguiente figura muestra dos servidores duplicados que proporcionan un sistema de recuperación tras error N+1. Un sistema N+1 cuenta con un servidor adicional para proporcionar el 100% de capacidad en caso de que un servidor falle.

**Figura 5-3** Sistema de recuperación tras error N+1 con dos servidores

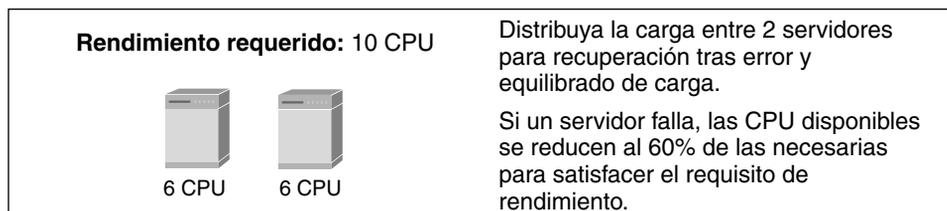


La potencia de cálculo de cada servidor en la [Figura 5-3](#) anterior es idéntica. Sólo un servidor trata los requisitos de rendimiento. El otro servidor proporciona el 100% del rendimiento cuando llama a un servicio como copia de seguridad.

La ventaja de un diseño de recuperación tras error N+1 es un 100% de rendimiento durante una situación de recuperación tras error. Las desventajas incluyen un aumento en los costes de hardware sin una ganancia correspondiente en el rendimiento general (debido a que un servidor se encuentra en modo en espera para utilizarse únicamente en las situaciones de recuperación tras error).

La siguiente figura ilustra un sistema que implementa el equilibrado de carga más una recuperación de errores que distribuye el rendimiento entre dos servidores.

**Figura 5-4** Equilibrado de carga más recuperación tras error en dos servidores

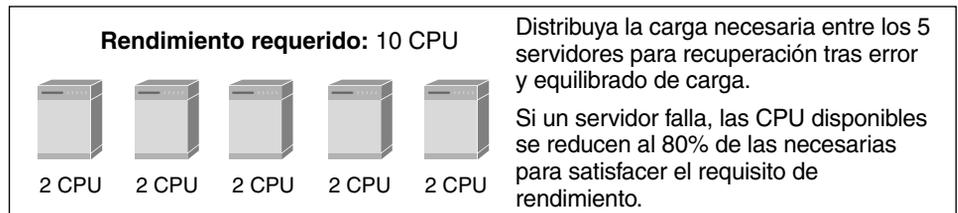


En el sistema que se muestra en la [Figura 5-4](#) anterior, si un servidor falla, todos los servicios están disponibles, aunque a un porcentaje de la capacidad completa. El servidor restante proporciona 6 CPU de potencia de cálculo, que es el 60% del requisito de 10 CPU.

Una ventaja de este diseño es la capacidad latente de 2 CPU cuando ambos servidores están disponibles.

La siguiente figura ilustra una distribución entre un número de servidores para el rendimiento y equilibrado de carga.

**Figura 5-5** Distribución de la carga entre  $n$  servidores



Debido a que hay cinco servidores en el diseño que se muestra en la [Figura 5-5](#), si un servidor falla, el resto de los servidores proporcionan un total de 8 CPU de potencia de cálculo, que es el 80% del requisito de rendimiento de 10 CPU. Si añade un servicio adicional con una capacidad de 2 CPU al diseño, tendrá efectivamente un diseño N+1. Si un servidor falla, se satisface el 100% del rendimiento por los servidores restantes.

Este diseño incluye las siguientes ventajas:

- Rendimiento adicional en caso de que un servidor falle
- Disponibilidad incluso cuando falla más de un servidor
- Los servidores se pueden rotar para realizar el mantenimiento y llevar a cabo actualizaciones
- Normalmente, varios servidores de gama baja cuestan menos que un único servidor de gama alta

Sin embargo, los costes de administración y mantenimiento pueden aumentar significativamente con servidores adicionales. También se debe considerar el coste del hospedaje de los servidores en un centro de datos. En un momento dado, se obtiene una menor rentabilidad al agregar servidores adicionales.

## Software de Sun Cluster

Para las situaciones que requieren un mayor grado de disponibilidad (como cuatro o cinco nueves), deberá considerar el software de Sun Cluster como parte de su diseño de disponibilidad. Un sistema de clúster es el acoplamiento de servidores redundantes con almacenamiento y otros recursos de red. Los servidores de un clúster se comunican continuamente entre sí. Si uno de los servidores pierde la conexión, el resto de los dispositivos del clúster aíslan el servidor y recuperan cualquier aplicación o datos que haya fallado desde el nodo que haya fallado u otro nodo. Este proceso de recuperación tras error se obtiene de forma relativamente rápida con una pequeña interrupción del servicio a los usuarios del sistema.

El software de Sun Cluster requiere un hardware adicional dedicado y conocimientos especializados para realizar la configuración, administración y mantenimiento.

## Ejemplo de diseño de disponibilidad

Esta sección contiene dos ejemplos de estrategias de disponibilidad basadas en la solución de comunicaciones basadas en identidades para una empresa de tamaño medio con un número de empleados entre 1.000 y 5.000, como se describió anteriormente en [“Ejemplo de comunicaciones basadas en identidades” en la página 65](#). La primera estrategia de disponibilidad ilustra el equilibrio de carga para Messaging Server. La segunda ilustra una solución de recuperación tras error que utiliza software de Sun Cluster.

### Ejemplo de equilibrado de carga para Messaging Server

La siguiente muestra los cálculos de potencia de CPU para cada componente lógico de Messaging Server en la arquitectura lógica. Esta tabla repite el cálculo final determinado en la sección [“Actualice los cálculos de CPU” en la página 83](#).

**Tabla 5-7** Ajustes de cálculos de CPU para compatibilidad de componentes

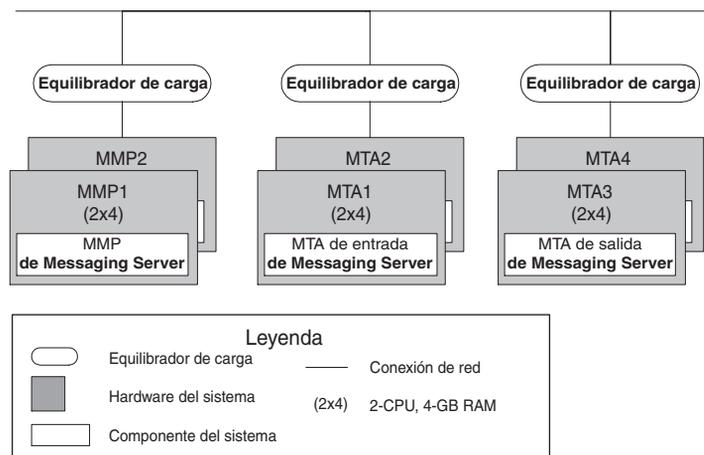
Componente	CPU	Memoria
Messaging Server (MTA; entrada)	2	4 GB
Messaging Server (MTA; salida)	2	4 GB
Messaging Server (MMP)	2	4 GB
Messaging Server (almacén de mensajes)	2	4 GB

Para este ejemplo, asuma que durante la etapa de requisitos técnicos, se especificaron los siguientes requisitos de calidad de servicio:

- **Disponibilidad.** La disponibilidad general del sistema debe ser 99,99% (no incluye el tiempo de inactividad programado). El fallo de un sistema informático individual no debe traducirse en un fallo del servicio.
- **Escalabilidad.** Ningún servidor debe tener más de un 80% de capacidad utilizada en la carga máxima diaria y el sistema debe ser capaz de aumentar un 10% anualmente.

Para satisfacer el requisito de disponibilidad, para cada componente de Messaging Server se deben proporcionar dos instancias, una de cada servidor de hardware separado. Si un servidor de un componente falla, el otro presta el servicio. La siguiente figura muestra el diagrama de red para esta estrategia de disponibilidad.

**Figura 5-6** Ejemplo de estrategia de disponibilidad para Messaging Server



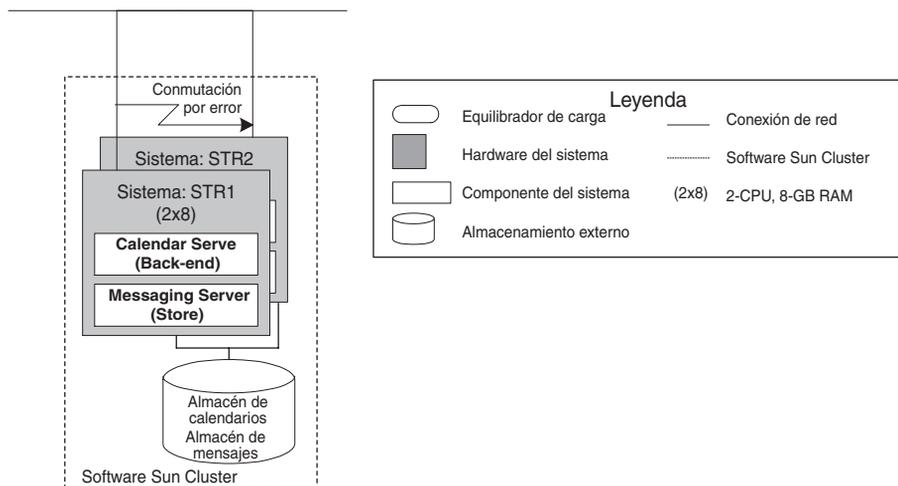
En la figura anterior, el número de CPU se ha duplicado desde su cálculo original. Las CPU se doblan por los siguientes motivos:

- En el caso de que se produzca un fallo en el servidor, el servidor restante proporciona la potencia de CPU necesaria para encargarse de la carga.
- Para el requisito de escalabilidad de que ningún servidor utilice más del 80% de su capacidad en una carga pico, la potencia de CPU ofrece este margen de seguridad.
- En cuanto al requisito de escalabilidad de un aumento anual del 10% de la carga, la potencia de CPU adicional añade la capacidad latente que puede tratar el aumento de cargas hasta que se necesite un escalamiento adicional.

## Ejemplo de recuperación tras error utilizando el software de Sun Cluster

La siguiente figura muestra un ejemplo de estrategia de recuperación tras error para Calendar Server de servidor y el almacén de mensajes de Messaging Server. El almacén de mensajes y entorno de servidor de Calendar Server se duplican en servidores de hardware independientes y se configuran para la recuperación tras error con el software de Sun Cluster. El número de CPU y la memoria correspondiente se duplica en cada servidor de Sun Cluster.

**Figura 5-7** Diseño de recuperación tras error utilizando el software de Sun Cluster



## Ejemplo de duplicación de servicios de directorio

Los servicios de Directory Server se pueden duplicar para distribuir las transacciones en varios servidores, proporcionando una alta disponibilidad. Directory Server proporciona varias estrategias para la duplicación de los servicios, como las siguientes:

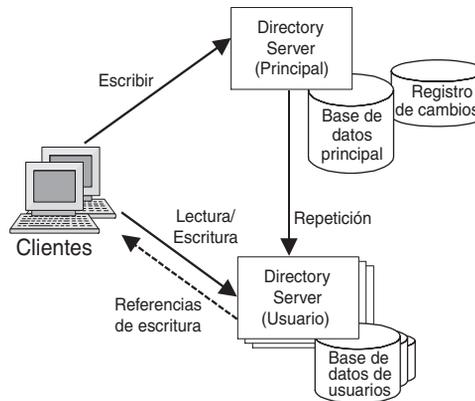
- **Varias bases de datos.** Almacena distintas partes del árbol de directorios en bases de datos separadas.
- **Encadenamiento y referencias.** Vincula los datos distribuidos en un único árbol de directorios.
- **Repetición de una réplica principal.** Proporciona un origen centralizado para la base de datos principal que luego se distribuye en réplicas de usuario.
- **Repetición de varias réplicas principales.** Distribuye la base de datos principal entre varios servidores. Cada una de estas réplicas principales distribuye su base de datos entre sus réplicas de usuarios.

Las estrategias de disponibilidad para Directory Server es un tema complejo que no se engloba dentro del ámbito de esta guía. Las siguientes secciones, “[Repetición de una réplica principal](#)” y “[Repetición de varias réplicas principales](#)” proporcionan una descripción general de las estrategias básicas de duplicación. Para obtener información detallada acerca de las estrategias de disponibilidad para Directory Server, consulte la *Directory Server Guía de planificación de la implementación*, <http://docs.sun.com/doc/817-7607>.

## Repetición de una réplica principal

La siguiente figura muestra una estrategia de duplicación de una réplica principal que ilustra los conceptos de replicación básicos.

**Figura 5-8** Ejemplo de repetición de una réplica principal



En la duplicación de una réplica maestra, una instancia de Directory Server gestiona la base de datos de directorios principal, registrando todos los cambios. La base de datos principal se duplica en cualquier número de bases de datos de usuario. Las instancias del usuario de Directory Server se optimizan para las operaciones de lectura y búsqueda. Cualquier operación de escritura recibida por un usuario se devuelve a la base de datos principal. La base de datos principal actualiza periódicamente las bases de datos de los usuarios.

Entre las ventajas de la repetición de una única réplica principal se incluyen:

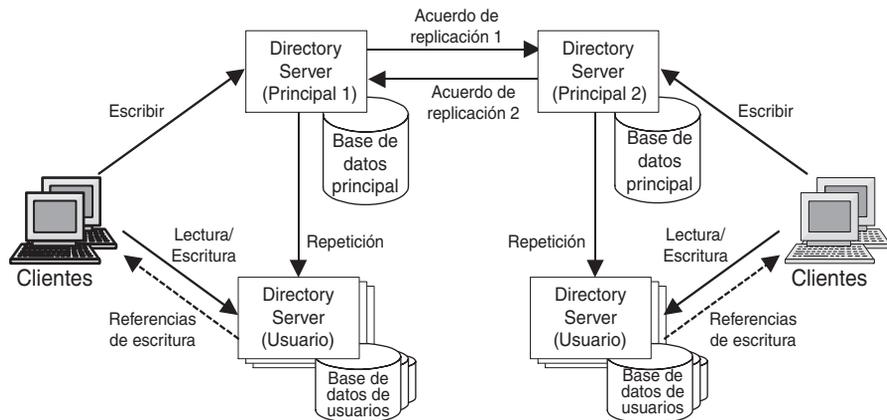
- Una instancia de Directory Server optimizada para las operaciones de lectura y escritura de la base de datos
- Un número de instancias del usuario de Directory Server optimizadas para las operaciones de lectura y búsqueda.
- Escalabilidad horizontal para las instancias de usuario de Directory Server

## Repetición de varias réplicas principales

La siguiente figura muestra una estrategia de repetición de varias réplicas principales que se puede utilizar para distribuir globalmente el acceso a los directorios.

En una repetición de varias réplicas principales, una o más instancias de Directory Server gestionan la base de datos de directorios principal. Cada réplica principal cuenta con un acuerdo de duplicación que especifica los procedimientos para sincronizar las bases de datos principales. Cada réplica principal se duplica en una serie de bases de datos de usuario. Al igual que con la repetición de una única réplica principal, las instancias de usuario de Directory Server se optimizan para el acceso de lectura y escritura. Cualquier operación de escritura recibida por un usuario se devuelve a la base de datos principal. La base de datos principal actualiza periódicamente las bases de datos de los usuarios.

**Figura 5-9** Ejemplo de repetición de varias réplicas principales



La estrategia de repetición de varias réplicas principales proporciona todas las ventajas de la repetición de una única réplica maestra más una estrategia de disponibilidad que puede proporcionar el equilibrado de carga para realizar las actualizaciones a las réplicas principales. También puede implementar una estrategia de disponibilidad que proporciona un control local de las operaciones de directorio, que es una consideración importante para las empresas con centros de datos distribuidos globalmente.

# Determinación de estrategias para la escalabilidad

La escalabilidad es la capacidad para aumentar la capacidad del sistema, normalmente mediante la adición de recursos de sistema, pero sin realizar cambios en la arquitectura de implementación. En el análisis de requisitos, normalmente se realizan predicciones del crecimiento previsto para un sistema basado en los requisitos empresariales y en los siguientes análisis de uso. Estas predicciones del número de usuarios de un sistema y la capacidad del sistema para satisfacer sus necesidades son a menudo cálculos que pueden variar en gran medida con respecto a los números reales del sistema implementado. El diseño debe ser lo suficientemente flexible como permitir una variación con respecto a las predicciones.

Un diseño escalable incluye la suficiente capacidad latente para tratar el aumento de la carga hasta que un sistema se pueda actualizar con recursos adicionales. Este tipo de diseños se puede escalar fácilmente para tratar el aumento de la carga sin tener que volver a diseñar el sistema.

## Capacidad latente

La capacidad latente es uno de los aspectos de escalabilidad en los que se incluyen los recursos de disponibilidad y rendimiento adicionales en el sistema de forma que se puedan afrontar cargas pico inusuales. También puede supervisar cómo se utiliza la capacidad latente en un sistema implementado para ayudar a determinar cuándo se debe escalar el sistema añadiendo recursos. La capacidad latente es una manera de crear seguridad en el diseño.

El análisis de casos de uso puede ayudar a identificar los escenarios que pueden crear cargas pico inusuales. Utilice este análisis de cargas máximas inusuales más un factor para cubrir el crecimiento imprevisto para diseñar la capacidad latente que crea seguridad en el sistema.

El diseño del sistema debe ser capaz de gestionar la capacidad prevista durante un tiempo razonable, normalmente los 6 a 12 primeros meses de funcionamiento. Los ciclos de mantenimiento se pueden utilizar para agregar recursos o aumentar la capacidad según se necesite. Idealmente, se deberían poder programar las actualizaciones del sistema periódicamente, pero a menudo, la predicción de los aumentos en la capacidad necesarios es una tarea difícil. Para determinar el momento de actualizar un servicio, se deberá realizar una supervisión exhaustiva de los recursos y de las previsiones empresariales.

Si tiene previsto implementar su solución en etapas progresivas, deberá programar el aumento de la capacidad del sistema para que coincida con otras mejoras programadas en cada etapa.

## Ejemplo de escalabilidad

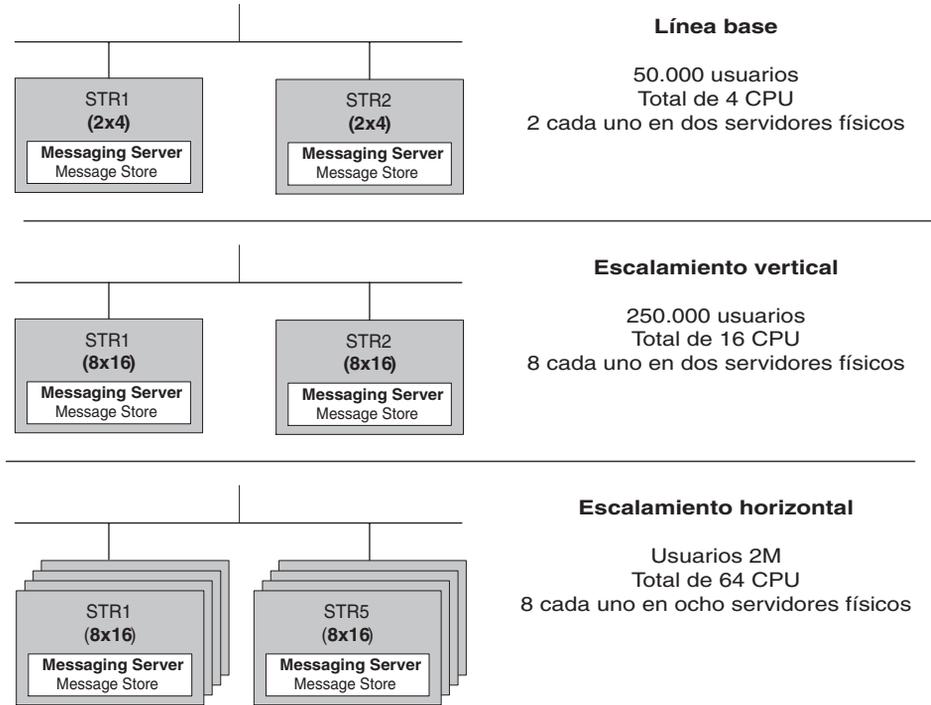
El ejemplo de esta sección muestra el escalamiento horizontal y vertical de una solución que implementa Messaging Server. Para el escalamiento horizontal, se añaden CPU adicionales a un servidor para que pueda gestionar el aumento de la carga. Para el escalamiento vertical, se gestiona el aumento de la carga mediante la adición de servidores para poder distribuir la carga.

La base de este ejemplo supone una base de 50.000 usuarios que reciben el servicio mediante dos instancias de almacenamiento de mensajes que están distribuidas para el equilibrio de carga. Cada servidor cuenta con dos CPU para un total de cuatro CPU. La siguiente figura muestra cómo se puede escalar el sistema para tratar el aumento de cargas para 250.000 usuarios y 2.000.000 usuarios.

---

**NOTA** La [Figura 5-10](#) muestra las diferencias entre el escalamiento vertical y el escalamiento horizontal. Esta figura no muestra otros factores que se deben tener en cuenta al realizar el escalamiento, como el equilibrado de carga, la recuperación tras error y los cambios en los patrones de uso.

---

**Figura 5-10** Ejemplos de escalamiento horizontal y vertical

## Identificación de cuellos de botella de rendimiento

Una de las claves para un diseño de implementación con éxito consiste en identificar los cuellos de botella del rendimiento y desarrollar una estrategia para evitarlos. Un cuello de botella del rendimiento se produce cuando la velocidad de acceso a los datos no cumple con los requisitos especificados para el sistema.

Los cuellos de botella se pueden ordenar de acuerdo con distintos tipos de hardware, tal y como se indica en la siguiente tabla de puntos de acceso de un sistema. Esta tabla también sugiere las soluciones potenciales para los cuellos de botella en cada clase de hardware.

**Tabla 5-8** Puntos de acceso a los datos

Clase de hardware	Velocidad de acceso relativa	Soluciones para la mejora del rendimiento
Procesador	Nanosegundos	<p>Escalamiento vertical: Añada más potencia de procesamiento, mejore la caché del procesador</p> <p>Escalamiento horizontal: Añada potencia de procesamiento paralelo para el equilibrado de carga</p>
Memoria del sistema (RAM)	Microsegundos	<p>Memoria del sistema dedicada para tareas específicas</p> <p>Escalamiento vertical: añade memoria adicional</p> <p>Escalamiento horizontal: cree instancias adicionales para el procesamiento paralelo y el equilibrado de carga</p>
Lectura y escritura en disco	Milisegundos	<p>Optimice el acceso al disco con matrices de disco (RAID)</p> <p>Dedique el acceso al disco a funciones específicas, como sólo lectura o sólo escritura</p> <p>Introduzca en caché los datos a los que se accede frecuentemente en la memoria del sistema</p>
Interfaz de red	Varía en función del ancho de banda y la velocidad de acceso de los nodos de la red	<p>Aumente el ancho de banda</p> <p>Añada hardware acelerador al transportar los datos seguros</p> <p>Mejore el rendimiento de los nodos de la red, de forma que los datos estén accesibles más rápidamente</p>

**NOTA** La [Tabla 5-8](#) muestra las clases de hardware en función de la velocidad de acceso relativa, implicando que los puntos de acceso lentos, como los discos, tienen una mayor probabilidad de ser la fuente de los cuellos de botella. Sin embargo, unos procesadores que no tengan la suficiente potencia para gestionar grandes cargas también son una fuente de cuellos de botella.

Normalmente, se comienza el diseño de implementación con los cálculos de potencia de procesamiento de base para cada componente de la implementación y sus relaciones de dependencia. A continuación se determina cómo evitar los cuellos de botella relacionados con la memoria del sistema y acceso a los discos. Por último, se examina la interfaz de la red para determinar los cuellos de botella potenciales y centrarse en las estrategias que se deben implantar para solucionarlos.

## Optimización de acceso al disco

Un componente vital del diseño de implementación es la velocidad de acceso al disco para los conjuntos de datos a los que se accede frecuentemente, como los directorios LDAP. El acceso al disco ofrece el acceso más lento a los datos y es probablemente el origen de un cuello de botella de rendimiento.

Una forma de optimizar el acceso al disco es separar las operaciones de escritura de las de lectura. Las operaciones de escritura no sólo son más caras que las de lectura, sino que normalmente las operaciones de lectura (operaciones de búsqueda en los directorios LDAP) se producen con mucha más frecuencia que las de escritura (actualizaciones a los datos en los directorios LDAP).

Otra manera de optimizar el acceso al disco es mediante discos dedicados en función de los distintos tipos de operaciones de E/S. Por ejemplo, proporcione un acceso al disco separado para las operaciones de registro de Directory Server, como registros de transacciones y de eventos, y operaciones de lectura y escritura LDAP.

Asimismo, considere la implementación de una o más instancias de Directory Server dedicadas a las operaciones de lectura y escritura y la utilización de instancias duplicadas distribuidas en servidores locales para el acceso de lectura y de búsqueda. Las opciones de encadenamiento y enlace también están disponibles para optimizar el acceso a los servicios de directorios.

El capítulo “Tamaño del sistema” en la *Directory Server Guía de planificación de la implementación*, <http://docs.sun.com/doc/817-7607>, plantea varios factores en la planificación del acceso al disco. Entre los temas tratados en este capítulo, se incluyen:

- **Requisitos mínimos de memoria y espacio en disco.** Ofrecer una estimación de la memoria y capacidad de disco necesarios para los distintos tamaños de directorios.
- **Establecimiento de tamaño de memoria física para el acceso a caché.** Ofrece unas directrices para estimar el tamaño de la caché de acuerdo con la utilización prevista de Directory Server y con la planificación de utilización total de la memoria.
- **Establecimiento del tamaño de los subsistemas de disco.** Ofrece información sobre los requisitos de espacio en disco de acuerdo con los sufijos de directorio y los factores de Directory Server que afectan a la utilización del disco y a la distribución de los archivos en los discos, incluidas varias alternativas de matrices de discos.

## Diseño para la óptima utilización de los recursos

El diseño de la implementación no es simplemente el cálculo de los recursos necesarios para satisfacer los requisitos de calidad de servicio. En el diseño de la implementación también se analizan todas las opciones disponibles y se selecciona la mejor solución que minimiza los costes pero que satisface los requisitos de calidad del servicio. Debe analizar los cambios asociados a cada decisión del diseño para asegurar que las ventajas que se ofrecen en un área no se ven anuladas por un incremento del coste en otra.

Por ejemplo, el escalamiento horizontal para la disponibilidad puede aumentar la disponibilidad general pero al coste de un mayor mantenimiento y servicio. El escalamiento vertical para el rendimiento puede aumentar la potencia de cálculo sin incurrir en gastos adicionales, pero la potencia adicional se puede utilizar de forma poco eficaz en algunos servicios.

Antes de completar la estrategia de diseño, examine las decisiones para asegurarse de que ha equilibrado la utilización de los recursos con las ventajas generales de la solución propuesta. Este análisis implica normalmente el examen de cómo afectan las calidades del sistema en un área a las de otra. La siguiente tabla muestra algunas calidades del sistema y las consideraciones correspondientes para la gestión de recursos.

**Tabla 5-9** Consideraciones para la gestión de recursos

<b>Calidad del sistema</b>	<b>Descripción</b>
Rendimiento	En el caso de soluciones de rendimiento que concentran CPU en servidores individuales, ¿podrán los servicios utilizar eficazmente la potencia de cálculo? (Por ejemplo, algunos servicios tienen un límite sobre el número máximo de CPU que se pueden utilizar de forma eficaz.)
Capacidad latente	<p>¿La estrategia gestiona cargas que superan los cálculos de rendimiento?</p> <p>¿Las cargas excesivas se tratan mediante el escalamiento vertical en los servidores, equilibrando la carga en otros servidores o con ambas acciones?</p> <p>¿La capacidad latente es suficiente para tratar las cargas pico inusuales hasta que alcance el siguiente hito de escalamiento de la implementación?</p>
Seguridad	¿Ha previsto el aumento de rendimiento necesario para tratar las transacciones seguras?
Disponibilidad	<p>Para las soluciones redundantes horizontalmente, ¿ha calculado los gastos de mantenimiento a largo plazo?</p> <p>¿Ha tenido en cuenta el tiempo de inactividad programado necesario para mantener el sistema?</p> <p>¿Ha equilibrado los costes entre servidores de gama alta y de gama baja?</p>
Escalabilidad	<p>¿Ha calculado hitos para escalar la implementación?</p> <p>¿Cuenta con una estrategia adecuada para ofrecer la suficiente capacidad latente para tratar los aumentos de carga previstos hasta que se alcancen los hitos para escalar la implementación?</p>
Facilidad de mantenimiento	<p>¿Ha tenido en cuenta los costes de administración, supervisión y mantenimiento en el diseño de disponibilidad?</p> <p>¿Ha considerado soluciones de administración delegada (que permiten que los usuarios finales realicen algunas tareas de administración) para reducir los costes de administración?</p>

# Gestión de riesgos

La mayor parte de la información en la que está basada el diseño de la implementación, como los requisitos de calidad de servicio y análisis de uso, no está compuesta por datos empíricos sino por datos basados en los cálculos y proyecciones obtenidos en última instancia de los análisis de negocios. Estas previsiones pueden ser imprecisas por varios motivos, como circunstancias imprevistas en el clima empresarial, métodos defectuosos de recopilación de los datos o simplemente debido a errores humanos. Antes de completar un diseño de implementación, vuelva a comprobar los análisis en los que está basado el diseño y asegúrese de que el diseño tiene en cuenta cualquier desviación razonable de los cálculos o previsiones.

Por ejemplo, si el análisis de uso subestima la utilización real del sistema, corre el riesgo de crear un sistema que no pueda manejar la cantidad de tráfico que se encuentre. Un diseño que tenga un rendimiento inferior al previsto se considerará un fallo.

Por otro lado, si crea un sistema que es mucho más potente de lo necesario, estará utilizando recursos que se podrían utilizar en otro lugar. La clave es incluir un margen de seguridad por encima de los requisitos pero evitar el uso desmesurado de los recursos.

El uso desmesurado de los recursos se traduce en un error del diseño porque los recursos infrautilizados se podrían aplicar a otras áreas. Además, los participantes pueden percibir estas soluciones de recursos excesivos como que no cumplen con los contratos en buena fe.

## Ejemplo de arquitectura de implementación

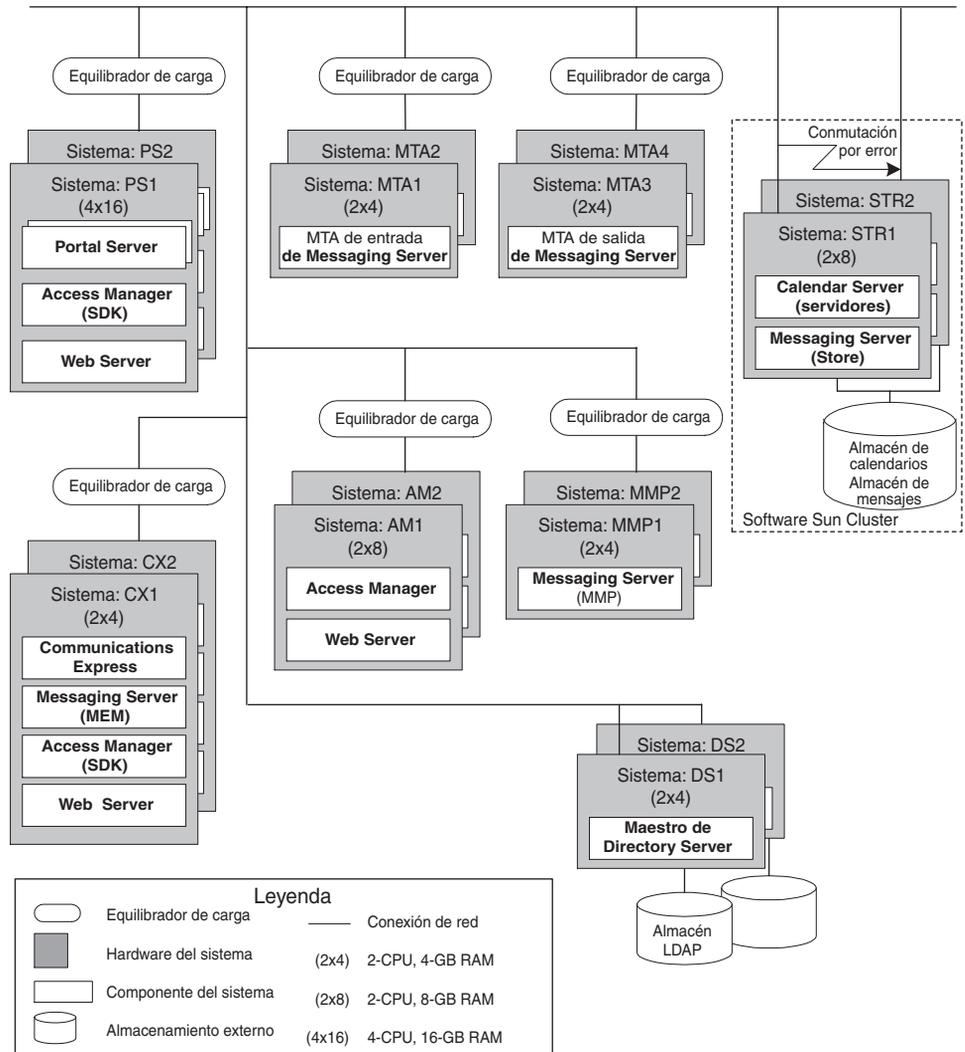
La siguiente figura representa una arquitectura de implementación completada para la implementación de ejemplo presentada anteriormente en esta guía. Esta figura proporciona una idea de cómo se debe presentar una arquitectura de implementación.

---

**PRECAUCIÓN** La arquitectura de implementación de la siguiente figura sólo es válida para fines ilustrativos. No representa una implementación que se haya diseñado, creado o probado y no debería considerarse como un consejo para la planificación de la implementación.

---

Figura 5-11 Ejemplo de arquitectura de implementación



Ejemplo de arquitectura de implementación

# Implantación de un diseño de implementación

En la etapa de implantación del ciclo de vida de la solución, se trabaja a partir de las especificaciones y planes creados durante el diseño de implementación y se comprueba la arquitectura de implementación, que en última instancia despliega la implementación en el entorno de producción. Aunque la implementación no se encuentra dentro del ámbito de esta guía, este capítulo realiza una descripción general de esta etapa.

Este capítulo incluye los siguientes apartados:

- “Información acerca de la implantación de diseños de implementación” en la página 108
- “Instalación y configuración del software” en la página 108
- “Desarrollo de pilotos y prototipos” en la página 109
- “Desarrollo de pilotos y prototipos” en la página 109
- “Comprobación de las implementaciones piloto y prototipo” en la página 110
- “Despliegue de una implementación de producción” en la página 111

# Información acerca de la implantación de diseños de implementación

Una vez que la arquitectura de implementación se ha aprobado y se han completado los planes y especificaciones de implementación, se inicia la etapa de implementación del ciclo de vida de la solución. La implementación es un conjunto complejo de procesos y procedimientos que requieren una detallada planificación para asegurar su éxito. La implementación incluye las siguientes tareas:

- Creación de la infraestructura de red y de hardware.
- Instalación y configuración del software de acuerdo con un plan de instalación.
- Migración de los datos de las aplicaciones existentes a la solución actual.
- Implementación de un plan de administración de usuarios.
- Diseño y despliegues de pilotos y prototipos en un entorno de prueba de acuerdo con el plan de prueba.
- Diseño y ejecución de pruebas funcionales y pruebas de estrés de acuerdo con un plan de prueba.
- Despliegue de la solución desde un entorno de prueba en un entorno de producción de acuerdo con el plan de despliegue.
- Formación de los administradores y usuarios de la implementación de acuerdo con el plan de formación.

Los detalles de la implementación no se encuadran dentro del ámbito de esta guía. Sin embargo, las siguientes secciones proporcionan información general acerca de algunas de estas tareas.

## Instalación y configuración del software

La instalación y configuración de Sun Java™ Enterprise System para una aplicación empresarial distribuida requiere la planificación y coordinación de varias tareas y procedimientos. Durante la etapa de diseño de implementación, se crea un plan de instalación basado en la arquitectura general de implementación que proporciona la información acerca de la configuración e instalación necesaria para instalar el software Java Enterprise System.

Los puntos destacados de este plan de instalación incluyen:

- Determinación de la secuencia y tipo de instalación.
- Supervisión de hosts de software instalado anteriormente y preparación de la instalación.
- Recopilación de información acerca de la configuración de cada componente de Java Enterprise System que esté instalando.

*Guía de instalación de Java Enterprise System*, <http://docs.sun.com/doc/819-0806>, proporciona información sobre cómo recopilar información para un plan de instalación. La guía de instalación proporciona información detallada acerca de la configuración así como hojas de trabajo que se pueden utilizar para documentar esta información. La guía de instalación también proporciona directrices sobre los escenarios más comunes de instalación que implican varios componentes de Java Enterprise System. Para obtener más información, consulte la sección “Preparación de la instalación” en *Guía de instalación de Java Enterprise System*, <http://docs.sun.com/doc/819-0806>.

## Desarrollo de pilotos y prototipos

Normalmente, las implementaciones de Java Enterprise System se pueden clasificar en dos categorías: las que están basadas en los servicios prestados con Java Enterprise System y las que requieren un importante número de servicios personalizados integrados con los servicios Java Enterprise System. Puede pensar en el tipo anterior de implementación como una implementación 80:20 (80% de los servicios prestados por Java Enterprise System) y por tanto, en el anterior como una implementación 20:80.

Para implementaciones 80:20, normalmente se desarrolla una implementación piloto con fines de prueba durante la etapa de implementación. Debido a que las implementaciones 80:20 utilizan servicios de Java Enterprise System maduros que proporcionan funciones “estándar”, las implementaciones piloto se trasladan de forma relativamente rápida desde los pasos de desarrollo, prueba y modificación a las implementaciones de producción. Una implementación piloto no sólo comprueba las funciones de una solución, sino que también proporciona información acerca del rendimiento de los sistemas.

Las implementaciones 20:80 por otro lado, introducen nuevos servicios personalizados que no tienen el historial de interoperabilidad de las implementaciones 80:20. Por este motivo, se crea un prototipo, que es una implementación de prueba de concepto que normalmente requiere un ciclo de desarrollo, comprobación y modificación más riguroso antes de introducirse en producción. Un prototipo permite determinar si la solución propuesta resuelve el problema en un entorno de prueba. Una vez que el prototipo demuestra que la funcionalidad es suficiente, puede realizar pruebas más rigurosas y llevar a cabo una implementación piloto.

---

**NOTA** Las implementaciones empresariales reales pueden variar en gran medida en cuanto al grado de desarrollo personalizado de los servicios que requieren. La forma en que se utilizan las implementaciones piloto y prototipo para los fines de prueba depende de la complejidad y naturaleza de la implementación.

---

## Comprobación de las implementaciones piloto y prototipo

El objetivo de realizar pruebas de las implementaciones piloto y prototipo es determinar, en la medida de lo posible en las condiciones de la prueba, si la implementación satisface los requisitos del sistema y los objetivos empresariales.

Idealmente, las pruebas funcionales deberían modelar los escenarios basándose en todos los casos de uso identificados y se debería desarrollar un conjunto de parámetros para medir el cumplimiento. La comprobación funcional también puede implicar una implementación limitada en un grupo seleccionado de usuarios beta para determinar si se satisfacen los requisitos empresariales.

Las pruebas de estrés miden el rendimiento en caso de cargas máximas. Normalmente, estas pruebas utilizan una serie de entornos simulados y generadores de carga para medir el procesamiento de los datos y el rendimiento. Los requisitos de sistema para la implementación constituyen normalmente la base del diseño y la superación de las pruebas de estrés.

---

**NOTA** Las pruebas funcionales y de estrés son especialmente importantes para implementaciones de gran tamaño donde los requisitos del sistema no estén bien definidos, no haya una implementación previa en la que basar los cálculos y en las que la implementación requiere una gran cantidad de desarrollo nuevo.

---

La comprobación puede indicar problemas con la especificación del diseño de implementación y puede implicar varias interacciones de diseño, creación y prueba antes de poder desplegar la implementación en un entorno de producción. Al realizar la comprobación de las implementaciones prototipo, puede descubrir problemas con el diseño de implementación, en cuyo caso podrá volver a las etapas anteriores del ciclo de vida de la solución y tratar dichos problemas.

Asegúrese de que ha realizado pruebas exhaustivas de su diseño de implementación antes de realizar una implementación piloto. Una implementación piloto indica que ha comprobado el diseño de implementación con una serie de pruebas anteriores. Los problemas descubiertos durante la comprobación de una implementación piloto se solucionan normalmente con los parámetros del diseño de implementación.

Debido a que la comprobación nunca simula completamente un entorno de producción, y también debido a que la naturaleza de una solución implementada puede evolucionar y cambiar, deberá continuar supervisando los sistemas implementados para identificar todas las áreas que requieran mantenimiento, servicio o puesta a punto.

## Despliegue de una implementación de producción

Una vez que la implementación piloto o de prueba de concepto pasa los criterios de prueba, estará preparado para desplegar la implementación en un entorno de producción. Normalmente, un entorno de producción se despliega por fases. Un despliegue por fases es especialmente importante para las implementaciones de gran tamaño que afectan a un importante número de usuarios.

La implementación por fases puede iniciarse con un pequeño conjunto de usuarios y con el tiempo ampliar la base de usuarios hasta que esté disponible para todos los usuarios. Una implementación por fases también se puede iniciar con un conjunto limitado de servicios y con el tiempo, introducir los servicios restantes por etapas. La introducción de los servicios por etapas puede ayudar a aislar, identificar y resolver los problemas que se puedan producir en un entorno de producción.

Despliegue de una implementación de producción

# Índice

## A

- Access Manager 55, 81
- acuerdos de nivel de servicios 33
  - que afectan al diseño de implementación 75
  - requisitos 50
- análisis de uso 38
  - que afectan al diseño de implementación 75
- Application Server 55
- aprobación del proyecto 72
- arquitectura de implementación 73
  - ejemplo 104
- arquitectura lógica
  - diseño 53
  - ejemplo de comunicaciones basadas en identidades 66
  - ejemplos 60
  - que afectan al diseño de implementación 75
- arquitectura tridimensional 52
- arquitecturas lógicas 51
- asistencia
  - Solaris 14
- aspectos relacionados con la migración 20
  - como limitación empresarial 34

## C

- cálculo de los requisitos de procesadores 76, 77
  - casos de uso 82
  - ejemplo 79
  - transacciones seguras 84

- Calendar Server 55, 81
- capa de cliente
  - modelo de arquitectura en varias capas 59
- capa de datos
  - modelo de arquitectura en varias capas 60
- capa de presentación
  - modelo de arquitectura en varias capas 59
- capa de servicios de negocios
  - modelo de arquitectura en varias capas 60
- capacidad latente 49
  - consideraciones de escalabilidad 97
- capas lógicas
  - modelo de arquitectura en varias capas 59
- casos de uso 40
  - cálculo de los requisitos de procesadores 82
  - ejemplo de comunicaciones basadas en identidades 66
  - Ejemplo de Messaging Server 62
  - que afectan al diseño de implementación 75
- ciclo de vida de la solución 22
  - etapa de análisis de negocios 23, 27
  - etapa de diseño de implementación 25, 72
  - etapa de diseño lógico 24, 51
  - etapa de implementación 25, 108
  - etapa de operaciones 26
  - etapa de requisitos técnicos 23, 37
- Communications Express 55
- comprobación
  - comprobaciones funcionales 110
  - pilotos y prototipos 110
  - pruebas de estrés 110
- comprobaciones funcionales 110

## Apartado D

- coste de propiedad 35
  - que afectan al diseño de implementación 76
- cultura corporativa 31

## D

- Directory Proxy Server 55, 58
- Directory Server 55, 62, 81
  - repetición de una réplica principal 94, 95
  - repetición de varias réplicas principales 94, 96
- diseño de arquitectura en varias capas 59
- diseño de implementación
  - acuerdos de nivel de servicios 75
  - análisis de uso 75
  - aprobación del proyecto 72
  - casos de uso 75
  - coste de propiedad 76
  - duplicación de servicios 77
  - factores 75
  - metodología 76
  - objetivos empresariales 76
  - requisitos de calidad de servicio 75
  - requisitos de procesadores 76
  - resultados 73
  - sobre 72
- diseño lógico
  - sobre 51
- disponibilidad
  - duplicación de servicios 89
  - ejemplos 92
  - equilibrado de carga 88
  - establecimiento de prioridades 44
  - optimización de recursos 103
  - recuperación tras error 88
  - requisitos de calidad de servicio 43
  - sistema de recuperación tras error N+1 90
  - sistemas redundantes horizontalmente 90
- DMZ
  - zona de acceso externo 70
- documentación 12
  - Guía de instalación 55, 109
  - visión general 12
  - Visión general técnica 19, 52, 53

- duplicación de servicios 77
  - ejemplo de Directory Server 94
  - estrategia de disponibilidad 89

## E

- ejemplo de comunicaciones basadas en identidades 65
  - cálculo de los requisitos de procesadores 79
  - casos de uso 66
- ejemplos
  - arquitectura de implementación 104
  - arquitectura lógica 60
  - Arquitectura lógica de Messaging Server 61
  - cálculo de los requisitos de procesadores 79
  - cálculo de requisitos de procesador para transacciones seguras 85
  - comunicaciones basadas en identidades 65
  - Directory Server 94
  - diseño de disponibilidad 92
  - duplicación de servicios 94
  - equilibrado de carga 90, 92
  - escalabilidad 98
  - recuperación tras error 94
  - repetición de una réplica principal 95
  - repetición de varias réplicas principales 96
  - zonas de acceso 69
- equilibrado de carga 88
  - ejemplo 90
- escalabilidad
  - cálculo del crecimiento previsto 46
  - ejemplo 98
  - estrategias 97
  - optimización de recursos 103
  - requisitos de calidad de servicio 45
- escenario de implementación 51, 70, 72
- especificaciones de implementación 73
- estrategias de disponibilidad
  - determinación 88
- etapa de análisis de negocios 23
  - sobre 27
- etapa de diseño de implementación 25
- etapa de diseño lógico 24

etapa de implementación 25, 109  
 desarrollo de pilotos y prototipos 109  
 sobre 108

etapa de operaciones 26

etapa de requisitos técnicos 23  
 análisis de uso 38  
 casos de uso 40  
 requisitos de calidad de servicio 41  
 sobre 37

## F

facilidad de mantenimiento  
 optimización de recursos 103  
 requisitos de calidad de servicio 49

## G

gestión de riesgos 104  
 diseño de implementación 77

## I

identificación de los cuellos de botella  
 diseño de implementación 77

implementaciones 20:80 19  
 etapa de implementación 110

implementaciones 80:20 19, 109

instalación de Java Enterprise System 108

Instant Messaging 55

## J

Java Enterprise System  
 arquitectura tridimensional 52  
 aspectos relacionados con la migración 20  
 componentes 53

componentes de acceso 58

despliegue de una implementación de  
 producción 111

implementaciones 20:80 19

implementaciones 80:20 19

instalación 108

relaciones de dependencia de los  
 componentes 54

servicios 19

servicios del sistema 18

servicios personalizados 19

sobre 17

## L

limitaciones empresariales 34  
 aspectos relacionados con la migración 34  
 coste de propiedad 35  
 limitaciones presupuestarias 35  
 obligaciones de programación 34

limitaciones presupuestarias 35

## M

Manual de ejecución 74

Message Queue 55

Messaging Server 55  
 Almacén de mensajes (STR) 81  
 casos de uso 62  
 ejemplo de arquitecturas lógicas 61  
 ejemplo de equilibrado de carga 92  
 Message Multiplexor (MMP) 58, 62, 81  
 Message Store (STR) 57, 62  
 Message Transfer Agent (MTA) 57, 61  
 Messenger Express Multiplexor (MEM) 58  
 servicios distintos lógicamente 57

## O

- objetivos empresariales
  - definición 28
  - que afectan al diseño de implementación 76
- obligaciones de programación 34
- optimización
  - acceso al disco 101
  - utilización de los recursos 102
- optimización de recursos
  - diseño de implementación 77

## P

- patrones de uso 31
- pilotos 109
  - comprobación 110
- plan de despliegue 74
- plan de formación 74
- plan de gestión de usuarios 73
- plan de instalación 73
- plan de migración 73
- plan de operaciones (Manual de ejecución) 74
- plan de prueba 74
- plan de recuperación tras desastre 74
- planes de implementación 73
- planificación de implementación
  - ciclo de vida de la solución 22
  - planteamiento progresivo 33
  - sobre 21
- Portal Server 55, 58
  - Mobile Access 58
  - Secure Remote Access 55, 58
- prototipos 109
  - comprobación 110
- pruebas de estrés 110

## Q

- QoS (requisitos de calidad de servicio) 41

## R

- recuperación tras error 88
  - ejemplo 94
  - software de Sun Cluster 92
- relaciones de dependencia de los componentes 54
  - compatibilidad con contenedor Web 57
- rendimiento
  - identificación de los cuellos de botella 99
  - optimización de recursos 103
  - requisitos de calidad de servicio 42
- repetición de una réplica principal 94
  - ejemplo 95
- repetición de varias réplicas principales 94
  - ejemplo 96
- requisitos de calidad de servicio 41, 49
  - papel en el diseño de implementación 72
  - que afectan al diseño de implementación 75
- requisitos de nivel de servicio 50
- requisitos de procesadores
  - cálculo 77
- requisitos empresariales
  - acuerdos de nivel de servicios 33
  - comprensión de los usuarios 29
  - cultura corporativa 31
  - definición 28
  - objetivos de seguridad 32
  - objetivos empresariales 28
  - patrones de uso 31
  - requisitos obligatorios 32
  - requisitos operativos 30
- requisitos obligatorios 32
- requisitos operativos 30
- requisitos técnicos
  - requisitos de nivel de servicio 50
- requisitos técnicos.
  - capacidad latente 49
  - disponibilidad 43
  - escalabilidad 45
  - facilidad de mantenimiento 49
  - rendimiento 42
  - seguridad 47

**S**

- seguridad
  - cálculo de los requisitos de procesadores 76
  - optimización de recursos 103
  - requisitos de calidad de servicio 47
- sistema de recuperación tras error N+1 90
- sistemas redundantes horizontalmente 90
- sistemas tolerantes a fallos 44
- SLA 33
- software de Sun Cluster 92
  - ejemplo de recuperación tras error 94
- Solaris
  - asistencia 14
  - parches 14

**W**

- Web Server 55, 81

**Z**

- zona de acceso externo (DMZ) 70
- zona de acceso interno (intranet) 70
- zona de acceso seguro 70
- zonas de acceso 69

