

ADMINISTRACIÓN DE APLICACIONES EN TOMCAT

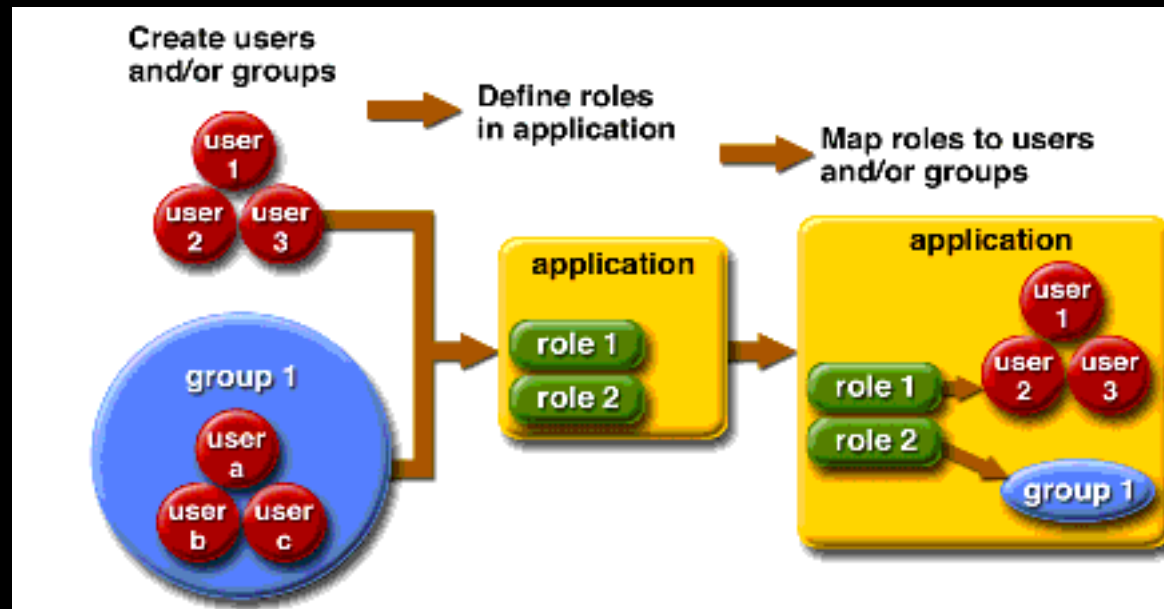


ADMINISTRACIÓN DE APLICACIONES EN TOMCAT

- Roles de admin y manager
- Uso del manager desde la URL
- Introducción a JMX, habilitación y protección
- Uso del JMX proxy
- Uso de herramientas remotas JMX
- Administración y configuración JMX
- Gestión del heap de memoria de la JVM
- Técnicas de tuning y obtención de métricas
- Profiling del servidor
- Optimización del GC y HotSpot
- Optimización de la configuración de los conectores



Roles de Admin y manager



Roles de Admin y manager

- El modelo se construye sobre los conceptos de usuarios y roles
- La aplicación declara en su descriptor qué roles
- pueden acceder a qué recursos y los usuarios que dispongan de ese rol tendrán acceso
- Los Realm, se configuran en Tomcat (no en la aplicación) y así se pueden añadir y borrar usuarios dinámicamente



Roles de Admin y manager

- El modelo se construye sobre los conceptos de usuarios y roles
- La aplicación declara en su descriptor qué roles
- pueden acceder a qué recursos y los usuarios que dispongan de ese rol tendrán acceso
- Los Realm, se configuran en Tomcat (no en la aplicación) y así se pueden añadir y borrar usuarios dinámicamente
- Los usuarios y roles se almacenan en un fichero de texto editable (p.e. XML) que se carga en memoria



Roles de Admin y manager

Configuración de
<CATALINA_HOME>/conf/tomcat-users.xml

```
<tomcat-users>  
<role rolename="manager-gui"/>  
<user username="manager" password="contraseña" roles="manager-gui"/>  
</tomcat-users>
```



Roles de Admin y manager

Para una mayor seguridad sería conveniente cifrar la contraseña.

Por ejemplo con cifrado sha:

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        digest="SHA" resourceName="UserDatabase"/>
</Realm>
```

```
21:29 $ ./digest.sh -a SHA contraseña
contraseña:8c31b65bdec9f18b695d7318186fd1feed690d
```

```
<tomcat-users>
<role rolename="manager-gui"/>
<user username="manager" password="8c31b65bdec9f18b695d7318186fd1feed690d" roles="manager-gui"/>
</tomcat-users>
```



Uso del JMX proxy

Tomcat Manager contiene un **servlet** ligero llamado **JMXProxy**, que puede ser usado para enviar comando JMX y manda respuestas en formato URI.

Para introducir un comando lo haríamos de la siguiente forma:

<http://urlDeNuestraApp/manager/jmxproxy/?qry=comando>



Roles de Admin y manager

Perfiles para la aplicación Manager:

- **manager-gui** — Da acceso a la interfaz HTML.
- **manager-status** — Solo da acceso a la página “Server Status”.
- **manager-script** — Acceso en modo texto a las herramientas “tools-friendly” y a la página “Server Status”.
- **manager-jmx** — Da acceso a la interfaz **JMX proxy** y a la página “Server Status”.



Uso del manager desde la URL

Herramienta web que permite llevar a cabo tareas de administración:

- Desplegar aplicación web
- Listar aplicaciones desplegadas y sesiones activas
- Listar recursos JNDI
- Elaborar roles de seguridad
- Iniciar una aplicación detenida
- Detener una aplicación
- Replegar una aplicación
- Mostrar estadísticas de sesión

Uso del manager desde la URL

- Por defecto y por motivos de seguridad esta desactivado
- Hay que configurar un Realm que permita el acceso
- Por defecto se controla mediante un Realm que lee los contenidos del fichero <CATALINA_HOME>/conf/tomcat-users.xml
- Hay que activar un usuario con su password correspondiente y el rol manager
- Siempre que lo activemos en el servidor debemos restringir el acceso al localhost



Uso del manager desde la URL

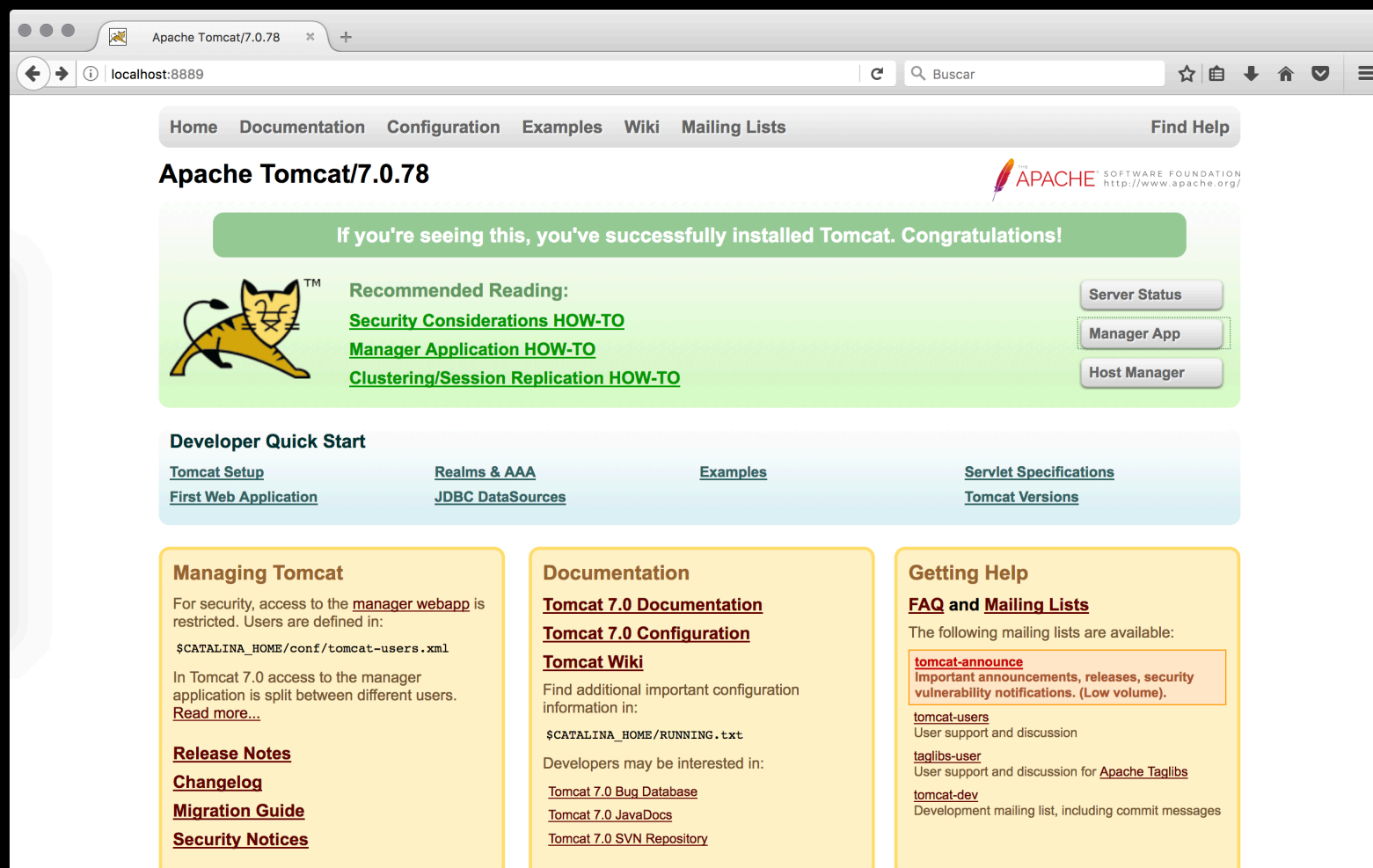
```
<Context antiResourceLocking="false" privileged="true" >  
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"  
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />  
</Context>
```

```
<Context antiResourceLocking="false" privileged="true" >  
<Valve className="org.apache.catalina.valves.RemoteAddrValve"  
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|192\.168\.1\.\d+" />  
</Context>
```

```
# ssh irespaldiza@192.168.0.111 -f -L 8889:192.168.0.111:8889 -N
```




Uso del manager desde la URL



The screenshot shows the Apache Tomcat/7.0.78 Manager App interface in a web browser. The browser's address bar shows 'localhost:8889'. The page has a navigation bar with links: Home, Documentation, Configuration, Examples, Wiki, Mailing Lists, and a 'Find Help' button. The main heading is 'Apache Tomcat/7.0.78'. A green banner states: 'If you're seeing this, you've successfully installed Tomcat. Congratulations!'. Below this, there is a 'Recommended Reading' section with links to 'Security Considerations HOW-TO', 'Manager Application HOW-TO', and 'Clustering/Session Replication HOW-TO'. To the right of these links are three buttons: 'Server Status', 'Manager App', and 'Host Manager'. The 'Manager App' button is highlighted with a dashed border. Below the recommended reading is a 'Developer Quick Start' section with links to 'Tomcat Setup', 'First Web Application', 'Realms & AAA', 'JDBC DataSources', 'Examples', 'Servlet Specifications', and 'Tomcat Versions'. The page is divided into three main columns: 'Managing Tomcat', 'Documentation', and 'Getting Help'. The 'Managing Tomcat' column contains text about security and access to the manager webapp, a link to 'Read more...', and links to 'Release Notes', 'Changelog', 'Migration Guide', and 'Security Notices'. The 'Documentation' column contains links to 'Tomcat 7.0 Documentation', 'Tomcat 7.0 Configuration', 'Tomcat Wiki', and a list of additional configuration information including 'Tomcat 7.0 Bug Database', 'Tomcat 7.0 JavaDocs', and 'Tomcat 7.0 SVN Repository'. The 'Getting Help' column contains links to 'FAQ and Mailing Lists' and a list of available mailing lists: 'tomcat-announce', 'tomcat-users', 'taglibs-user', and 'tomcat-dev'.

Apache Tomcat/7.0.78

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 **Recommended Reading:**

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

[Server Status](#)

[Manager App](#)

[Host Manager](#)

Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

[Security Notices](#)

Documentation

[Tomcat 7.0 Documentation](#)

[Tomcat 7.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

`$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

- [Tomcat 7.0 Bug Database](#)
- [Tomcat 7.0 JavaDocs](#)
- [Tomcat 7.0 SVN Repository](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:



- [tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)
User support and discussion
- [taglibs-user](#)
User support and discussion for [Apache Taglibs](#)
- [tomcat-dev](#)
Development mailing list, including commit messages



Uso del manager desde la URL

- Es una aplicación web como cualquier otra y por tanto tiene un descriptor de despliegue propio.
- En su archivo web.xml se pueden modificar los criterios de seguridad:
 - Operaciones permitidas y roles administrativos
 - Tipo de autenticación

Uso del manager desde la URL



Server Status

Manager
[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Complete Server Status](#)

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.21	1.6.0_25-b06	Sun Microsystems Inc.	Linux	2.6.32-220.17.1.el6.x86_64	amd64	konosys	192.168.16.28

OS

Physical memory: 2006.77 MB Available memory: 116.71 MB Total page file: 2015.99 MB Free page file: 1989.64 MB Memory load: 95
Process kernel time: 0.0 s Process user time: 0.0 s

JVM

Free memory: 417.53 MB Total memory: 657.81 MB Max memory: 910.25 MB

"ajp-apr-8009"

Max threads: 200 Current thread count: 10 Current thread busy: 1 Kept alive sockets count: 19
Max processing time: 13532 ms Processing time: 37.003 s Request count: 1177 Error count: 577 Bytes received: 0.00 MB Bytes sent: 1.86 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request
R	?	?	?	?	?	?
S	2 ms	0 KB	0 KB	2.137.166.81	www.konosys.es	GET /manager/status?org.apache.catalina.filters.CSRF_NONCE=D492BCFE06EA0F684D5EE3EB0C00DD9E HTTP/1.1
R	?	?	?	?	?	?
R	?	?	?	?	?	?
R	?	?	?	?	?	?
R	?	?	?	?	?	?

P: Parse and prepare request S: Service F: Finishing R: Ready K: Keepalive



"http-apr-8080"

Max threads: 200 Current thread count: 10 Current thread busy: 0 Kept alive sockets count: 0
Max processing time: 17773 ms Processing time: 46.307 s Request count: 551 Error count: 0 Bytes received: 0.00 MB Bytes sent: 8.28 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request
R	?	?	?	?	?	?



Uso del manager desde la URL

Gestor de Aplicaciones Web de Tomcat

Mensaje:

OK

Gestor

[Listar Aplicaciones](#)
[Ayuda HTML de Gestor](#)
[Ayuda de Gestor](#)
[Estado de Servidor](#)

Aplicaciones					
Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	8	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/chat-portlet	Ninguno especificado	chat-portlet	true	6	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/google-maps-portlet	Ninguno especificado	google-maps-portlet	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/kaleo-web	Ninguno especificado	kaleo-web	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/knowledge-base-portlet	Ninguno especificado	knowledge-base-portlet	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/mail-portlet	Ninguno especificado	mail-portlet	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/manager	Ninguno especificado	Tomcat Manager Application	true	1	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>



Uso del manager desde la URL

- **Trayectoria:** ruta de la aplicación web
- **Nombre a mostrar:** <display-name> del DD
- **Ejecutándose:** true o false según sea
- **Sesiones:** número de sesiones activas.

Haciendo click sobre el número se obtiene más información estadística

- **Comandos:** arrancar, parar, recargar, replegar



Uso del manager desde la URL

Desplegar	
Desplegar directorio o archivo WAR localizado en servidor	
Trayectoria de Contexto (opcional):	<input type="text"/>
URL de archivo de Configuración XML:	<input type="text"/>
URL de WAR o Directorio:	<input type="text"/>
<input type="button" value="Desplegar"/>	
Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="button" value="Seleccionar archivo"/> No se ha seleccionado ningún archivo
<input type="button" value="Desplegar"/>	



Introducción a JMX, habilitación y protección

Monitorización

La monitorización es un aspecto clave para la administración de sistemas. Así podemos observar el rendimiento de un servidor para obtener estadísticas y/o reconfigurar algunos aspectos de nuestra aplicación o la configuración con los que mejorar el rendimiento.



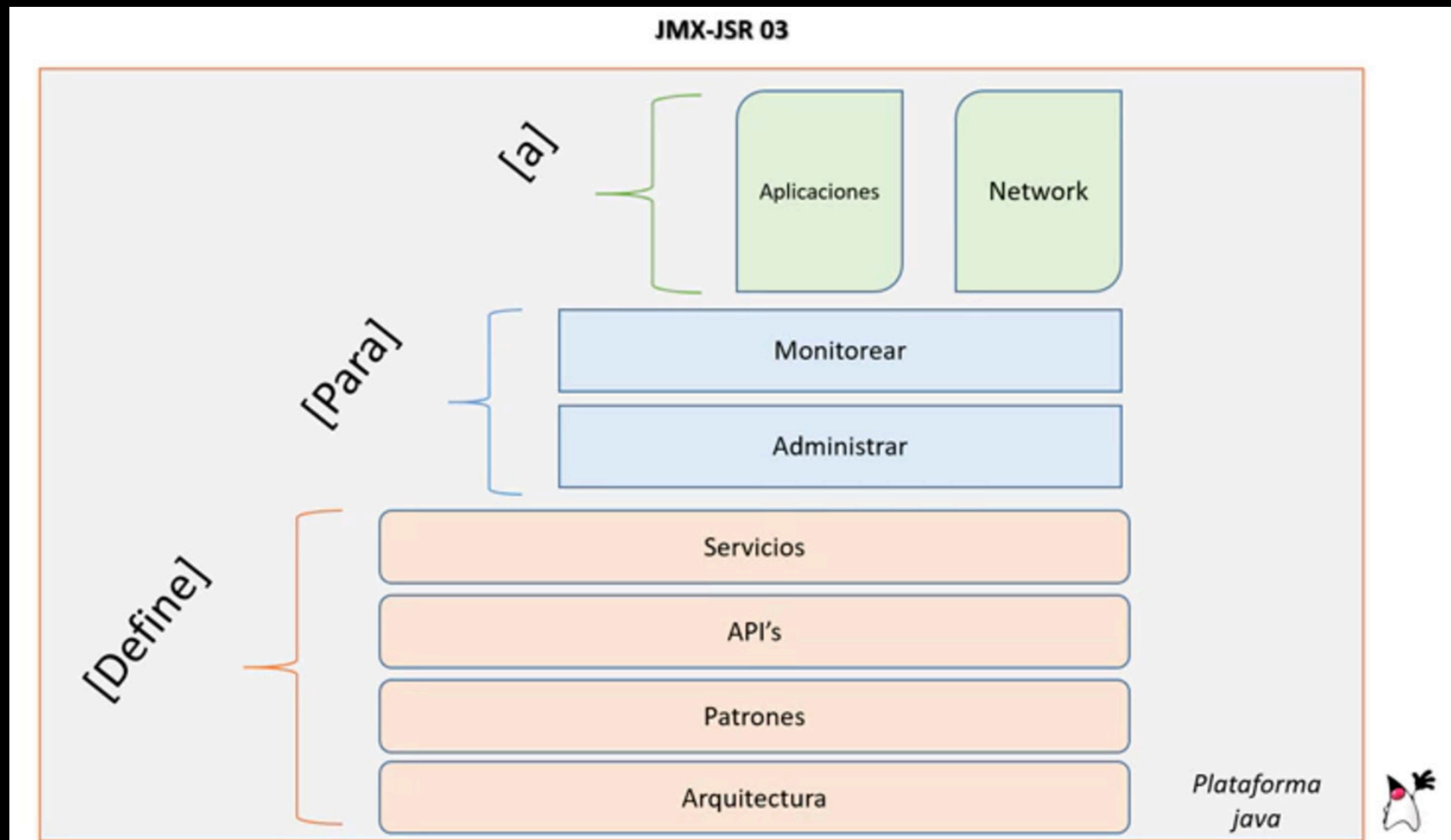
Introducción a JMX, habilitación y protección

JMX (Java Manager eXtension)

Se define como la arquitectura, los patrones de diseño, las APIs y los servicios para monitorear y administrar aplicaciones y redes en el lenguaje de programación Java.



Introducción a JMX, habilitación y protección



Introducción a JMX, habilitación y protección

Ventajas:

- Permite gestionar las aplicaciones Java de forma sencilla.
- Proporciona una arquitectura escalable.
- Se puede integrar con soluciones de supervisión preexistentes.
- Permite usar tecnologías Java estándar.
- Se definen **solo** las interfaces necesarias para la supervisión.



Introducción a JMX, habilitación y protección

Tenemos que añadir un nuevo script, llamado **setenv.sh**, a la carpeta bin, darle permisos de ejecución y dotarlo del siguiente contenido:

```
CATALINA_OPTS=-Dcom.sun.management.jmxremote  
-Dcom.sun.management.jmxremote.port=%my.jmx.port%  
-Dcom.sun.management.jmxremote.ssl=false  
-Dcom.sun.management.jmxremote.authenticate=false
```



Introducción a JMX, habilitación y protección

Si queremos protegerlo con contraseña:

```
CATALINA_OPTS=-Dcom.sun.management.jmxremote  
-Dcom.sun.management.jmxremote.port=%my.jmx.port%  
-Dcom.sun.management.jmxremote.ssl=false  
-Dcom.sun.management.jmxremote.authenticate=true  
-Dcom.sun.management.jmxremote.password.file=../conf/  
jmxremote.password  
-Dcom.sun.management.jmxremote.access.file=../conf/  
jmxremote.access
```



Introducción a JMX, habilitación y protección

Y editar los archivos:

`$CATALINA_BASE/conf/jmxremote.access`

`monitorRole readonly`

`controlRole readwrite`

`$CATALINA_BASE/conf/jmxremote.password:`

`monitorRole tomcat`

`controlRole tomcat`



Uso del JMX proxy

Tomcat Manager contiene un **servlet** ligero llamado **JMXProxy**, que puede ser usado para enviar comando JMX y manda respuestas en formato URI.

Para introducir un comando lo haríamos de la siguiente forma:

<http://urlDeNuestraApp/manager/jmxproxy/?qry=comando>



Uso del JMX proxy

on URL - *service:jmx:rmi:///jndi/rmi://localhost:8050/jmxrmi*

rtcut the very long URL syntax.

onnection port

ection user name.

ection password.

rnal connection reference. With this attribute you can configure more the one connection inside th

nd usage (for access analysis or debugging)

property of the given name **exists** in the current project.

property of the given name **not exists** in the current project.

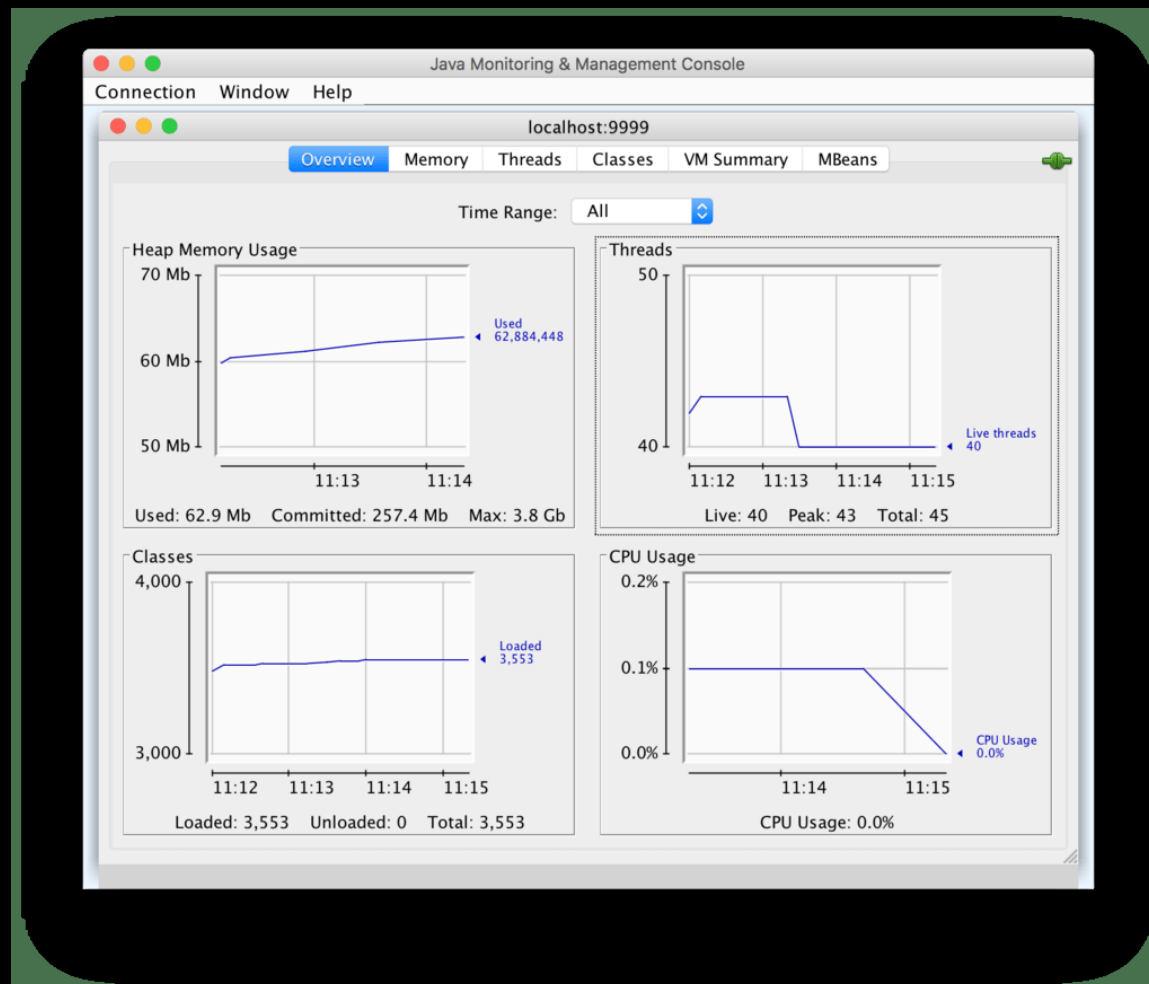


Uso de herramientas remotas JMX

Existen innumerables aplicaciones que se pueden conectar de forma remota a JMX. Usaremos Jconsole como ejemplo:



Uso de herramientas remotas JMX



Administración y configuración JMX

Attribute	Description	Default value
url	Set JMX connection URL - <i>service:jmx:rmi:///jndi/rmi://localhost:8050/jmxrmi</i>	
host	Set the host, shortcut the very long URL syntax.	localhost
port	Set the remote connection port	8050
username	remote JMX connection user name.	
password	remote JMX connection password.	
ref	Name of the internal connection reference. With this attribute you can configure more the one connection inside the same Ant project.	jmx.server
echo	Echo the command usage (for access analysis or debugging)	false
if	Only execute if a property of the given name exists in the current project.	
unless	Only execute if a property of the given name not exists in the current project.	



Administración y configuración JMX

- Ejemplo de una conexión JMX

```
jmx:open  
  host="${jmx.server.name}"  
  port="${jmx.server.port}"  
/>
```



Administración y configuración JMX

- Ejemplo de una conexión JMX desde una URL, con autenticación y que se almacena en otra referencia:

```
<jmx:open  
  url="service:jmx:rmi:///jndi/rmi://  
localhost:9024/jmxrmi"  
  ref="jmx.server.9024"  
  username="controlRole"  
  password="tomcat"  
>
```



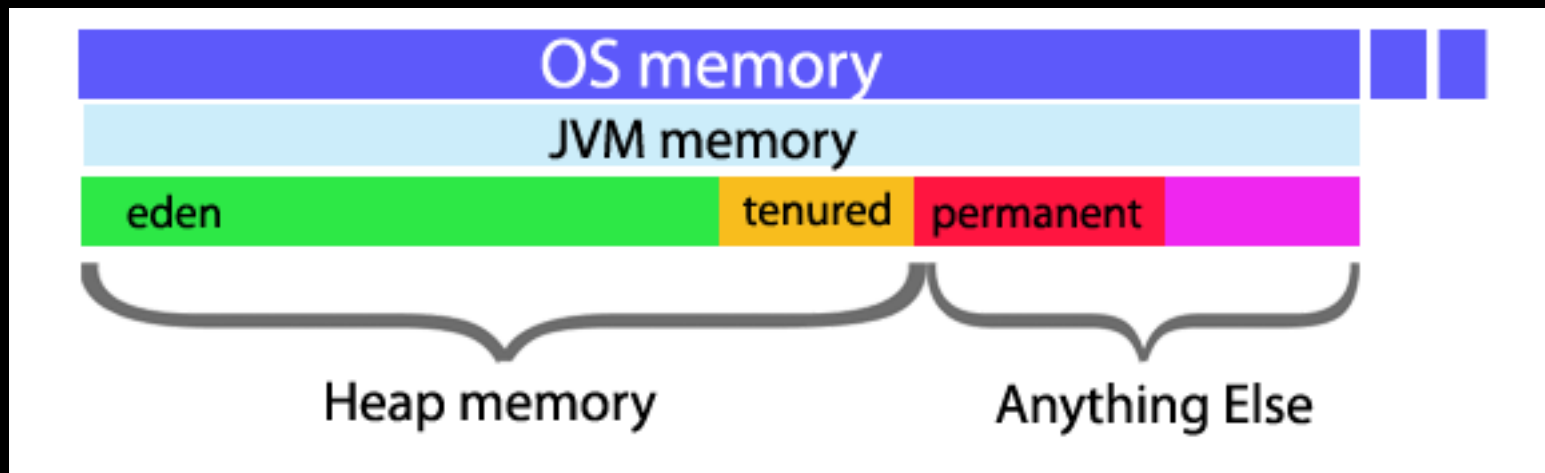
Administración y configuración JMX

- Ejemplo de una conexión JMX desde una URL, con autenticación y que se almacena en otra referencia pero con condiciones (que exista jmx.if y que no exista jmx.unless):

```
<jmx:open  
  url="service:jmx:rmi:///jndi/rmi://localhost:9024/jmxrmi"  
  ref="jmx.server.9024"  
  username="controlRole"  
  password="tomcat"  
  if="jmx.if"  
  unless="jmx.unless"  
>
```



Gestión del heap de memoria de la JVM



Gestión del heap de memoria de la JVM

Heap Memory

- **Xmx** representa el tamaño máximo de memoria que pudiera llegar a alcanzar la JVM en la que es ejecutada nuestra aplicación. Si la aplicación necesitase mas de la especificada en este parámetro nos encontraremos con el error **OutOfMemoryError**.
- **Xms** marca la cantidad de memoria asignada a la JVM cuando iniciamos la aplicación. Este valor suele ajustarse a la mitad o al valor exacto del especificado en Xmx.



Gestión del heap de memoria de la JVM

- **Xss** indica la cantidad de memoria disponible para cada hilo de ejecución. Un valor bajo en este parámetro puede provocar excepciones del tipo **java.lang.StackOverflowError**. Debemos de tener cuidado al aumentar el valor de este parámetro si encontramos la excepción anterior, pues un valor alto puede ser contraproducente, ya que podría impedir la creación de nuevos hilos de ejecución.
- **Xmn** controla la cantidad de memoria dinámica que puede consumir el área de memoria de nueva generación. Si definimos un valor bajo, el GC se ejecutará con mas frecuencia, mientras que un valor alto provocará que hilos que no son referenciados de forma persistente y que deberían de ser limpiados, pasen al área de memoria de vieja generación.



Técnicas de tuning y obtención de métricas

Para poder afinar nuestra aplicaciones y las configuraciones de nuestro servidor es necesario observar su comportamiento. Para poder realizar esta estadística podemos valernos de las herramientas anteriormente descritas así como de algún sistema para simular una prueba de carga. En mi caso recomiendo JMeter.



Optimización del GC y HotSpot

Optimización del recolector de basura (GC)

XX:+UseSerialGC: Activa el recolector de copia en serie. Utiliza un solo hilo para copiar los objetos que sobreviven en el espacio de Eden (Pool Eden Space) tras un GC, al pool espacio de supervivientes (Pool Survivor Space) y de este al área de memoria de vieja generación (Old Gen) cuando estima que el objeto lleva un tiempo suficiente en la young memory.

XX:+UseParallelGC: A diferencia del recolector de copia en serie (SerialGC), ParallelGC utiliza múltiples hilos en paralelo.



Optimización del GC y HotSpot

XX:+UseParallelOldGC: Se combina con el parámetro anterior para utilizar la paralelización en el Old Gen

XX:+UseAdaptiveSizePolicy: Se combina junto con los dos parámetros anteriores y sirve para ajustarse al tamaño. Mejora el rendimiento del GC.

XX:+UseParNewGC: En este caso el GC trabaja de forma paralela pero dispone de mejoras con respecto al tratamiento de los objetos almacenados en el área de memoria de vieja generación que hace que trabaje de forma concurrente. Este parámetro está obsoleto en versiones 8 y 9 de JDK.



Optimización del GC y HotSpot

Configuración de subprocesos

Hay dos tipos de hilos Java: green y nativo. Los subprocesos nativos son soportados por el sistema operativo, mientras que los subprocesos green se gestionan completamente dentro del espacio de la máquina virtual Java.

En general los procesos nativos tienen mejor rendimiento si bien los green rinden mejor en procesos de sincronización y la activación de subprocesos por lo que si nuestro proceso puede ser gestionado de las dos formas tendremos que decidir cual de ellos usar.



Optimización de la configuración de los conectores

Servidor de aplicaciones

- Tomcat requiere un servidor web que actúe de frontend (Apache, nginx u otro)
- El contenido estático es servido por el frontend
- Las peticiones a servlets y JSPs son redirigidas a Tomcat por el servidor web

Standalone

- No hay un servidor web que actúe de frontend
- Recibe peticiones en protocolos específicos como AJP que son enviados por el frontend
- Todos los contenidos son servidos por Tomcat
- Recibe peticiones HTTP

Optimización de la configuración de los conectores

Los conectores se definen en

`<CATALIMA_HOME>/conf/server.xml`

El único atributo obligatorio es el port

```
<Conector port="8080" protocol="HTTP/1.1"  
maxThreads="150" connectionTimeout="2000"  
redirectPort="8443" />
```



Optimización de la configuración de los conectores

Atributo	Descripción	Obligatorio	Valor por defecto
acceptCount	Tamaño de la cola cuando todos los threads que atienden peticiones están ocupados. Si llega una nueva petición con la cola llena, se denegará	No	10
address	La dirección IP a la que se vincula el conector. Si no se especifica, se vinculará a todas las direcciones que tenga el host	No	
allowTrace	Si es true permite el método TRACE http	No	false
compressibleMimeTypes	Lista separada por comas de tipos mime para los que se permite compresión	No	text/html,text/xml,text/plain
compression	Permite usar compresión GZIP http 1.1 para optimizar el ancho de banda. Valores posibles: off, on y force	No	off
connectionLinger	Número de milisegundos que las conexiones de sockets persistirán antes de cerrar la conexión. Un valor inferior a 0 significa que no se mantendrán	No	-1
connectionTimeout	Número de milisegundos que el conector esperará desde que un cliente hace una petición y solicita una URI	No	60000
disableOutputTimeout	Activa un temporizador para las subidas de datos durante la ejecución de un servlet	No	false
enableLookups	Si vale true permite hacer búsquedas en el DNS cuando un desarrollador hace una llamada <i>request.getRemoteHost()</i> , en caso contrario sólo devolverá la dirección IP	No	False
maxHttpHeaderSize	Controla el tamaño máximo de las cabeceras de petición y respuesta en bytes	No	4096 (4K)
maxKeepAliveRequest	Permite reutilizar la misma conexión para enviar más de una petición HTTP. El valor numérico indica el máximo número de peticiones permitidas antes de cerrar la conexión	No	100
maxPostSize	Tamaño máximo en bytes del paquete POST. Un 0 o un número negativo desactiva el máximo	No	2097152 (2MB)
maxSavePostSize	Al igual que el anterior pero en operaciones de certificado de cliente o autenticación de formulario	No	4096 (4K)



Optimización de la configuración de los conectores

Atributo	Descripción	Obligatorio	Valor por defecto
maxSpareThreads	Número máximo de threads inactivos que puede haber antes de que Tomcat empiece a detenerlos	No	50
minSpareThreads	Número mínimo de threads inactivos que puede haber antes de que Tomcat empiece a crear nuevos	No	4
maxThreads	Número máximo de threads que se pueden crear	No	200
noCompressionUserAgents	Permite indicar nombre de clientes (navegadores) para los que no se puede usar la compresión http	No	
protocol	En este caso HTTP/1.1	No	HTTP/1.1
proxyName	El nombre del proxy detrás del que potencialmente se puede ejecutar Tomcat	No	
proxyPort	El puerto del proxy detrás del que potencialmente se puede ejecutar Tomcat	No	
redirectPort	Si el conector soporta solo peticiones sin SSL y se envía una petición a un recurso SSL, el conector redirigirá la petición al puerto indicado	No	443
restrictedUserAgents	Lista de clientes (navegadores) separados por comas que no soportan HTTP/1.1	No	
scheme	El nombre del protocolo (http, https...)	No	http
secure	Si es true se trata de un conector SSL	No	false
server	Especifica la cabecera de servidor cuando enviamos la respuesta HTTP	No	Apache-Coyote/1.1
socketBuffer	Tamaño en bytes del búfer que se utiliza para almacenar los datos del socket. El valor -1 desactiva el búfer	No	9000
tcpNoDelay	Si vale true, activa la opción de socket de red TCP_NO_DELAY	No	true
threadPriority	Establece la prioridad de los threads de gestión de peticiones	No	NORM_PRIORITY
URIEncoding	La codificación de caracteres por defecto	No	ISO-8859-1
xpoweredBy	Si vale true, se incluye una cabecera X-Powered-By en las respuestas http	No	false

Optimización de la configuración de los conectores

Conector HTTP/1.1 con SSL

Utiliza los mismos atributos que el conector HTTP/1.1
(adicionalmente veremos otros más)

Debería tener el atributo `secure` a `true` y el `scheme` a `https`

```
<Conector port="8443" protocol="HTTP/1.1"  
maxThreads="150" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS" />
```



Optimización de la configuración de los conectores

Atributo	Descripción	Obligatorio	Valor por defecto
algorithm	Especifica el algoritmo de cifrado de certificado que se utilizará	No	SunX509
ciphers	Una lista separada por comas de cifrados de encriptación	No	
clientAuth	Si vale true el cliente ha de presentar un certificado válido. De todos modos, aunque valga false, si el recurso está protegido deberá presentar certificado válido	No	false
keystoreFile	Especifica la ruta al fichero del almacén de claves. Contiene las claves privadas y públicas del servidor en forma de certificado	No	.keystore
keystorePass	La contraseña que se requiere para acceder al keystoreFile	No	Changeit
keystoreType	El tipo de fichero de almacenamiento de claves	No	JKS
sslProtocol	Indica la versión del protocolo SSL que se utilizará	No	TLS



Optimización de la configuración de los conectores

¿Standalone o servidor de front-end?

La eficiencia del código nativo APR y las JVM modernas hace que la elección no sea trivial considerando sólo la eficiencia

Una configuración clúster con balanceo de carga obliga a tener un front-end

El soporte de seguridad de Apache es significativamente mejor que el de Tomcat

Si la estrategia web incluye otros mecanismos dinámicos (PHP, Perl, Python, ASP...) requeriremos un servidor de front-end especializado



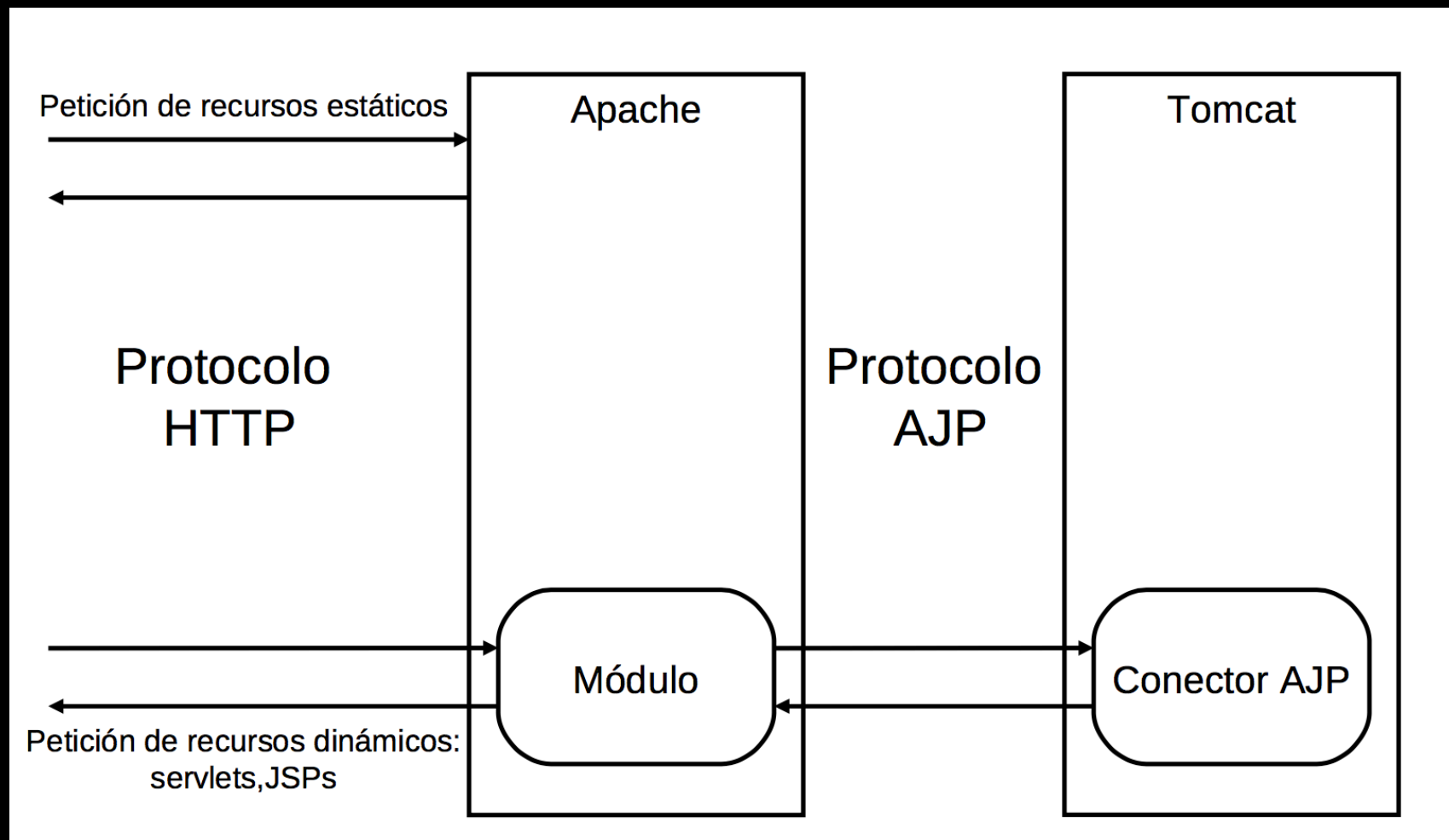
Optimización de la configuración de los conectores

Configuración front-end con Apache

- Para implementar la comunicación se requieren módulos especializados en ambos extremos
- Estos módulos se comunican con el protocolo AJP
- En el lado de Apache son módulos escritos en C/C++
- En el lado Tomcat son conectores AJP escritos en Java



Optimización de la configuración de los conectores



Optimización de la configuración de los conectores

Módulos disponibles para Apache

Dos opciones

- `mod_jk`: el módulo “tradicional”
- `mod_proxy`: módulo estándar de Apache que en sus últimas versiones soporta AJP

¿Cuál es la mejor opción?

El esfuerzo de desarrollo se ha centrado en `mod_proxy` y es el estándar de Apache

Suele ser más fácil de configurar



Optimización de la configuración de los conectores

mod_proxy: Configuración de Apache

Añadir la directiva correspondiente en httpd.conf:

LoadModule proxy_module modules/mod_proxy.so

LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

LoadModule proxy_balancer_module modules/mod_proxy_balancer.so



Optimización de la configuración de los conectores

mod_proxy: Configuración de Apache

Editar la configuración de proxy en http.conf, p.e.:

```
ProxyRequests Off ProxyPreserveHost On <Proxy *>  
Order deny,allow  
Allow from all </Proxy>  
ProxyPass /examples/jsp ajp://localhost:8009/examples/  
jsp ProxyPassReverse /examples/jsp ajp://localhost:8009/  
examples/jsp <Location /examples/jsp>  
Order allow,deny  
Allow from all  
</Location>
```



Optimización de la configuración de los conectores

En el lado de Tomcat:

las peticiones AJP se reciben en un conector AJP independiente del módulo que esté usando Apache

```
<Connector port="8009" protocol="AJP/1.3"
redirectPort="8443" />
```

