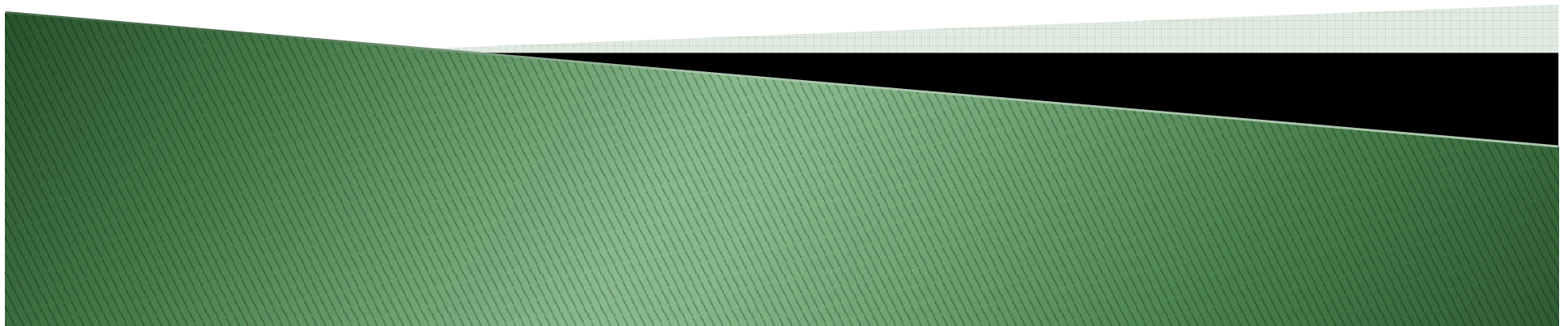


# Unidad 8

## Integración de *Apache* y *Tomcat*

Despliegue de aplicaciones web



# Índice

---

- ▶ Introducción.
- ▶ Modos de funcionamiento de *Tomcat*.
- ▶ Razones para la integración.
- ▶ ¿Cuándo integrar *Apache* y *Tomcat*?
- ▶ Conectores.
  - Introducción
  - *HTTP Connector*.
  - *AJP Connector*.
  - *ARP Native*.

# Índice

---

- ▶ Configuración
  - Introducción.
  - Posibilidades.
  - mod\_proxy.
  - mod\_jk.
- ▶ Bibliografía.

# Introducción

---

## ► *Apache HTTP Server*

- <http://httpd.apache.org>.
- Servidor Web potente, flexible y ajustado al HTTP/1.1.



- Modular.
  - Altamente configurable y extensible.
  - Ofrece su propio API (*Application Programming Interface*).

# Introducción

---

## ► *Apache Tomcat*

- <http://tomcat.apache.org/>
- Servidor que implementa un contenedor web J2EE (Servlet/JSP).
- Proporciona también un servidor HTTP (escrito en Java).



# Modos de funcionamiento de *Tomcat*

---

## ► *Standalone*

- Gestiona todas las peticiones HTTP.
- Sirve todos los contenidos (estáticos y *Servlets/JSPs*).

## ► Servidor *Servlets/JSPs*

- Se integra con un servidor Web (*Apache, IIS, ...*) (que actúa como *frontend*) que recibe las peticiones HTTP y se encarga de servir el contenido estático.
- Las peticiones a *Servlets* y *JSPs* son redirigidas y servidas por *Tomcat*.

# Razones para la integración

---

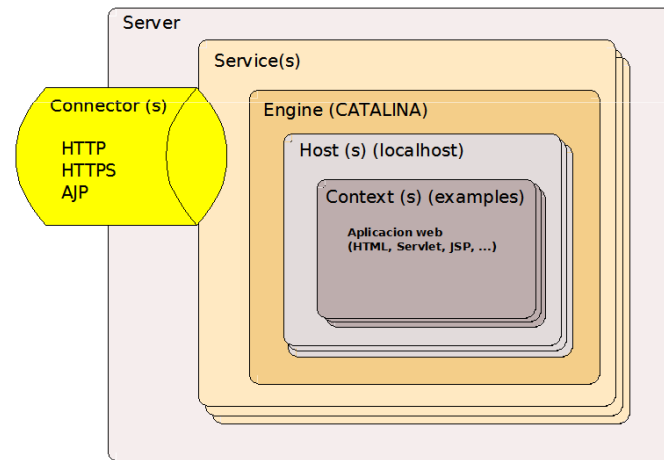
- ▶ Rendimiento
  - *Apache* sirve contenido estático (html, imágenes, vídeos, ...) más rápido que *Tomcat*.
  - Aunque las últimas versiones de *Tomcat* el rendimiento del servidor ha aumentado considerablemente.
- ▶ Opciones de configuración y seguridad *de Apache*.
  - Apache ofrece muchas más opciones de configuración que *Tomcat* (cientos de módulos disponibles).
- ▶ *Apache* es muy robusto y escalable.

# Conectores

## Introducción

---

- ▶ Un conector (<Connector>) es una asociación con puerto IP para manejar las peticiones y las respuestas con los clientes.



- ▶ Web

- <http://tomcat.apache.org/tomcat-7.0-doc/connectors.html>



# Conectores

## Introducción

---

- ▶ Existen varios conectores para *Tomcat*.
- ▶ Los principales son.
  - *HTTP Connector*.
  - *AJP Connector*.

# Conectores

## *HTTP Connector*

---

- ▶ Conector configurado por defecto para manejar peticiones HTTP en el puerto 8080.
- ▶ Permite que *Tomcat* funcione un servidor *standalone*.

```
<Connector port="8080" protocol="HTTP/1.1"  
           connectionTimeout="20000"  
           URIEncoding="UTF-8"  
           redirectPort="8443" />
```

- ▶ Web
  - <http://tomcat.apache.org/tomcat-7.0-doc/config/http.html>

# Conectores

## *HTTP Connector*

---

- ▶ Existen varias implementaciones del conector con diferentes funcionalidades.
  - *Java Blocking connector BIO.*
  - *Java Nio Blocking connector NIO .*
  - *APR/native connector.*

	Java Blocking Connector BIO	Java Nio Blocking Connector NIO	APR/native Connector APR
Classname	Http11Protocol	Http11NioProtocol	Http11AprProtocol
Tomcat Version	3.x onwards	6.x onwards	5.5.x onwards
Support Polling	NO	YES	YES
Polling Size	N/A	maxConnections	maxConnections
Read HTTP Request	Blocking	Non Blocking	Blocking
Read HTTP Body	Blocking	Sim Blocking	Blocking
Write HTTP Response	Blocking	Sim Blocking	Blocking
Wait for next Request	Blocking	Non Blocking	Non Blocking
SSL Support	Java SSL	Java SSL	OpenSSL
SSL Handshake	Blocking	Non blocking	Blocking
Max Connections	maxConnections	maxConnections	maxConnections

# Conectores

## *AJP Connector*

---

- ▶ Conector para comunicar *Tomcat* con un conector web usando el protocolo AJP.
- ▶ Utilizado para integrar de forma “invisible” *Tomcat* y un servidor web y permitir que sirva contenido estático.
- ▶ Soporta la comunicación con dos conectores web
  - mod\_proxy
  - JK 1.2.x
- ▶ Web
  - <http://tomcat.apache.org/tomcat-7.0-doc/config/ajp.html>

# Conectores

## *AJP Connector*

---

- ▶ Existen varias implementaciones del conector con diferentes funcionalidades.
  - *Java Blocking connector BIO.*
  - *Java Nio Blocking connector NIO .*
  - *APR/native connector.*

	Java Blocking Connector BIO	Java Nio Blocking Connector NIO	APR/native Connector APR
Classname	AjpProtocol	AjpNioProtocol	AjpAprProtocol
Tomcat Version	3.x onwards	7.x onwards	5.5.x onwards
Support Polling	NO	YES	YES
Polling Size	N/A	maxConnections	maxConnections
Read Request Headers	Blocking	Sim Blocking	Blocking
Read Request Body	Blocking	Sim Blocking	Blocking
Write Response	Blocking	Sim Blocking	Blocking
Wait for next Request	Blocking	Non Blocking	Non Blocking
Max Connections	maxConnections	maxConnections	maxConnections

# Conectores

## *ARP Native*

---

- ▶ *APR (Apache Portable Runtime).*
- ▶ Librería de código escrita en C/C++ dependiente de la plataforma (Windows, Linux y Unix)

	Java Blocking Connector	Java Nio Blocking Connector	APR/native Connector
	BIO	NIO	APR
Classname	Http11Protocol	Http11NioProtocol	Http11AprProtocol
Tomcat Version	3.x onwards	6.x onwards	5.5.x onwards
Support Polling	NO	YES	YES
Polling Size	N/A	maxConnections	maxConnections
Read HTTP Request	Blocking	Non Blocking	Blocking
Read HTTP Body	Blocking	Sim Blocking	Blocking
Write HTTP Response	Blocking	Sim Blocking	Blocking
Wait for next Request	Blocking	Non Blocking	Non Blocking
SSL Support	Java SSL	Java SSL	OpenSSL
SSL Handshake	Blocking	Non blocking	Blocking
Max Connections	maxConnections	maxConnections	maxConnections

	Java Blocking Connector	Java Nio Blocking Connector	APR/native Connector
	BIO	NIO	APR
Classname	AjpProtocol	AjpNioProtocol	AjpAprProtocol
Tomcat Version	3.x onwards	7.x onwards	5.5.x onwards
Support Polling	NO	YES	YES
Polling Size	N/A	maxConnections	maxConnections
Read Request Headers	Blocking	Sim Blocking	Blocking
Read Request Body	Blocking	Sim Blocking	Blocking
Write Response	Blocking	Sim Blocking	Blocking
Wait for next Request	Blocking	Non Blocking	Non Blocking
Max Connections	maxConnections	maxConnections	maxConnections

# Conectores

## *ARP Native*

---

- ▶ Al utilizar código nativo incrementa la eficiencia y escalabilidad.
- ▶ Permite utilizar código nativo de *OpenSSL*.

# Configuración

## Introducción

---

### ► Acciones a realizar

- Configurar Apache para que redirija las peticiones seleccionadas a *Tomcat*.
- Configurar *Apache* y *Tomcat* para que se puedan comunicar entre si (usando un protocolo conocido por ambos: HTTP o AJP).
- Configurar *Tomcat* para que entienda las peticiones que llegan de *Apache*.



# Configuración

## Posibilidades

---

### ► *Tomcat*

- Se pueden usar los conectores HTTP y AJP dependiendo del protocolo a usar para comunicarse con *Apache*.

### ► *Apache*

- Se pueden utilizar dos módulos (escritos en C/C++) para redirigir las peticiones a *Tomcat*.
- **mod\_proxy**
  - Solo para las versiones de 2.0, 2.2 y 2.4 de *Apache*.
- **mod\_jdk**
  - Todas las versiones de *Apache*.

# Configuración

## mod\_proxy

---

- ▶ Módulo para habilitar *Apache* como proxy HTTP.
- ▶ Ofrece múltiples opciones de configuración.
- ▶ Dos opciones
  - (*Apache*) mod\_proxy\_http → (*Tomcat*) *HTTP Connector*
  - (*Apache*) mod\_proxy\_ajp → (*Tomcat*) *AJP Connector*
- ▶ Web
  - [http://httpd.apache.org/docs/2.2/mod/mod\\_proxy.html](http://httpd.apache.org/docs/2.2/mod/mod_proxy.html)

# Configuración mod\_proxy

---

- ▶ mod\_proxy\_http (1)
  - 1) Habilitar mod\_proxy en *Apache*

```
alumno@ServidorLinux01:/etc/apache2$ sudo a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
    service apache2 restart
alumno@ServidorLinux01:/etc/apache2$ sudo service apache2 stop
* Stopping web server apache2
... waiting [ OK ]
alumno@ServidorLinux01:/etc/apache2$ sudo service apache2 start
* Starting web server apache2 [ OK ]
alumno@ServidorLinux01:/etc/apache2$ _
```

- Fichero mods-enabled/proxy.conf

```
#ProxyRequests On
<Proxy *>
    AddDefaultCharset off
    Order deny,allow
#    Deny from all
#    #Allow from .example.com
</Proxy>
```

# Configuración mod\_proxy

---

## ► mod\_proxy\_http (2)

- 2) Configurar el conector HTTP de *Tomcat* para indicar el/los puerto/s de *Apache*.

```
<Connector port="8080" protocol="HTTP/1.1"  
           connectionTimeout="20000"  
           URIEncoding="UTF-8"  
           redirectPort="8443"  
           proxyPort="80"  
/>
```

```
<Connector  
  port="8443"  
  protocol="HTTP/1.1"  
  SSLEnabled="true"  
  maxThreads="150"  
  scheme="https" secure="true"  
  clientAuth="false"  
  sslProtocol="TLS"  
  keystoreFile="/var/lib/tomcat7/daw01keystore" keystorePass="despliegue"  
  keyAlias="tomcat" keyPass="despliegue"  
  proxyPort="443"  
/>
```

# Configuración mod\_proxy

## ► mod\_proxy\_http (3)

- 3) Configurar *Apache* para que redirija las peticiones seleccionas a *Tomcat*.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    ProxyPass /sample http://localhost:8080/sample
    ProxyPassReverse /sample http://localhost:8080/sample

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        DirectoryIndex despliegue.html
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    ProxyPreserveHost on
    ProxyPass /sample http://localhost:8080/sample
    ProxyPassReverse /sample http://localhost:8080/sample
    ProxyPass /curso http://localhost:8080/curso
    ProxyPassReverse /curso http://localhost:8080/curso
    ProxyPass /compras http://localhost:8080/compras
    ProxyPassReverse /compras http://localhost:8080/compras

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        DirectoryIndex despliegue.html
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

# Práctica

## ► Práctica 8.1

- Integración de *Apache* y *Tomcat* usando `mod_proxy`.

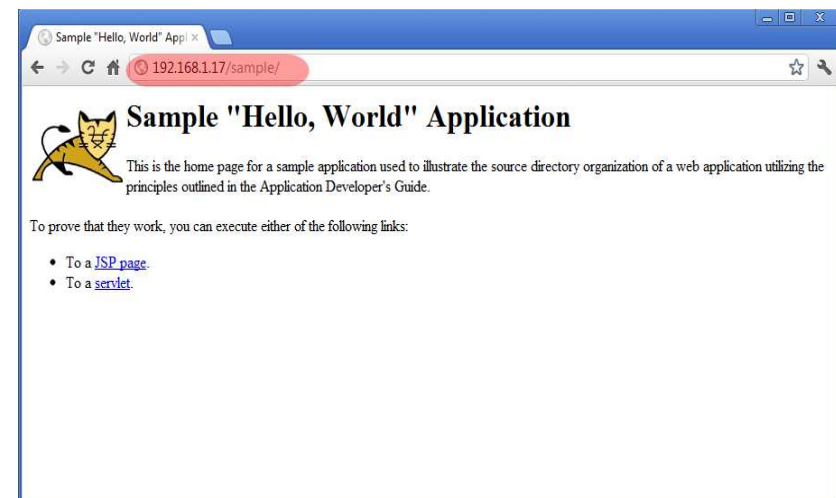
```
#ProxyRequests On
<Proxy *>
    AddDefaultCharset off
    Order deny,allow
    #    Deny from all
    #    #Allow from .example.com
</Proxy>
```

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    URIEncoding="UTF-8"
    redirectPort="8443"
    proxyPort="80"
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    ProxyPass /sample http://localhost:8080/sample
    ProxyPassReverse /sample http://localhost:8080/sample

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        DirectoryIndex despliegue.html
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```



# Configuración mod\_jk

---

- ▶ Módulo (implementación de *Tomcat*) para *Apache* que implementa el protocolo AJP para comunicarse con *Tomcat*.
- ▶ Workers
  - Instancias de *Tomcat* en ejecución.
- ▶ Web
  - <http://tomcat.apache.org/connectors-doc/>
  - [http://tomcat.apache.org/connectors-doc/webserver\\_howto/apache.html](http://tomcat.apache.org/connectors-doc/webserver_howto/apache.html)

# Configuración mod\_jk

- ▶ 1) Instalar y habilitar mod\_jk en Apache.
- ▶ 2) Configurar mod\_jk

```
# limitations under the License.

# Configuration Example for mod_jk
# used in combination with Apache 2.2.x

<IfModule jk_module>

    # We need a workers file exactly once
    # and in the global server
    JkWorkersFile /etc/libapache2-mod-jk/workers.properties

    # Our JK error log
    # You can (and should) use rotatelogs here
    JkLogFile /var/log/apache2/mod_jk.log

    # Our JK log level (trace,debug,info,warn,error)
    JkLogLevel info

    # Our JK shared memory file
    JkShmFile /var/log/apache2/jk-runtime-status
```

```
#
# workers.tomcat_home should point to the location where you
# installed tomcat. This is where you have your conf, webapps and lib
# directories.
#
workers.tomcat_home=/usr/share/tomcat7
```

```
#
#----- worker list -----
#
#
# The workers that your plugins should create and work with
#
worker.list=ajp13_worker

#
#----- ajp13_worker WORKER DEFINITION -----
#
#
# Defining a worker named ajp13_worker and of type ajp13
# Note that the name and the type do not have to match.
#
worker.ajp13_worker.port=8009
worker.ajp13_worker.host=localhost
worker.ajp13_worker.type=ajp13
```



# Configuración mod\_jk

---

- ▶ 3) Habilitar el conector AJP en *Tomcat*.

```
<!-- Define an AJP 1.3 Connector on port 8009 -->  
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

- ▶ 4) Configurar *Apache* para que redirija las peticiones seleccionas a *Tomcat*.

```
<VirtualHost *:80>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www  
  
    ProxyPreserveHost on  
    ProxyPass /sample http://localhost:8080/sample  
    ProxyPassReverse /sample http://localhost:8080/sample  
    ProxyPass /curso http://localhost:8080/curso  
    ProxyPassReverse /curso http://localhost:8080/curso  
    ProxyPass /compras http://localhost:8080/compras  
    ProxyPassReverse /compras http://localhost:8080/compras  
  
    JkMount /Calendar/ a.jp13_worker  
    JkMount /Calendar/* a.jp13_worker
```

# Práctica

## ► Práctica 8.2

- Integración de *Apache* y *Tomcat* usando mod\_jk.

```
#----- worker list -----  
#  
# The workers that your plugins should create and work with  
#  
worker.list=ajp13_worker  
#  
#----- ajp13_worker WORKER DEFINITION -----  
#  
#  
# Defining a worker named ajp13_worker and of type ajp13  
# Note that the name and the type do not have to match.  
#  
worker.ajp13_worker.port=8009  
worker.ajp13_worker.host=localhost  
worker.ajp13_worker.type=ajp13
```

```
<VirtualHost *:80>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www  
  
    ProxyPreserveHost on  
    ProxyPass /sample http://localhost:8080/sample  
    ProxyPassReverse /sample http://localhost:8080/sample  
    ProxyPass /curso http://localhost:8080/curso  
    ProxyPassReverse /curso http://localhost:8080/curso  
    ProxyPass /compras http://localhost:8080/compras  
    ProxyPassReverse /compras http://localhost:8080/compras  
  
    JkMount /Calendar/ ajp13_worker  
    JkMount /Calendar/* ajp13_worker
```

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
```

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

# Bibliografía

---

- ▶ <http://httpd.apache.org>.
- ▶ <http://tomcat.apache.org>
- ▶ <http://wiki.apache.org/tomcat>
- ▶ *Apache Tomcat 7. Aleksa Vukotic y James Goodwill. Editorial Apress.*
- ▶ <http://www.jguru.com/faq/java-tools/tomcat>