

# IMPLANTACIÓN DE ARQUITECTURAS WEB

Servidores de aplicaciones

# Índice

---

<b>1.</b>	<b>Servidores de aplicacións .....</b>	<b>3</b>
1.1	Recursos necesarios no servidor de aplicacións.....	3
<b>2.</b>	<b>Tomcat.....</b>	<b>4</b>
2.1	Arquitectura .....	4
2.2	Instalación .....	5
2.2.1	Instalación en Linux.....	5
2.2.2	Instalación en Windows.....	5
2.3	Despregamento de aplicacións web .....	6
2.4	Estrutura dunha aplicación web J2EE.....	6
2.5	Descritor de despregamento.....	7

# 1. Servidores de aplicacións

---

Un servidor de aplicacións é un paquete software que proporciona servizos ás aplicacións, como seguridade, soporte para transaccións, balanceo de carga e xestión de sistemas distribuídos.

Usualmente trátase dun dispositivo software que proporciona servizos de aplicación ás computadoras cliente. Un servidor de aplicacións xeralmente xestiona a maior parte (ou a totalidade) das funcións de lóxica de negocio e de acceso aos datos da aplicación. Os principais beneficios da aplicación da tecnoloxía de servidores de aplicación son a centralización e a diminución da complexidade no desenvolvemento de aplicacións.

O termo foi inicialmente empregado para servidores da plataforma Java Enterprise Edition (J2EE), sendo o máis popular Apache Tomcat, pero existindo tamén outros coma WebLogic, de Oracle, pero na actualidade tamén se estende a outras tecnoloxías. Por exemplo, co aumento da popularidade de .NET, Microsoft fala do seu produto Internet Information Server como un servidor de aplicacións. E tamén se poden atopar servidores de aplicación de código aberto e comerciais doutros provedores, por exemplo, Base4 Server (aplicacións .NET) e Zope (aplicacións escritas en Python).

## 1.1 Recursos necesarios no servidor de aplicacións

Para poder instalar o noso propio servidor de aplicacións precisamos, como mínimo, os seguintes recursos:

- Máquina cunha potencia capaz de atender as peticións que teña que procesar. É aconsellable que se trate dun servidor dedicado ou, como moito, compartido pero con funcións relacionadas, por exemplo, un servidor de transferencia de arquivos (FTP).
- Sistema operativo estable.
- Enderezo IP estático.
- Enderezo simbólico configurado nun servidor DNS para que dirixa as peticións ao enderezo IP correspondente.
- Conexión a Internet 24h.
- Software do servidor.

Para realizar as probas de funcionamento dos servidores, servirémonos dun contorno de virtualización con VirtualBox que conste de tres máquinas virtuais:

- Ubuntu.
- Windows.
- Máquina cliente, o sistema operativo déixase a elección do alumno. Se os recursos físicos do ordenador empregado non son suficientes para a execución de tres máquinas virtuais a un tempo, sería posible empregar a máquina real como cliente ou incluso o servidor no que non se estean facendo as probas.

Salvo que se indique outra cousa, para as prácticas empregaremos o modo de rede “Rede interna” coas seguintes IP:

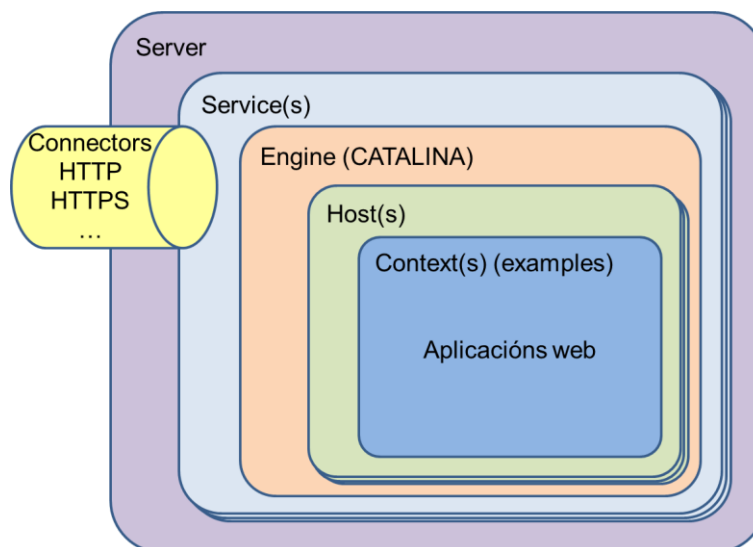
Máquina	IP
Ubuntu	192.168.0.1
Windows Server	192.168.0.2
Máquina cliente	192.168.0.3

## 2. Tomcat

Apache Tomcat (tamén chamado Jakarta Tomcat ou simplemente Tomcat) funciona como un colector de servlets desenvolvido baixo o proxecto Jakarta na Apache Software Foundation. Tomcat implementa as especificacións dos servlets e de JavaServer Pages (JSP).

Xestiona as solicitudes e respostas http e, ademais, é servidor de aplicacións ou colector de Servlets e JSP. Inclúe o compilador Jasper, que compila JSP convertíndoas en servlets.

### 2.1 Arquitectura



- O Server é o primeiro elemento contenedor.
- Un Conector (Connector) é unha asociación con porto IP para manexar as peticións e as respostas cos clientes.
- Un Servizo (Service) asocia un ou mais Connectors cun único Engine.
- Engine é un colector dun ou mais Hosts. É posible configurar Virtual Hosts. Recibe as peticións dos Conectores e as traslada ao host correspondente.
- Un Host define un servidor virtual (Virtual Host). Pode conter unha ou mais aplicacións web. Cada unha delas represéntase por un Context.
- Cada Context representa unha aplicación web (HTML, CSS, Servlets, JSP...) executándose dentro dun Host.

## 2.2 Instalación

### 2.2.1 Instalación en Linux

Para instalar Apache Tomcat en Linux teremos que ter instalado Java Development Kit e o propio servidor Tomcat. Ambos se atopan nos repositorios, pero instalando tomcat, de ser preciso, xa se instala openjdk automaticamente:

```
sudo apt-get install tomcat9
```

Tamén podemos instalar outros compoñentes:

```
sudo apt-get install tomcat9-examples
sudo apt-get install tomcat9-docs
sudo apt-get install tomcat9-admin
```

Para comprobar que o servidor está lanzado empregamos o comando `ps`, que amosa información sobre os procesos activos do sistema, coas opcións `-e` (para amosar todos os procesos) e `-f` (para que se amose o listado completo da información para cada proceso):

```
ps -ef | grep tomcat9
ou tamén
sudo service tomcat9 status
```

E para comprobar que está escoitando no porto 8080, o comando `netstat`, que amosa, entre outros, as conexións de rede, coas opcións `-l` (escoita), `-t` (TCP) e `-n` (numérico, para IP e portos):

```
netstat -ltn
```

Para facer unha proba de conexión ao servidor, no propio servidor, debe abrirse un navegador web e teclear:

```
http://localhost:8080
```

E aparecerá a páxina que se atope en `/var/lib/tomcat9/webapps/ROOT/index.html`, que, se non se fixeron cambios conterà, entre outras a mensaxe “It works!”.

Para comprobar que se poden establecer conexións desde outras máquinas, desde a máquina cliente tecleamos a IP do servidor Ubuntu:

```
http://192.168.0.1:8080
```

E obter os mesmos resultados que se comentaron para a proba en local.

### 2.2.2 Instalación en Windows

Para poder executalo, precisamos ter instalado o Java Development Kit, Unha vez instalado deberemos establecer dúas variables de contorno:

JAVA_HOME	C:\Program Files\Java\jdk...
JAVA_JRE	C:\Program Files\Java\jre...

Reiniciamos a máquina virtual do servidor para que os cambios teñan efecto.

A continuación, descargamos os arquivos precisos para a instalación de Apache Tomcat do enderezo <https://tomcat.apache.org/>

Vamos a instalar a versión 9 de Tomcat. Temos varias opcións de instalación, pero vamos a descargar unha distribución binaria para windows 64 bits. Descomprimos o arquivo e xa podemos acceder os diferentes elementos de Tomcat9.

Si queremos, podemos instalar o servizo tomcat9, mediante o seguinte comando (debemos atoparnos no directorio onde está o arquivo tomcat9.exe, é dicir \bin da carpeta onde está instalado Tomcat9):

```
tomcat9.exe -k install
```

Executamos o comando \bin\startup.bat e despois comprobaremos que o servidor está escoitando no porto 8080 co comando netstat, que permite monitorizar as conexións existentes entre a nosa computadora e o mundo exterior, coas opcións -a (para que as amose todas), -p TCP (para que filtre as do protocolo TCP) e -n (para que amose enderezos e portos de forma numérica):

```
netstat -a -p TCP -n
```

Comprobaremos que funciona en local <http://localhost:8080> e desde a máquina cliente do mesmo xeito que fixemos para Linux, tendo en conta unicamente que se debe configurar a devasa de Windows para que permita as conexións entrantes no porto 8080.

## 2.3 Despregamento de aplicacións web

Para realizar o despregamento dunha aplicación web en Tomcat basta con crear un directorio co nome da aplicación dentro do directorio webapps.

## 2.4 Estrutura dunha aplicación web J2EE

Unha aplicación web está organizada nunha estrutura xerárquica de directorios divididos en dúas partes:

- Un directorio privado `WEB-INF`, que contén os recursos que o cliente non pode descargar.
  - O descriptor de despregamento de aplicacións web (`web.xml`). Describe a estrutura da aplicación web.
  - Arquivos de descriptor de bibliotecas de arquivos.
  - `classes/`: directorio que contén as clases java compiladas que compoñen a aplicación (incluídos os servlets).
  - `lib/`: directorio que contén arquivos JAR de bibliotecas (bibliotecas de códigos e bibliotecas de programas de utilidade chamadas por clases do servidor).
- Un directorio público que contén os recursos públicos.

## 2.5 Descritor de despregamento

Un Descritor de Despregamento é un documento XML que describe as características de despregamento dunha aplicación, un módulo ou un compoñente. Calquera aplicación web ten que achegar un descritor de despregamento situado en `WEB-INF/web.xml`; no caso concreto de Tomcat o descritor `<TOMCAT_HOME>/conf/web.xml`.

O obxectivo do descritor de despregamento é especificar a configuración xeral da aplicación web tal como require o estándar J2EE. En particular:

- Os valores de parámetros de inicialización que se proporcionan para a aplicación web.
- As clases de servlet utilizadas pola aplicación web poden declararse e darlles nomes.
- Cada clase de servlet se relaciona con un ou máis patróns de URL: cando o colector de servlet recibe unha solicitude cuxo URL coincide cun patrón definido no arquivo `web.xml`, utilizarase o servlet correspondente para procesar a solicitude.
- Se é necesario, os valores de parámetros de inicialización que se proporcionan para cada servlet.
- Información de sesión (como o tempo de espera).
- A localización de bibliotecas de códigos personalizadas utilizadas polas páxinas JSP.

O descritor de despregamento é un ficheiro XML con etiqueta raíz `web-app` e que pode conter, entre outros, os seguintes elementos:

- `display-name`: Nome da aplicación (para tarefas de identificación, por exemplo, en Tomcat Manager aparece en "Nome a mostrar").
- `welcome-file-list`: Lista de ficheiros que se servirán, por orde, no caso no que non se indique ningún na URL. Cada un deles atópase nun elemento `welcome-file`.
- `session-config`
  - `session-timeout`: Tempo, en minutos, que tarda en expirar a sesión.
- `servlet`
  - `servlet-name`: Nome que lle damos ao servlet.
  - `servlet-class`: Clase Java na que se implementa o servlet.
- `servlet-mapping`
  - `servlet-name`: Nome que lle demos ao servlet.
  - `url-pattern`: Patrón de URL que haberá que usar para acceder.
- `security-constraint`
  - `web-resource-collection`
    - `web-resource-name`: Nome do recurso (para identificación).
    - `url-pattern`: Patrón de arquivos a protexer dentro da aplicación.
  - `auth-constraint`
    - `role-name`: Nome do rol definido en Tomcat para a autenticación.
- `login-config`
  - `auth-method`: Método de autenticación.
  - `realm-name`: Mensaxe que aparecerá na pantalla de identificación.