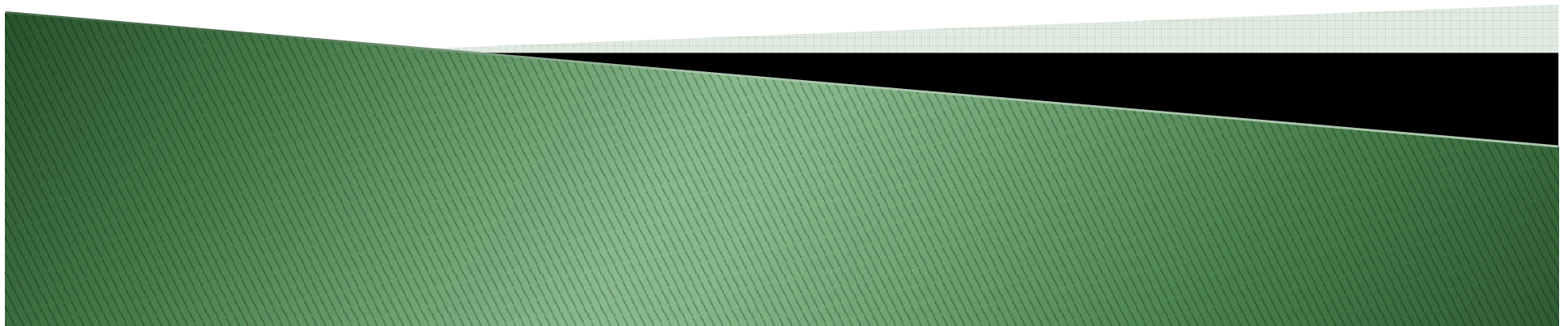


Unidad 7

Administración de *Tomcat*

Despliegue de aplicaciones web –
Instalación



Índice

- ▶ Introducción.
- ▶ Características.
- ▶ Versiones.
- ▶ Instalación.
- ▶ Inicio/parada del servidor
 - Introducción.
 - *Linux (Debian/Ubuntu).*
 - *Windows.*
- ▶ Arquitectura.
- ▶ Estructura de directorios.
- ▶ Ficheros de configuración.

Introducción

- ▶ **Apache** (<http://www.apache.org>)
 - Fundación que proporciona soporte a la comunidad con un numeroso conjunto de proyectos “*open source*”.
 - Desarrolladores y usuarios caracterizados por ser un grupo colaborativo, con el deseo de crear software de alta calidad.
 - <http://www.apache.org/foundation/>



The Apache Software Foundation

Community-led development since 1999.

Introducción

► Proyectos

- <http://projects.apache.org/>



The screenshot shows the Apache Software Foundation website. At the top, there is a logo of a feather and the text "The Apache Software Foundation" with the URL "http://www.apache.org/". Below this, there is a navigation menu on the left with links to "Home", "Indexes", "Feeds", "DOAP Files", "Documentation", and "References & Thanks". The "Indexes" link is selected, and it shows a list of links: "Alphabetical", "Categories", "Language", "PMC", "Project Listing", "Releases", and "Standards". The "Alphabetical Index" section is active, showing an alphabetical list of all projects indexed on the site. The list is organized by letter, with "A" currently selected. Under "A", there are several projects listed, each with a brief description and links to categories, languages, and PMCs. The projects listed are: Anakia, Apache .NET Ant Library, Apache Abdera, Apache Accumulo, Apache ActiveMQ, and Apache Ant.

The Apache Software Foundation
<http://www.apache.org/>

Alphabetical Index
An alphabetical list of all projects indexed on this site.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

- [Anakia](#)
Anakia is an XML transformation tool that uses JDOM and Velocity to transform XML documents into the format of your choice. It provides an alternative to using Ant's <style> task and XSL to process XML files.
Categories: [library](#)
Languages: [Java](#)
PMC: [Apache Velocity](#)
- [Apache .NET Ant Library](#)
This is a library of Ant tasks that help developing .NET software. It includes the "old".NET tasks like a C# compiler task but also comes with support for NUnit testing or running the popular NAnt or MSBuild build tools.
Categories: [build-management](#)
Languages: [Java](#)
PMC: [Apache Ant](#)
- [Apache Abdera](#)
The goal of the Apache Abdera project is to build a functionally-complete, high-performance implementation of the IETF Atom Syndication Format (RFC 4287) and Atom Publishing Protocol (RFC 5023) specifications.
Categories: [xml](#)
Languages: [Java](#)
PMC: [Apache Abdera](#)
- [Apache Accumulo](#)
The Apache Accumulo sorted, distributed key/value store is based on Google's BigTable design. It is built on top of Apache Hadoop, Zookeeper, and Thrift. It features a few novel improvements on the BigTable design in the form of cell-level access labels and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.
Categories: [database](#)
Languages: [Java](#)
PMC: [Apache Accumulo](#)
- [Apache ActiveMQ](#)
ActiveMQ is a fast and powerful Message Broker which supports many Cross Language Clients and Protocols and many advanced features while fully supporting JMS 1.1 and J2EE 1.4.
Categories: [network-server](#) [network-client](#)
Languages: [Java](#) [C](#) [C++](#) [Ruby](#) [Perl](#) [Python](#) [PHP](#) [C#](#)
PMC: [Apache ActiveMQ](#)
- [Apache Ant](#)
Apache Ant is a Java-based build tool.

Introducción

► *Apache Tomcat*

- <http://tomcat.apache.org/>



- [Apache Tobago](#)

The goal of Tobago is to provide the community with a well designed set of user interface components based on JSF.

Categories: [web-framework](#)

Languages: [Java](#)

PMC: [Apache MyFaces](#)

- [Apache Tomcat](#)

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the Java Community Process. Apache Tomcat is developed in an open and participatory environment and released under the Apache License version 2. Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Categories: [network-server](#)

Languages: [Java](#)

PMC: [Apache Tomcat](#)

- [Apache Torque](#)

Torque is an object-relational mapper for Java. In other words, Torque lets you access and manipulate data in a relational database using java objects. Unlike most other object-relational mappers, Torque does not use reflection to access user-provided classes, but it generates the necessary classes (including the Data Objects) from an XML schema describing the database layout (which can either be written by hand or generated from an existing database). The XML schema can also be used to generate and execute a SQL script which creates all the tables in the database.

Categories: [database](#)

Languages: [Java](#)

PMC: [Apache DB](#)

- [Apache Traffic Server](#)

Apache Traffic Server is fast, scalable and extensible HTTP/1.1 compliant caching proxy server. ATS can be used as a reverse, forward or even transparent HTTP proxy.

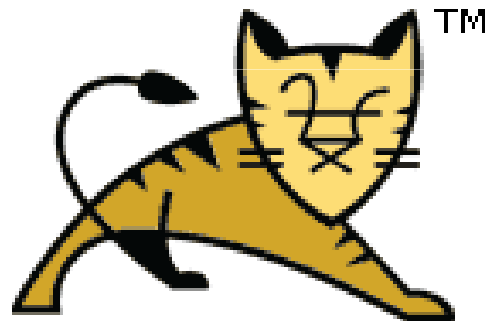
Categories: [http](#)

Languages: [C](#)

PMC: [Apache TrafficServer](#)

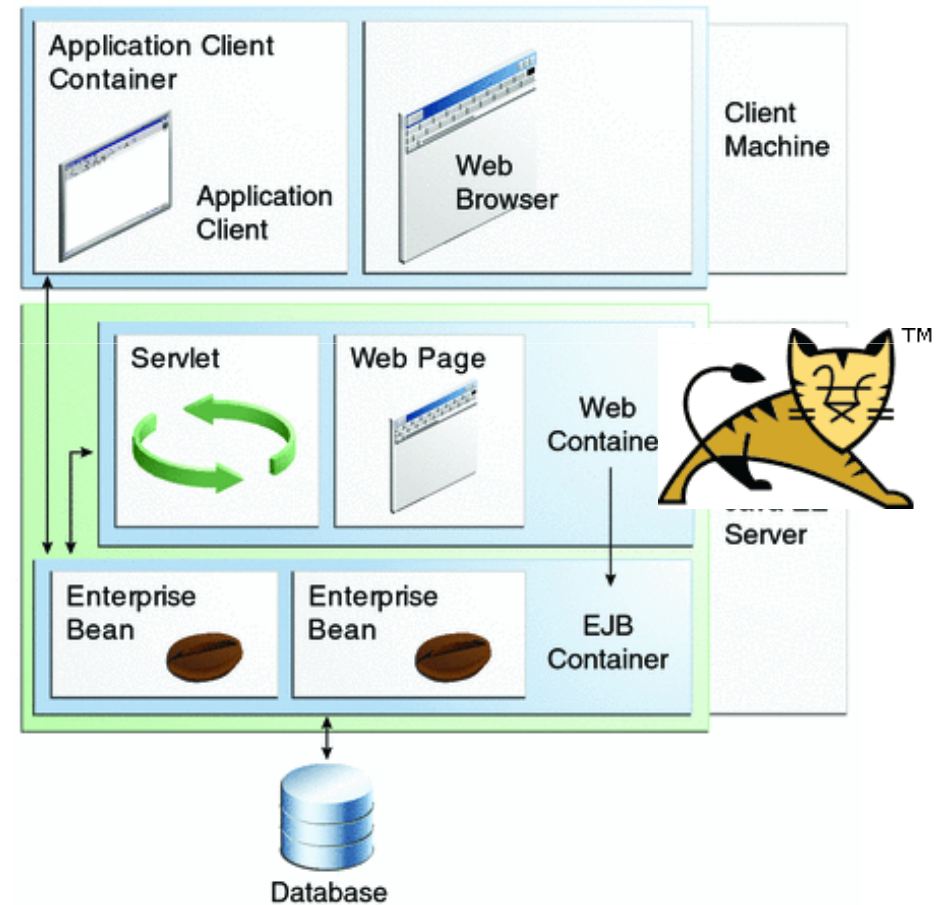
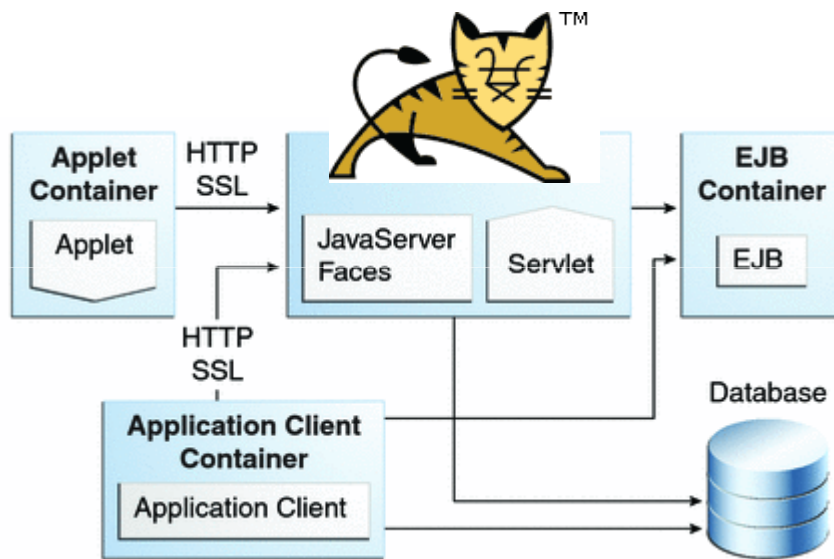
Características

- ▶ *Tomcat* es un servidor que implementa un contenedor web J2EE (Servlet/JSP).
- ▶ Proporciona también un servidor HTTP (escrito en Java).

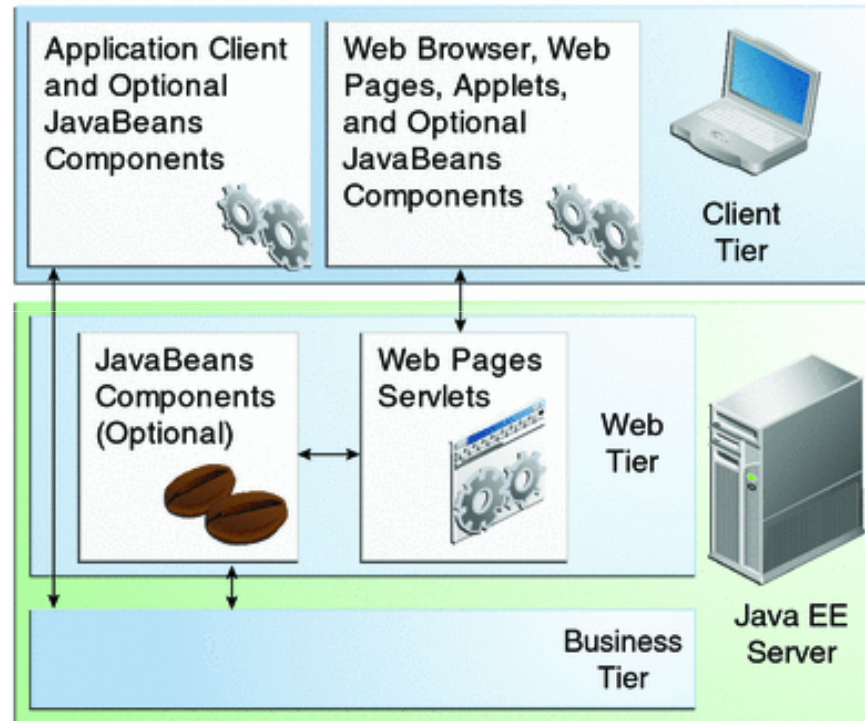
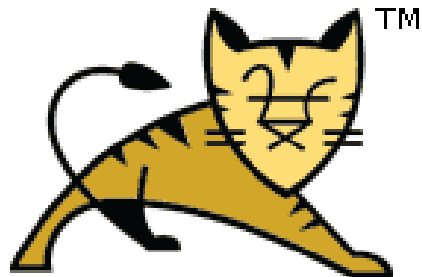


- ▶ Servidor “*Open Source*”.
- ▶ Multiplataforma: *Windows, Linux, Unix, ...*

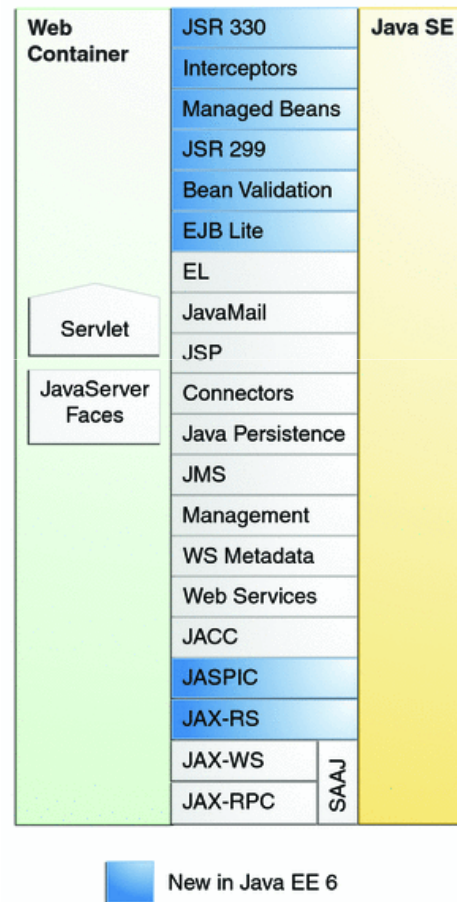
Características



Características



Características



Características

- ▶ Servidor estable y ligero.
- ▶ Más fácil de administrar y configurar que “servidores de aplicaciones completos”.

Versiones

► Versiones soportadas:

- Versión 7.X
- Versión 6.X.
 - [Tomcat 7.0](#)
 - [Tomcat 6.0](#)
 - [Tomcat 5.5](#)
- Versión 5.5.
- Obsoletas
 - Versión 5.0.
 - Versión 4.X.
 - Versión 3.X.

Versiones

- ▶ Las diferentes versiones de *Tomcat* ofrecen diferentes versiones de las especificaciones *Servlets* y *JSPs*.

Servlet/JSP Spec	Apache Tomcat version	Actual release revision	Minimum Java Version
3.0/2.2	7.0.x	7.0.27	1.6
2.5/2.1	6.0.x	6.0.35	1.5
2.4/2.0	5.5.x	5.5.35	1.4
2.3/1.2	4.1.x (archived)	4.1.40 (archived)	1.3
2.2/1.1	3.3.x (archived)	3.3.2 (archived)	1.1

- ▶ Web
 - <http://wiki.apache.org/tomcat/TomcatVersions>

Instalación

- ▶ *Tomcat* puede ser ejecutado tanto en entornos Windows como Unix/Linux.
- ▶ La única restricción es que la plataforma posea una JVM (*Java Virtual Machine*) instalada.
- ▶ Tipos de instalación
 - Descargar el código y ejecutarlo sobre la JVM.
 - Utilizar paquetes binarios adaptados para cada sistema operativo.
- ▶ Web
 - <http://tomcat.apache.org/tomcat-7.0-doc/setup.html>

Instalación

Binary Distributions

- Core:
 - [zip](#) ([pgp](#), [md5](#))
 - [tar.gz](#) ([pgp](#), [md5](#))
 - [32-bit Windows zip](#) ([pgp](#), [md5](#))
 - [64-bit Windows zip](#) ([pgp](#), [md5](#))
 - [64-bit Itanium Windows zip](#) ([pgp](#), [md5](#))
 - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [md5](#))
- Full documentation:
 - [tar.gz](#) ([pgp](#), [md5](#))
- Deployer:
 - [zip](#) ([pgp](#), [md5](#))
 - [tar.gz](#) ([pgp](#), [md5](#))
- Extras:
 - [JMX Remote jar](#) ([pgp](#), [md5](#))
 - [Web services jar](#) ([pgp](#), [md5](#))
 - [JULI adapters jar](#) ([pgp](#), [md5](#))
 - [JULI log4j jar](#) ([pgp](#), [md5](#))
- Embedded:
 - [tar.gz](#) ([pgp](#), [md5](#))
 - [zip](#) ([pgp](#), [md5](#))

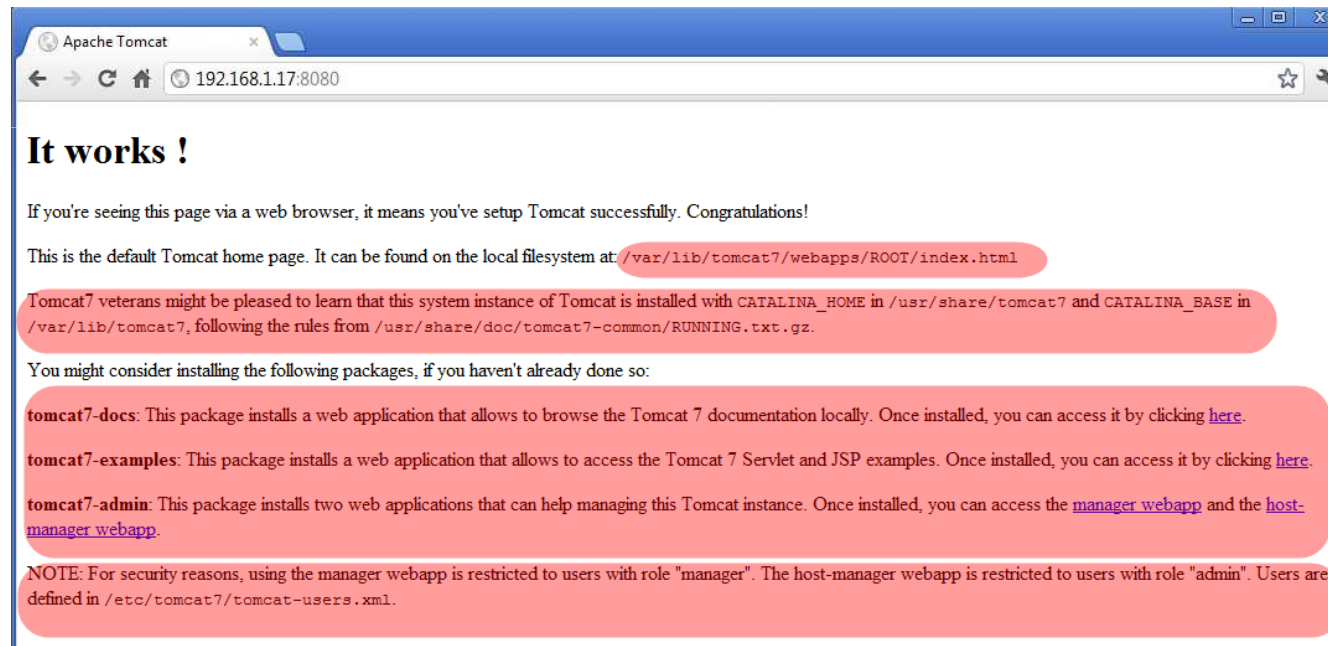
Source Code Distributions

- [tar.gz](#) ([pgp](#), [md5](#))
- [zip](#) ([pgp](#), [md5](#))

Práctica

► Práctica 7.1

- Instalación del servidor *Apache Tomcat 7.0* en *Linux*.



Inicio/parada del servidor

Introducción

- ▶ Cada que vez que cambien los ficheros de configuración.
- ▶ Hay que:
 - Parar/iniciar el servidor.
 - O indicarle que lea de nuevo los ficheros de configuración .
- ▶ Para que se apliquen los cambios.

Inicio/parada del servidor

Linux (Debian/Ubuntu)

► Inicio (varias posibilidades)

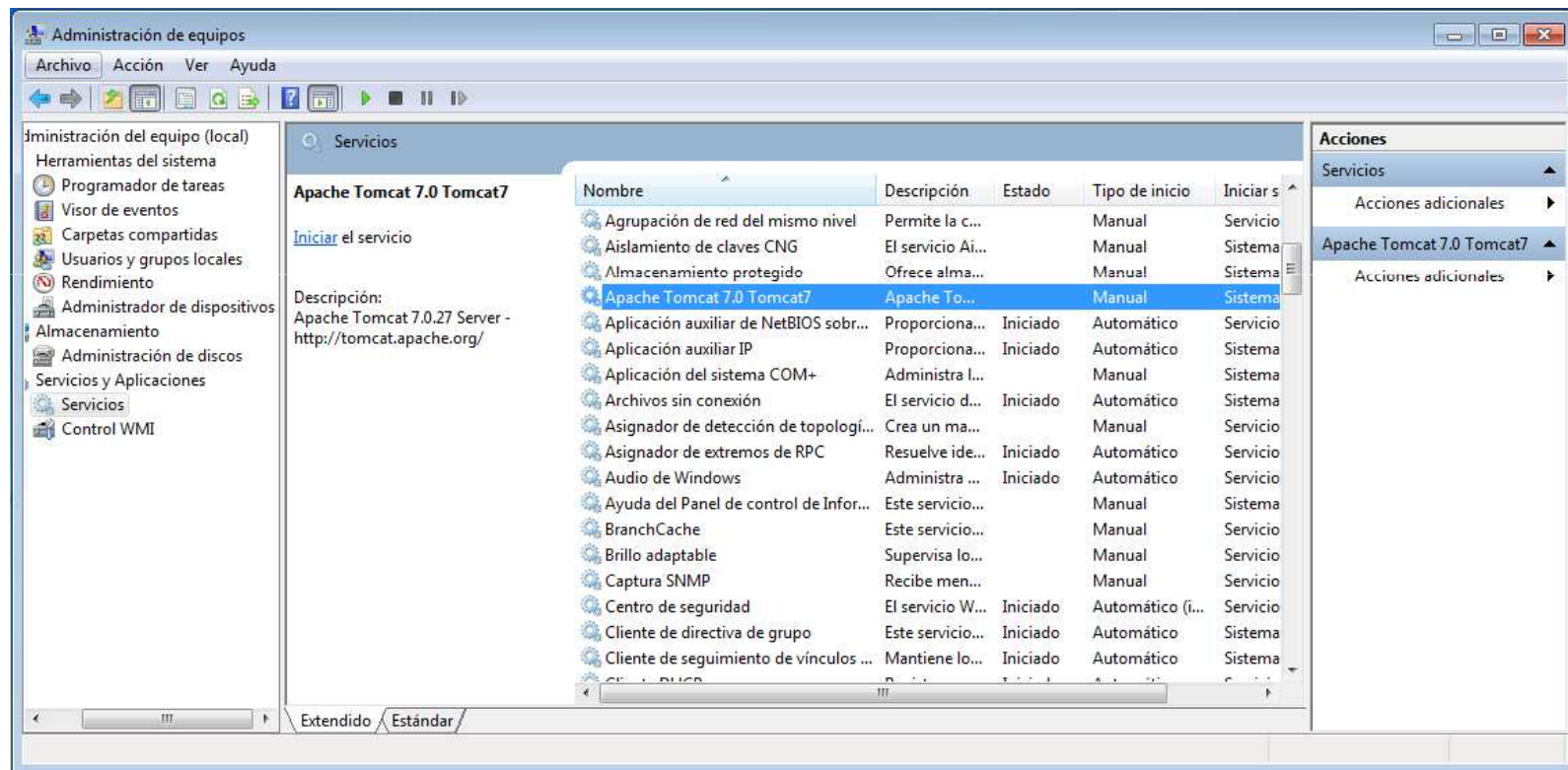
- `sudo service tomcat7 start`
- `sudo /etc/init.d/ tomcat7 start`
- `sudo -E $CATALINA_HOME/bin/starup.sh`
- `sudo -E $CATALINA_HOME/bin/catalina start`

► Parada (varias posibilidades)

- `sudo service tomcat7 stop`
- `sudo /etc/init.d/ tomcat7 stop`
- `sudo -E $CATALINA_HOME/bin/shutdown.sh`
- `sudo -E $CATALINA_HOME/bin/catalina stop`

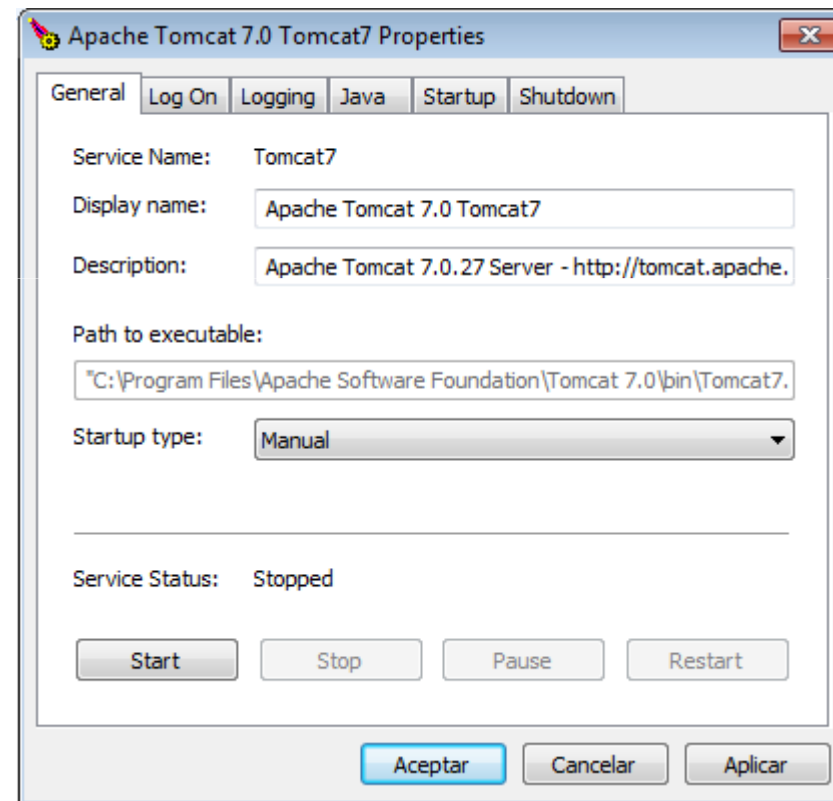
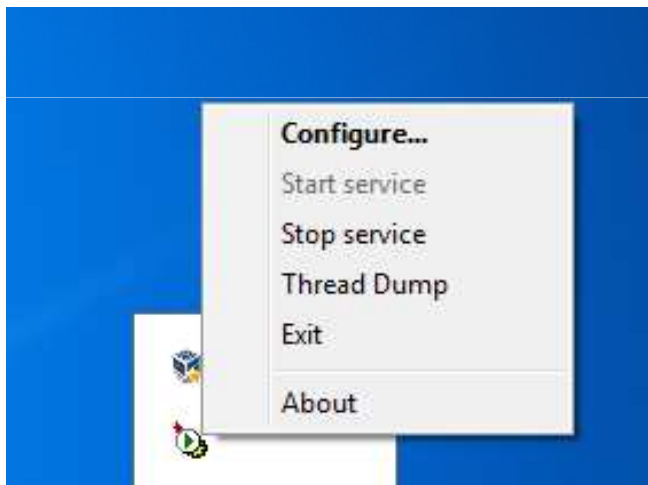
Inicio/parada del servidor *Windows*

► Iniciar/parar el servicio



Inicio/parada del servidor *Windows*

► *Tomcat Monitor*



Inicio/parada del servidor

Windows

► Inicio (varias posibilidades)

- `CATALINA_HOME/bin/starup.bat`
- `CATALINA_HOME/bin/catalina start`
- `CATALINA_HOME/bin/catalina run`

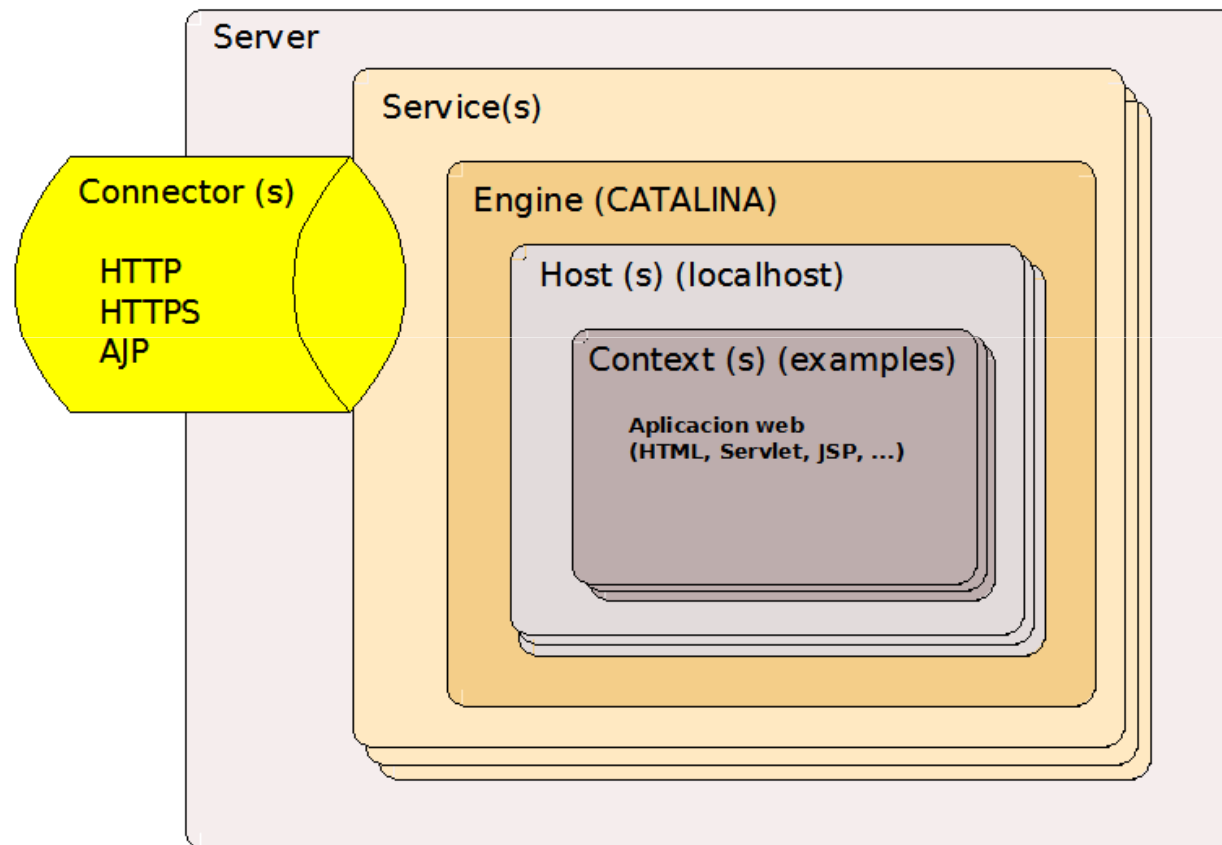
► Parada (varias posibilidades)

- `CATALINA_HOME/bin/shutdown.sh`
- `CATALINA_HOME/bin/catalina stop`

Arquitectura

- ▶ *Tomcat* esta formado por varios componentes que se organizan de forma jerárquica.
- ▶ Una instancia de *Tomcat* o servidor (*Server*) es el componente mas alto de la jerarquía.
- ▶ Consiste en un grupo de aplicaciones contenedoras.
- ▶ La configuración de los contenedores se realiza en el archivo principal de configuración `server.xml`.
- ▶ Web
 - <http://tomcat.apache.org/tomcat-7.0-doc/architecture/>

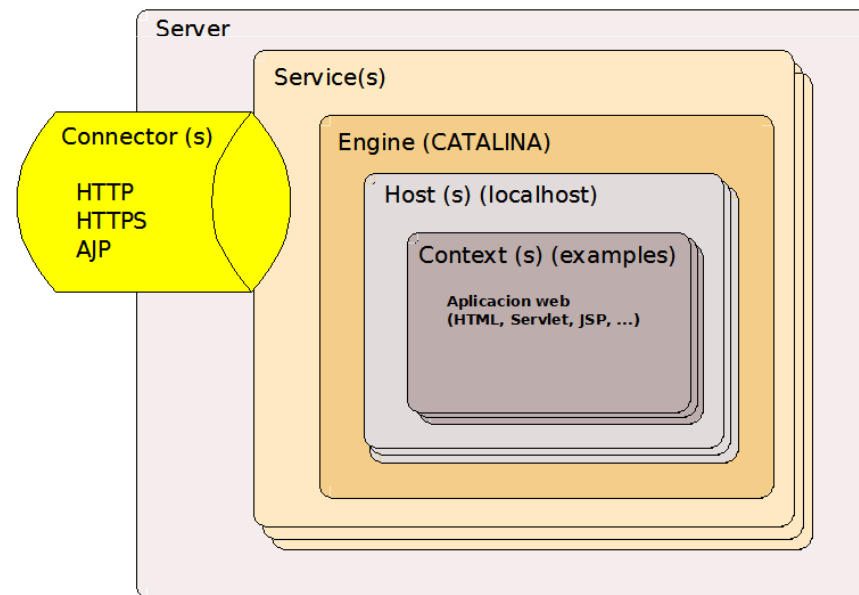
Arquitectura



Arquitectura

► *Server*

- <Server>
- El primer elemento contenedor
- Puede contener uno o varios *Services* .



Arquitectura

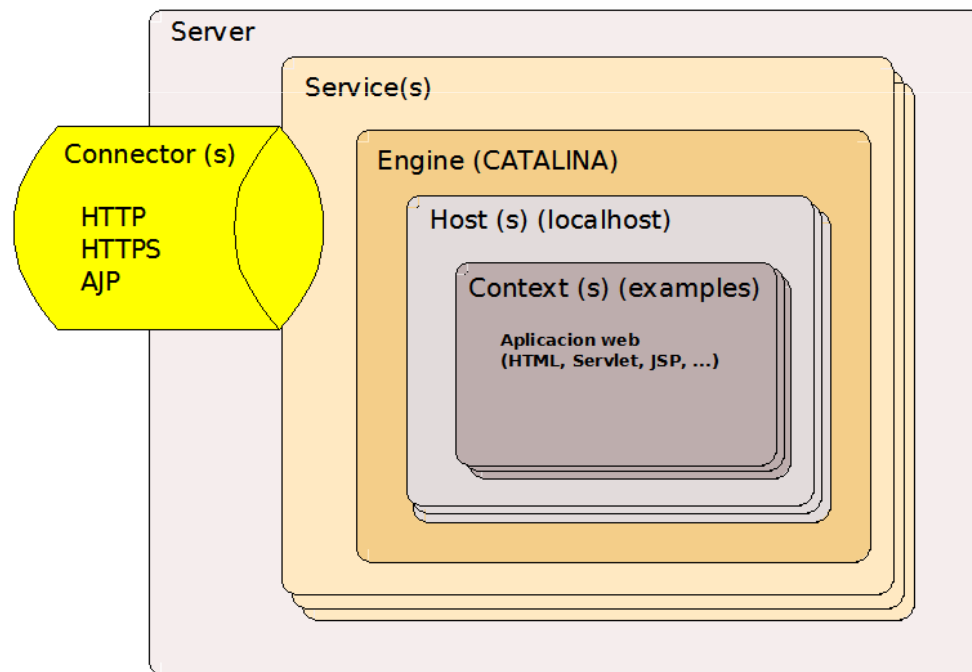
```
<Server port="8005" shutdown="SHUTDOWN">
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
  <!--
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  -->
  <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
  <Listener className="org.apache.catalina.core.JasperListener" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
```


Arquitectura

► *Service (s)*

- <Service>
- Asocia uno o mas *Connectors* con un único *Engine*.



Arquitectura

```
<Service name="Catalina">

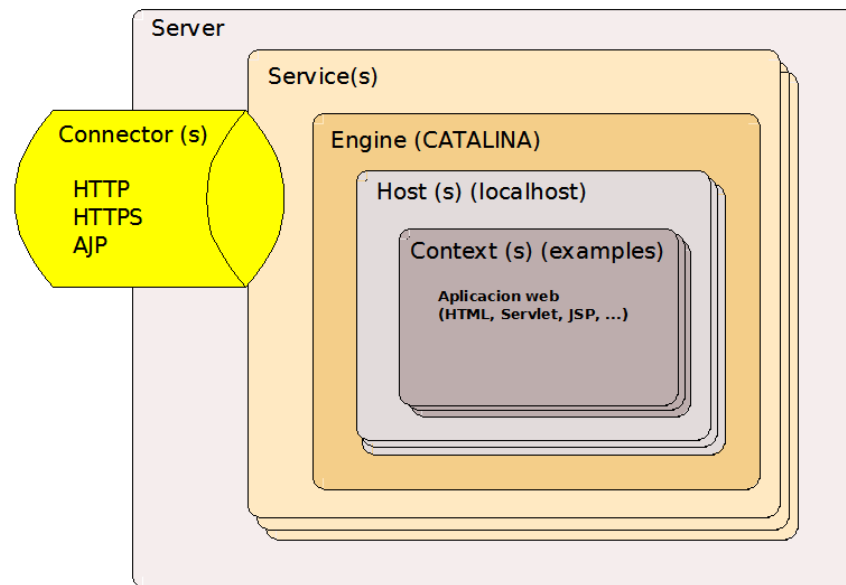
  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
    maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
  and responses are returned. Documentation at :
  Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
  Java AJP  Connector: /docs/config/ajp.html
  APR (HTTP/AJP) Connector: /docs/apr.html
  Define a non-SSL HTTP/1.1 Connector on port 8080
  -->
  <Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    URIEncoding="UTF-8"
    redirectPort="8443" />
```

Arquitectura

► *Connector (s)*

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



Arquitectura

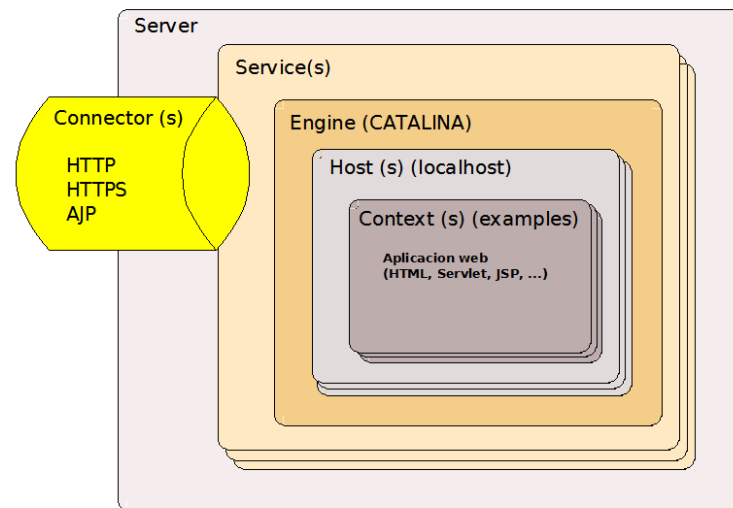
- Por defecto hay creado un conector HTTP

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

Arquitectura

► *Engine*

- <Engine>
- Contenedor de un o mas *Hosts*.
- Es posible configurar *Virtual Hosts* (como en *Apache*). Recibe las peticiones de los conectores y las traslada al *Host* correspondiente



Arquitectura

```
<Engine name="Catalina" defaultHost="localhost">

  <!--For clustering, please take a look at documentation at:
        /docs/cluster-howto.html  (simple how to)
        /docs/config/cluster.html (reference documentation) -->
  <!--
  <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
  -->

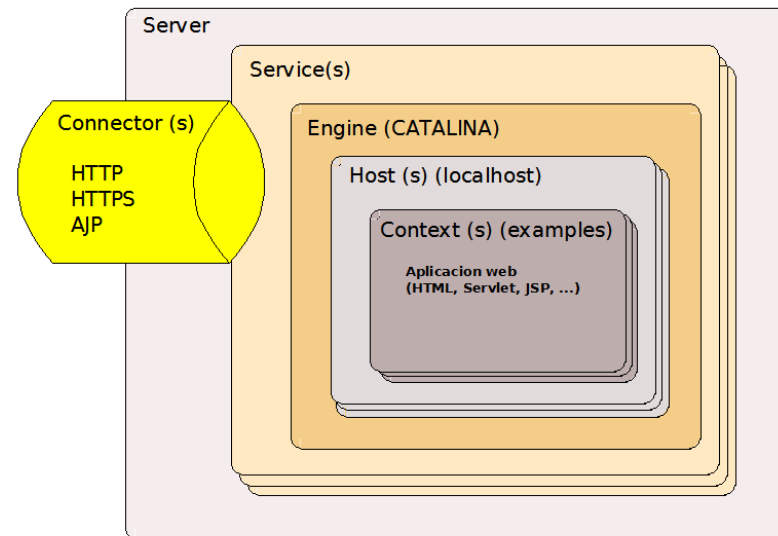
  <!-- Use the LockOutRealm to prevent attempts to guess user passwords
        via a brute-force attack -->
  <Realm className="org.apache.catalina.realm.LockOutRealm">
    <!-- This Realm uses the UserDatabase configured in the global JNDI
          resources under the key "UserDatabase". Any edits
          that are performed against this UserDatabase are immediately
          available for use by the Realm. -->
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
            resourceName="UserDatabase"/>
  </Realm>

  <Host name="localhost"  appBase="webapps"
        unpackWARs="true" autoDeploy="true">
```

Arquitectura

► *Host (s)*

- <Host>
- Define un servidor virtual (*Virtual Host*).
- Puede contener un o mas aplicaciones web (webapp). Cada una de ellas se representa por un *Context*.



Arquitectura

- Por defecto hay configurado un host llamado localhost.
- Las aplicaciones del host se colocan en \$CATALINA_BASE\webapps.

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

  <!-- SingleSignOn valve, share authentication between web applications
       Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

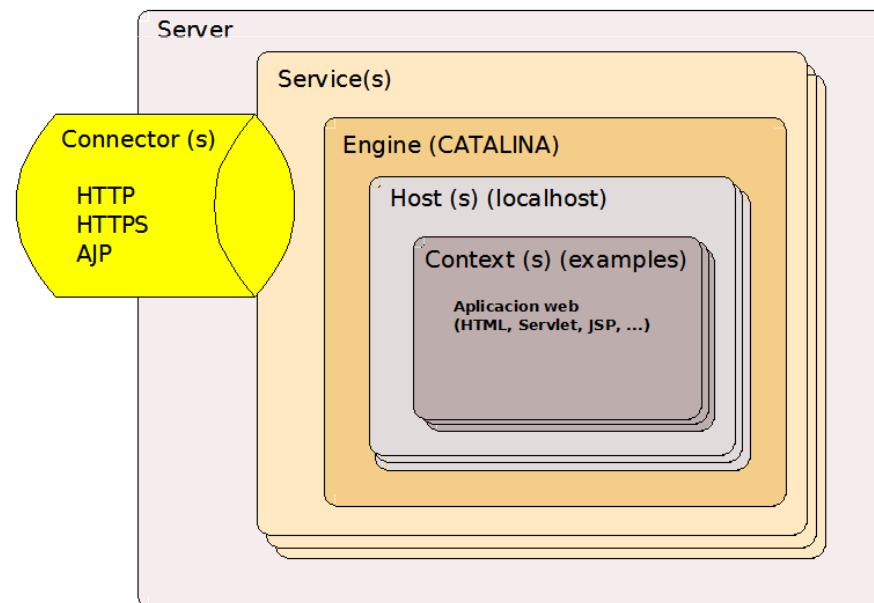
  <!-- Access log processes all example.
       Documentation at: /docs/config/valve.html
       Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log." suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
</Engine>
</Service>
</Server>
```


Arquitectura

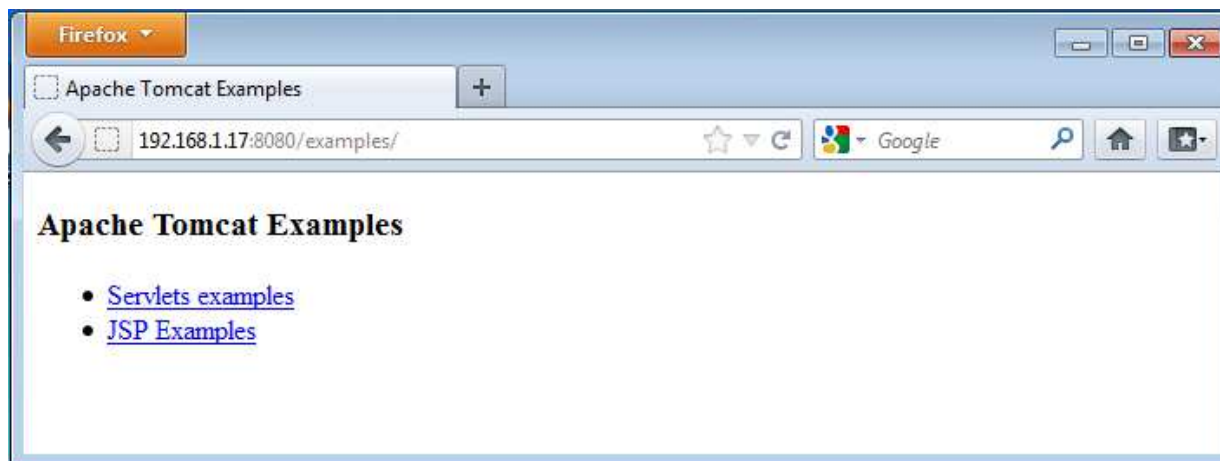
► *Context (s)*

- <Context>
- Cada *context* representa una aplicación web (HTML, CSS, Servlets, JSP,...) ejecutándose dentro de un *Host*.



Arquitectura

```
alumno@ServidorLinux01:/etc/tomcat7/Catalina/localhost$ ls -l
total 8
-rw-r--r-- 1 root root 922 mar  7 18:36 docs.xml
-rw-r--r-- 1 root root  78 mar  7 18:36 examples.xml
alumno@ServidorLinux01:/etc/tomcat7/Catalina/localhost$ cat examples.xml
<Context path="/examples"
      docBase="/usr/share/tomcat7-examples/examples" />
alumno@ServidorLinux01:/etc/tomcat7/Catalina/localhost$ cd /usr/share/tomcat7-examples/
alumno@ServidorLinux01:/usr/share/tomcat7-examples$ ls -l
total 4
drwxr-xr-x 5 root root 4096 may 28 15:27 examples
alumno@ServidorLinux01:/usr/share/tomcat7-examples$ cd examples/
alumno@ServidorLinux01:/usr/share/tomcat7-examples/examples$ ls
index.html  jsp  servlets  WEB-INF
alumno@ServidorLinux01:/usr/share/tomcat7-examples/examples$
```



Arquitectura

- ▶ Posteriormente se tratarán otros componentes:
 - *Realms*
 - Filtros
 - *Valves*
 - ...

Estructura de directorios

- ▶ Directorio de instalación → `$CATALINA_HOME`.
- ▶ *Tomcat* puede ser configurado para múltiples instancias definiendo `$CATALINA_BASE` para cada instancia.
- ▶ Si no se configuran múltiples instancias `$CATALINA_HOME` y `$CATALINA_BASE` son iguales.
- ▶ Web
 - <http://tomcat.apache.org/tomcat-7.0-doc/introduction.html>

Estructura de directorios

► Directorios (1)

◦ **/bin**

- *Scripts* de inicio y parada.
- Ficheros .jar requeridos para que *Tomcat* inicie.

```
bootstrap.jar      configtest.sh  setclasspath.sh  tomcat-juli.jar  
catalina.sh        daemon.sh      shutdown.sh      tool-wrapper.sh  
catalina-tasks.xml digest.sh      startup.sh       version.sh
```

◦ **/conf**

- Ficheros de configuración (xml) y DTDs relacionados.

```
Catalina           context.xml      policy.d         tomcat-users.xml  
catalina.properties logging.properties server.xml       web.xml
```

Estructura de directorios

► Directorios (2)

◦ `/lib`

- Ficheros `.jar` con librerías compartidas por todos los componentes de *Tomcat*.
- Todas las aplicaciones puede acceder a estas librerías.
- Incluye el API de *Servlet* y el API de JSP.

```
annotations-api.jar  commons-logging.jar  jsp-api.jar          tomcat-i18n-es.jar  
catalina-ant.jar     commons-pool.jar     servlet-api.jar      tomcat-i18n-fr.jar  
catalina-ha.jar      el-api.jar           tomcat-api.jar       tomcat-i18n-ja.jar  
catalina.jar         jasper-el.jar        tomcat-coyote.jar    tomcat-util.jar  
catalina-tribes.jar  jasper.jar           tomcat-i18n-es.jar
```

Estructura de directorios

► Directorios (3)

- **/logs**

- Ficheros de *logs*.

```
catalina.2012-05-28.log  localhost.2012-05-28.log  
catalina.out            localhost_access_log.2012-05-28.txt
```

- **/temp**

- Ficheros temporales usados por *Tomcat*.

- **/webapps**

- Directorio con las aplicaciones desplegadas.

Estructura de directorios

► Directorios (3)

- **/work**

- Directorio de trabajo donde *Tomcat* almacena los *Servlets* generados a partir de JSPs.
- Se organizan jerárquicamente en directorios con la siguiente estructura:
 - `Engine(Catalina)/host(localhost)/webapp(examples)/estructura del paquete de clases.`

```
alumno@ServidorLinux01:/var/lib/tomcat7/work$ ls
Catalina catalina.policy
alumno@ServidorLinux01:/var/lib/tomcat7/work$ cd Catalina/
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina$ ls
localhost
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina$ cd localhost/
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina/localhost$ ls
_ docs examples
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina/localhost$ cd examples/
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina/localhost/examples$ ls
org
alumno@ServidorLinux01:/var/lib/tomcat7/work/Catalina/localhost/examples$
```


Estructura de directorios

- ▶ Estructura en *Ubuntu/Debian* (paquete .deb)
 - `/usr/share/tomcat7` (`$CATALINA_HOME`)
 - `/bin`
 - `/lib`
 - `/var/lib/tomcat7` (`$CATALINA_BASE`)
 - `/conf` (enlace a `/etc/tomcat7/`)
 - `/work` (enlace a `/var/cache/tomcat7/`)
 - `/logs` (enlace a `/var/log/tomcat7/`)
 - `/tmp`
 - `/webapps`

Ficheros de configuración

► Ficheros en el directorio /conf (1)

- **server.xml**
 - Fichero de configuración principal.
- **context.xml**
 - Fichero de configuración global de *Contexts* que contiene la configuración por defecto que será compartida por todos los *Context*.
- **web.xml**
 - Fichero de configuración global de descriptores de despliegue que contiene la configuración por defecto que será compartida por todos los descriptores de despliegue.

Ficheros de configuración

► Ficheros en el directorio /conf (2)

- **tomcat-user.xml**
 - Base de datos con usuarios, roles y contraseñas para autenticación y control de acceso.
- **catalina.policy**
 - Políticas de seguridad sobre la JVM.
- **catalina.properties**
 - Definiciones de librerías cargadas al iniciar el servidor.
- **logging.properties**
 - Propiedades de logs.

```
Catalina      context.xml  policy.d    tomcat-users.xml  
catalina.properties  logging.properties  server.xml  web.xml
```

Ficheros de configuración

- ▶ Ficheros de configuración de cada *Context* (1)
 - Cada *Context* puede tener su propio directorio de configuración para personalizar la configuración definida en `server.xml` y `context.xml`.
 - La localización depende del nombre del *Engine*, de nombre del *Host* y del *Context*.

`CATALINA_BASE/conf/ENGINE_NAME/HOST_NAME/CONTEXT_NAME.xml`

`CATALINA_BASE/conf/Catalina/localhost/examples.xml`

```
alumno@ServidorLinux01:/var/lib/tomcat7/conf/Catalina/localhost$ ls
docs.xml  examples.xml
alumno@ServidorLinux01:/var/lib/tomcat7/conf/Catalina/localhost$ _
```

Ficheros de configuración

- ▶ Ficheros de configuración de cada *Context* (2)
 - También se puede crear dentro de cada aplicación.

`CATALINA_BASE/webapps/aplicación/META-INF/context.xml`

Ficheros de configuración

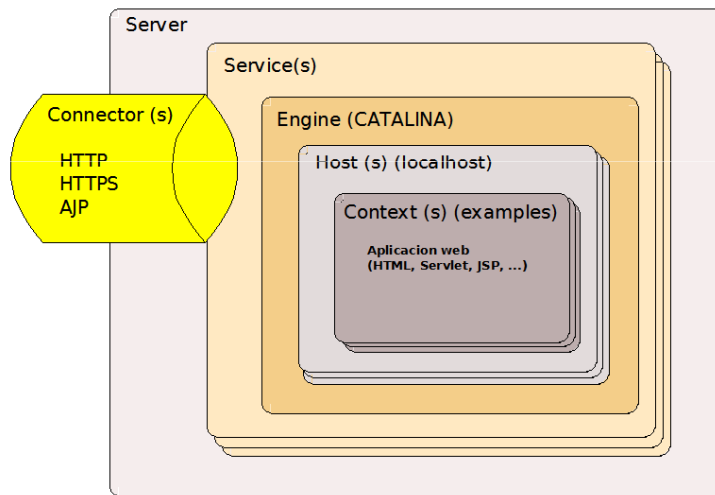
- ▶ **Descriptor de despliegue de cada aplicación**
 - Fichero `web.xml` de cada aplicación.

`CATALINA_BASE/webapps/aplicación/WEB-INF/web.xml`

Práctica

► Práctica 7.2

- Arquitectura y ficheros de configuración.



```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

  <!-- SingleSignOn valve, share authentication between web applications
       Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
       Documentation at: /docs/config/valve.html
       Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log." suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
</Engine>
</Service>
</Server>
```

```
Catalina      context.xml  policy.d     tomcat-users.xml
catalina.properties  logging.properties  server.xml   web.xml
```

Bibliografía

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