

Servidores de aplicaciones

Servidores de aplicaciones

Un servidor de aplicaciones es una forma moderna de middleware de plataforma. Es el software del sistema que reside entre el sistema operativo (SO) por un lado, los recursos externos (como un sistema de administración de bases de datos (SGBD), comunicaciones y servicios de Internet) por otro lado y las aplicaciones de los usuarios por el otro lado. La función del servidor de aplicaciones es actuar como **host** (o **contenedor**) para la lógica comercial del usuario al tiempo que facilita el acceso y el rendimiento de la aplicación comercial.

El servidor de aplicaciones se encarga de recibir las solicitudes de los clientes (como un navegador web), procesarlas y enviar una respuesta. Además de manejar las solicitudes, un servidor de aplicaciones también proporciona un entorno seguro y controlado para que las aplicaciones se ejecuten y se alojen.

Las aplicaciones que se ejecutan en un servidor de aplicaciones pueden ser simples o complejas, desde un sitio web estático hasta una aplicación empresarial de alta disponibilidad.

Además, los servidores de aplicaciones ofrecen características adicionales para mejorar la escalabilidad, la seguridad y el rendimiento de las aplicaciones. Por ejemplo, pueden controlar el acceso a los recursos de la aplicación, realizar un seguimiento de errores y optimizar la velocidad de respuesta de la aplicación.

Servidores de aplicaciones

Utilizar un servidor de aplicaciones tiene algunas ventajas, algunas de las más importantes son:

- **Escalabilidad:** Los servidores de aplicaciones permiten la escalabilidad horizontal y vertical, lo que significa que se pueden añadir más recursos a un servidor existente o agregar más servidores para manejar una mayor carga de trabajo.
- **Seguridad:** Los servidores de aplicaciones proporcionan características de seguridad integradas, como autenticación y autorización, que ayudan a proteger las aplicaciones y los datos de accesos no autorizados. Por ejemplo, pueden controlar el acceso a los recursos de la aplicación y realizar autenticación de usuarios.
- **Rendimiento:** Los servidores de aplicaciones optimizan el rendimiento de las aplicaciones mediante el uso de técnicas de caché, balanceo de carga y monitoreo en tiempo real.
- **Facilidad de uso:** La mayoría de los servidores de aplicaciones están diseñados para ser fáciles de usar y configurar, lo que permite a los desarrolladores y administradores de sistemas implementar y administrar aplicaciones de manera más eficiente.
- **Integración con otros sistemas:** Los servidores de aplicaciones permiten la integración con otros sistemas y aplicaciones, lo que mejora la eficiencia y la efectividad de los procesos empresariales.

Los servidores de aplicaciones proporcionan una plataforma robusta, segura y escalable para el desarrollo y ejecución de aplicaciones, lo que a su vez mejora la eficiencia y la productividad de las empresas.

Servidores de aplicaciones

Los servidores de aplicaciones también presentan algunas desventajas:

- **Costo:** Los servidores de aplicaciones pueden ser costosos, especialmente si se requieren características avanzadas como la escalabilidad y la seguridad.
- **Complejidad:** La configuración y el mantenimiento de un servidor de aplicaciones pueden ser complejos y requieren conocimientos técnicos especializados.
- **Vulnerabilidades de seguridad:** Aunque los servidores de aplicaciones proporcionan medidas de seguridad, también pueden ser vulnerables a ataques cibernéticos si no se configuran y mantienen correctamente.
- **Dificultad para solucionar errores:** Cuando surgen problemas en un servidor de aplicaciones, puede ser difícil identificar y solucionar los errores debido a su complejidad.
- **Limitaciones de rendimiento:** Aunque los servidores de aplicaciones pueden mejorar el rendimiento de la aplicación, también pueden estar limitados por factores como el hardware subyacente y la capacidad de procesamiento.

Los servidores de aplicaciones presentan algunas desventajas, como el costo, la complejidad, las vulnerabilidades de seguridad, la dificultad para solucionar errores y las limitaciones de rendimiento. Sin embargo, estos desafíos pueden ser superados con la planificación adecuada y la elección de un servidor de aplicaciones confiable y adecuado para sus necesidades.

Servidores de aplicaciones

Listado de servidores de aplicaciones por lenguaje:

https://en.wikipedia.org/wiki/List_of_application_servers

Java [\[edit \]](#)

Main article: Web container

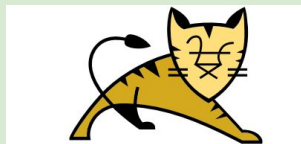
Product ↕	Vendor ↕	Edition ↕	Last release ↕	Jakarta EE compatibility ↕ [2]	Servlet ↕	JSP ↕	HTTP/2 ↕	License ↕
ColdFusion	Adobe Systems	2016.0.1	2016-05-01	7 partial platform	3.1	2.3	No	Proprietary, commercial
Enterprise Server	Borland	6.7	2007-01	1.4	2.4	2.0	No	Proprietary, commercial
Geronimo	ASF	3.0.1	2013-05-28	6 full platform	3.0	2.2	No	Free, Apache
GlassFish	Eclipse Foundation	6.2.5	2022-02-13	9.1 full platform	4.0	2.3	Yes	Free, EPL, GPL + classpath exception
JBoss Enterprise Application Platform	Red Hat	7.4	2021-07-21	8 full platform	4.0	2.3	Yes	Free, LGPL
Payara	Payara	5.2021.10	2021-12-15	8 full platform	4.0	2.3	Yes	Free, CDDL, GPL + classpath exception
Resin Servlet Container (open source)	Caucho Technology	4.0.62	2019-05-23	6 Web Profile ^[4]	3.0	2.2	No	Free, GPL
Resin Professional Application Server				6 Web Profile	3.0	2.2	No	Proprietary, commercial
Tomcat	ASF	10.1.2	2022-11-14	9 partial platform	6.0	3.1	Yes	Free, Apache v2
TomEE	ASF	8.0.8	2021-08-31	8 Web Profile	4.0	2.3	Yes	Free, Apache v2

Tomcat

Apache Tomcat es un servidor de aplicaciones Java open source que se utiliza para desplegar y ejecutar aplicaciones Java basadas el estándar JakartaEE. Tomcat tiene implementadas las especificaciones de Jakarta **Servlet**, Jakarta Server Pages (JSPs), Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications

Tomcat es conocido por su **simplicidad** y **facilidad de uso**, lo que lo hace ideal para el desarrollo y la prueba de aplicaciones Java. Además, Tomcat es un proyecto de **código abierto**, lo que significa que está disponible gratuitamente y puede ser utilizado por cualquier persona que lo necesite.

En resumen, Apache Tomcat es un servidor de aplicaciones Java popular y confiable que permite a los desarrolladores crear y desplegar aplicaciones web dinámicas y escalables con facilidad.




Apache Tomcat



JAKARTA EE

Tomcat

Apache Tomcat <http://tomcat.apache.org>



Apache Tomcat®



Apache Tomcat
Home
Taglibs
Maven Plugin

Download
Which version?
Tomcat 11 (alpha)
Tomcat 10
Tomcat 9
Tomcat 8
Tomcat Migration Tool for Jakarta EE
Tomcat Connectors
Tomcat Native
Taglibs
Archives

Documentation
Tomcat 11.0 (alpha)
Tomcat 10.1
Tomcat 10.0
Tomcat 9.0
Tomcat 8.5
Tomcat Connectors
Tomcat Native 2
Tomcat Native 1.2
Wiki
Migration Guide
Presentations
Specifications

Problems?
Security Reports
Find help
FAQ
Mailing Lists
Bug Database
IRC

Get Involved



Apache Tomcat

The Apache Tomcat® software is an open source implementation of the [Jakarta Servlet](#), [Jakarta Server Pages](#), [Jakarta Expression Language](#), [Jakarta WebSocket](#), [Jakarta Annotations](#) and [Jakarta Authentication](#) specifications. These specifications are part of the [Jakarta EE platform](#).

The Jakarta EE platform is the evolution of the Java EE platform. Tomcat 10 and later implement specifications developed as part of Jakarta EE. Tomcat 9 and earlier implement specifications developed as part of Java EE.

The Apache Tomcat software is developed in an open and participatory environment and released under the [Apache License version 2](#). The Apache Tomcat project 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. To learn more about getting involved, [click here](#).

Apache Tomcat software 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.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 8.5.85 Released2023-01-19

The Apache Tomcat Project is proud to announce the release of version 8.5.85 of Apache Tomcat. This release implements specifications that are part of the Java EE 7 platform. The notable changes compared to 8.5.85 include:

- The default value of `AccessLogValve`'s file encoding is now UTF-8.
- Correct a regression in the refactoring that replaced the use of the URL constructors. The regression broke lookups for resources that contained one or more characters in their name that required escaping when used in a URI path.
- When an HTTP/2 stream was reset, the current active stream count was not reduced. If enough resets occurred on a connection, the current active stream count limit was reached and no new streams could be created on that connection.
- Change the default of the `org.apache.el.GET_CLASSLOADER_USE_PRIVILEGED` system property to true unless the EL library is running on Tomcat in which case the default remains false as the EL library is already called from within a privileged block and skipping the unnecessary privileged block improves performance.

Full details of these changes, and all the other changes, are available in the [Tomcat 8 changelog](#).

Please note that Apache Tomcat 8.5.x will reach [End-of-life](#) (EOL) on 31 March 2024.

[Download](#)

Tomcat 10.1.5 Released2023-01-13

The Apache Tomcat Project is proud to announce the release of version 10.1.5 of Apache Tomcat. This release implements specifications that are part of the Jakarta EE 10 platform.

Applications that run on Tomcat 9 and earlier will not run on Tomcat 10 without changes. Java EE based applications designed for Tomcat 9 and earlier may be placed in the `$CATALINA_BASE/webapps-javaee` directory and Tomcat will automatically convert them to Jakarta EE and copy them to the `webapps` directory. This conversion is performed using the [Apache Tomcat migration tool for Jakarta EE tool](#) which is also available as a separate [download](#) for off-line use.

The notable changes in this release are:

- Correct a regression in the refactoring that replaced the use of the URL constructors. The regression broke lookups for resources that contained one or more characters in their name that required escaping when used in a URI path.
- When resetting an HTTP/2 stream because the final response has been generated before the request has been fully read, use the HTTP/2 error code `NO_ERROR` so that client does not discard the response. Based on a suggestion by Lorenzo Dalla Vecchia.
- Change the default of the `org.apache.el.GET_CLASSLOADER_USE_PRIVILEGED` system property to true unless the EL library is running on Tomcat in which case the default remains false as the EL library is already called from within a privileged block and skipping the unnecessary privileged block improves performance.

Tomcat

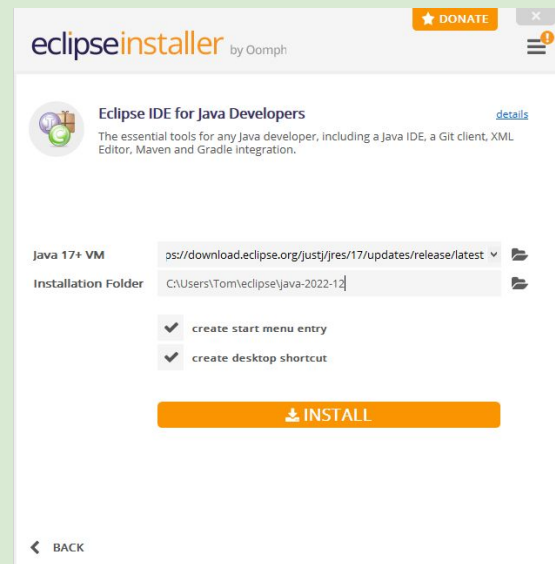
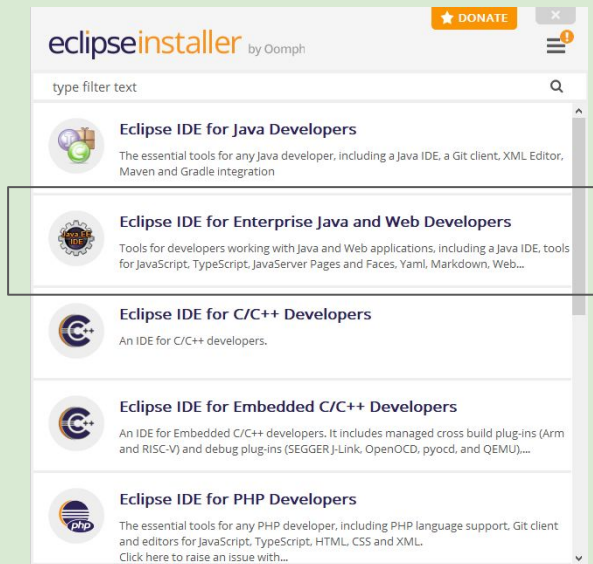
Versiones de tomcat

Servlet Spec	JSP Spec	EL Spec	WebSocket Spec	Authentication (JASPIC) Spec	Apache Tomcat Version	Latest Released Version	Supported Java Versions
6.1	4.0	6.0	TBD	TBD	11.0.x	11.0.0-M1 (alpha)	17 and later
6.0	3.1	5.0	2.1	3.0	10.1.x	10.1.5	11 and later
5.0	3.0	4.0	2.0	2.0	10.0.x (superseded)	10.0.27 (superseded)	8 and later
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.71	8 and later
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.85	7 and later
3.1	2.3	3.0	1.1	N/A	8.0.x (superseded)	8.0.53 (superseded)	7 and later
3.0	2.2	2.2	1.1	N/A	7.0.x (archived)	7.0.109 (archived)	6 and later (7 and later for WebSocket)
2.5	2.1	2.1	N/A	N/A	6.0.x (archived)	6.0.53 (archived)	5 and later
2.4	2.0	N/A	N/A	N/A	5.5.x (archived)	5.5.36 (archived)	1.4 and later
2.3	1.2	N/A	N/A	N/A	4.1.x (archived)	4.1.40 (archived)	1.3 and later
2.2	1.1	N/A	N/A	N/A	3.3.x (archived)	3.3.2 (archived)	1.1 and later

Tomcat

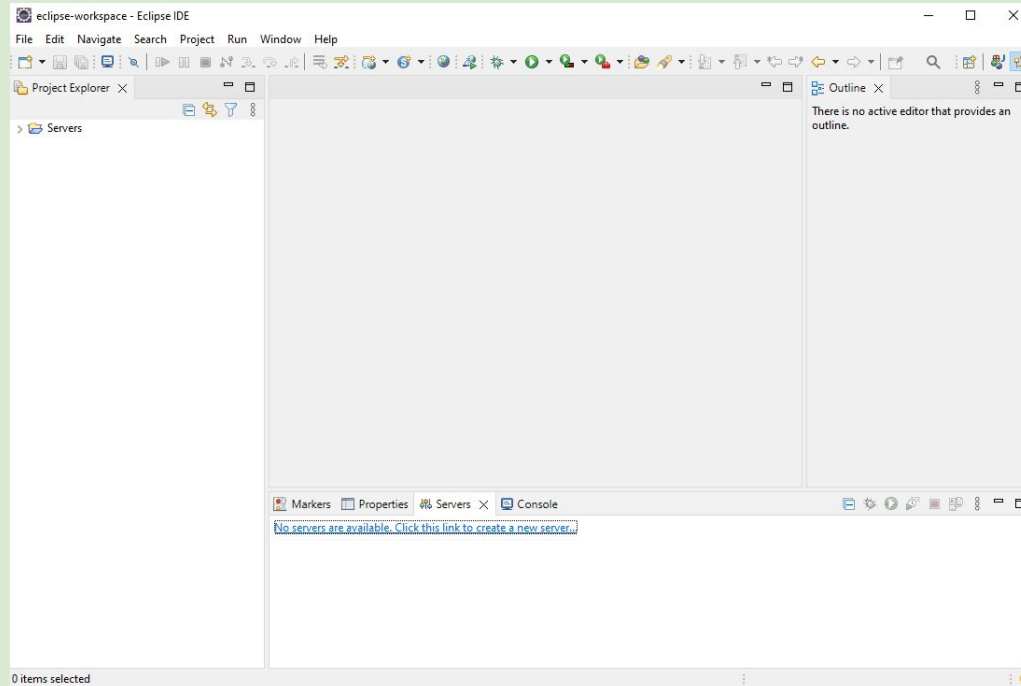
Instalación del IDE eclipse: <https://www.eclipse.org/downloads>

1. Descargamos el instalador y lo ejecutamos.
2. Seleccionamos Eclipse for Enterprise Java and Web Developers.
3. Procedemos a instalar.



Tomcat

Procedemos a ejecutar Eclipse.



Tomcat

Ahora vamos a descargar Tomcat de dos maneras diferentes:

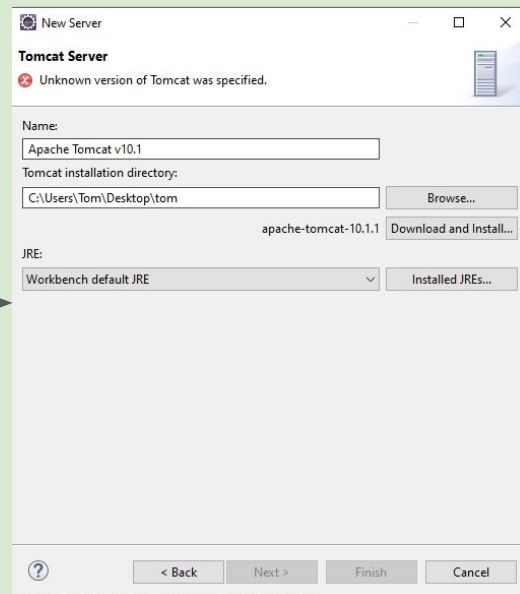
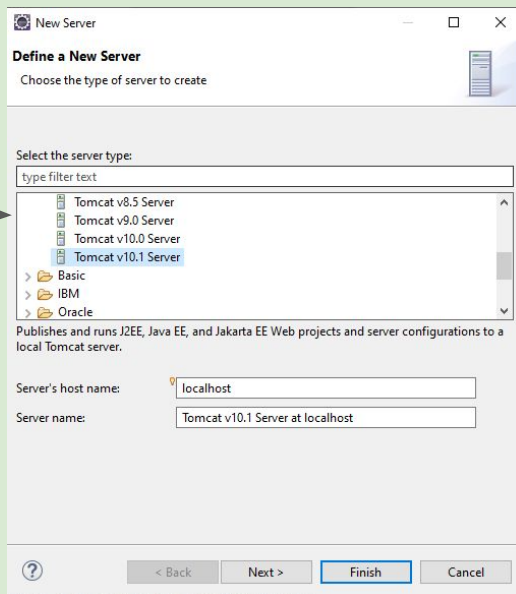
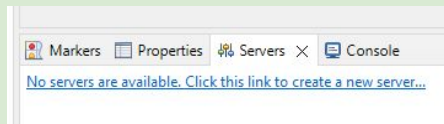
- Desde la web, en la que se asemeja a la instalación que haremos en un servidor de producción.
- Desde el IDE Eclipse en el que tendremos un tomcat para desarrollo.

Para descargas desde la web es tan sencillo cómo descargar el zip de la versión que queramos y descomprimir.



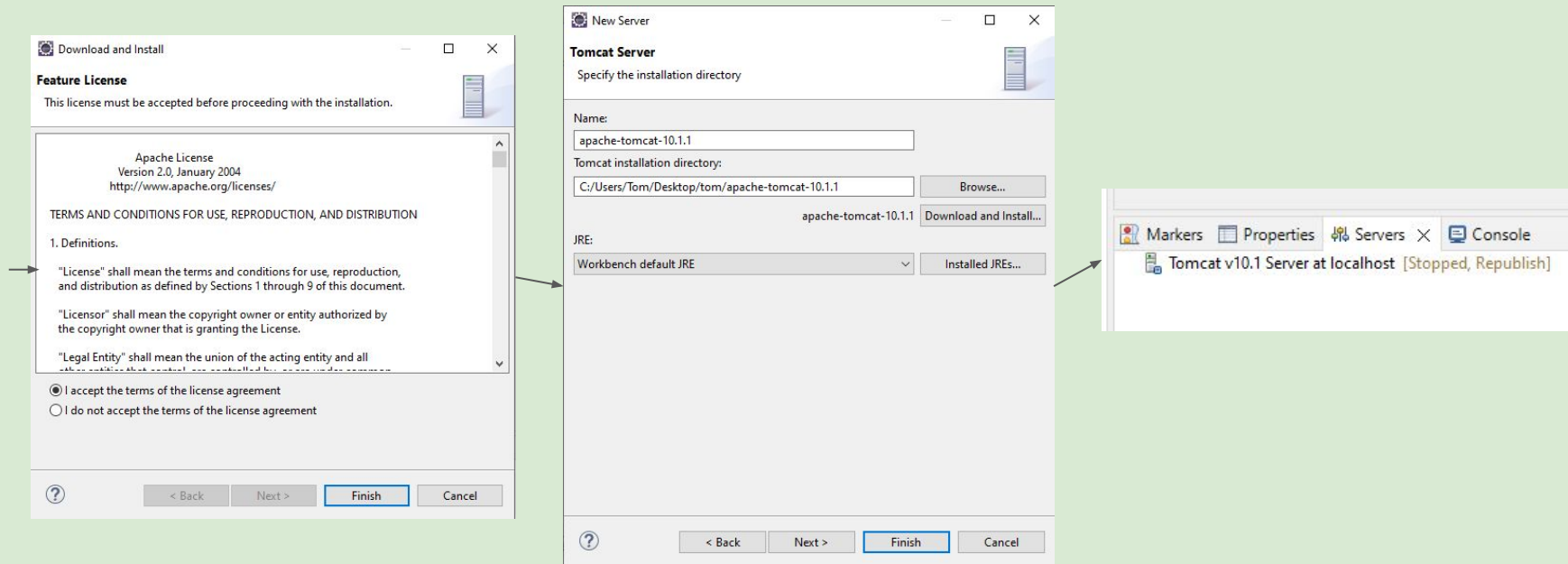
Tomcat

Para descargar y configurar el servidor en el IDE Eclipse. En la pestaña Servers (Si no la encontramos está en el menú Window>Show View>Others>Servers) aparece que no hay servidores. Hacemos click en el enlace para crear un servidor nuevo. Seleccionamos Tomcat 10.1. Seleccionamos la carpeta donde vamos a poner Tomcat (**Browse**) y le damos a **Download and Install**.



Tomcat

Aceptamos la licencia de Tomcat. Seleccionamos la carpeta de descarga que es la misma que indicamos antes. Esperamos un poco porque se tiene que descargar tomcat (se puede ver el proceso en la ventana en la esquina inferior derecha). Al acabar le damos a Finalizar y podemos ver que se ha creado un servidor.



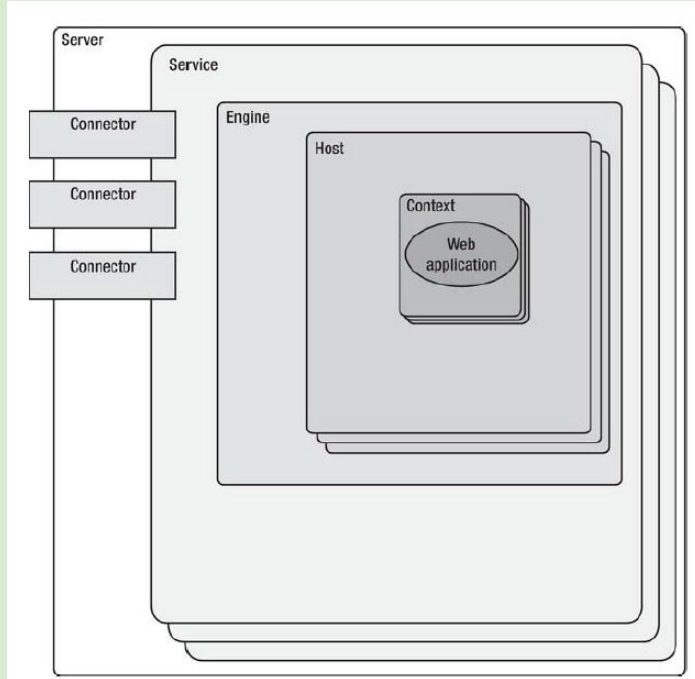
Servidores de aplicaciones

Apache-Tomcat usa una arquitectura que permite de forma lógica conectar sus diferentes componentes y que cada uno realice una función determinada en la estructura arborescente. Por ello se van a listar los componentes que posee este servidor de aplicaciones:

- a) **Server**: es el componente que representa el contenedor al completo y dentro de él se ejecutan los demás componentes.
- b) **Service**: es un componente intermedio que se comunica y que está dentro del componente server, que posee varias conexiones. El motor suele ser Catalina y los conectores por defecto HTTP y AJP (Apache Jserv Protocolo).
- c) **Engine**: es el servicio que permite procesar las peticiones que provienen desde el exterior y responder a tales solicitudes. Por ello, posee múltiples conectores que hacen posible resolver las peticiones.
- d) **Host**: El elemento host define los virtual hosts que están contenidos en cada instancia de Catalina (engine). Cada host puede ser el padre de una o más aplicaciones web representadas por un context. Como pueden ser `www.daw.com` o `www.daw.es`.
- e) **Connector**: es el encargado de controlar las comunicaciones con el cliente. Tomcat incluye varios conectores, como AJP Connector o HTTP Connector.
- f) **Context**: representa una aplicación web. Un host puede poseer varios o múltiples contextos pero con una única ruta.

Servidores de aplicaciones

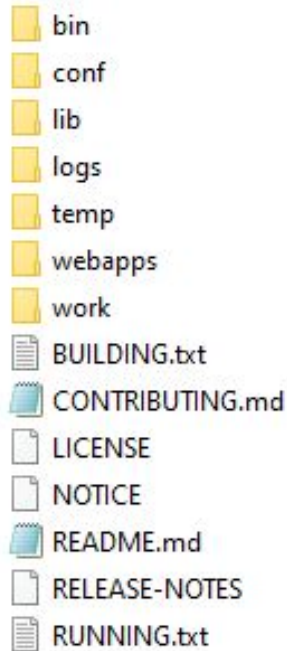
Podemos verlo de forma gráfica en la siguiente imagen:



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

Servidores de aplicaciones

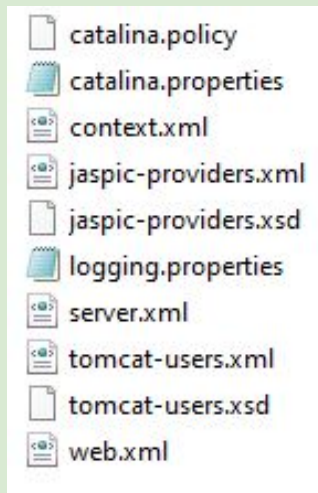
Los directorios comunes son:



- **bin:** Contiene los binarios y scripts de inicio de Tomcat.
- **conf:** La configuración global de Tomcat.
- **lib:** Contiene todos los .JAR comunes a todas las aplicaciones. Contiene todos los ficheros .jar usados en el servidor que corresponden a Tomcat, a JSP, etc.
- **logs:** Aquí van los archivos de registro.
- **temp:** Este directorio se usa para los archivos temporales que necesita Tomcat durante su ejecución.
- **webapps:** Aquí se ubican las aplicaciones web.
- **work:** Es un directorio para los archivos en uso, cuando se compilan los JSPs, etc.

Servidores de aplicaciones

Los archivos de configuración (/conf) son:



- **catalina.policy:** Este archivo contiene la política de seguridad relacionada con Java e impide que los Servlets o JSPs la sobrescriban por motivos de seguridad.
- **catalina.properties:** Contiene los archivos .JAR que no pueden sobrescribirse por motivos de seguridad y otros de uso común.
- **context.xml:** El archivo de contexto común a todas las aplicaciones. Se utiliza principalmente para informar de dónde se puede encontrar el archivo web.xml de las propias aplicaciones.
- **jaspic-providers:** Configuración de la integración de implementaciones de autenticación JASPIC de terceros con Tomcat.
- **logging.properties:** Establece las políticas generales para el registro de actividad del servidor, aplicaciones o paquetes.
- **server.xml:** Ya hemos visto que es el fichero principal de configuración de Tomcat y que tiene mucho que ver con su arquitectura.
- **tomcat-users.xml:** Contiene los usuarios, contraseñas y roles usados para el control de acceso. Es el archivo donde se encuentra la información de seguridad para las aplicaciones de administración de Tomcat.
- **web.xml:** Un descriptor de despliegue por defecto con la configuración compartida por todas las aplicaciones. Es un archivo con directivas de funcionamiento de las aplicaciones. Los descriptors de cada aplicación sobrescribirá la información escrita en este descriptor.