DAW Práctica 3.3: Primeros pasos con Apache Tomcat

En esta actividad vamos a instalar y hacer una primera configuración básica del servidor de aplicaciones Apache Tomcat.

**Procedimiento:**

Instrucciones:

1. Entrar en la web de Apache Tomcat y ver las distintas versiones que hay. Reflexión, ¿Qué sentido tiene que haya distintas versiones mantenidas a la vez?

Mantener distintas versiones de Apache Tomcat a la vez tiene varios propósitos importantes:

* Compatibilidad con Aplicaciones Existentes: Muchas aplicaciones están diseñadas para funcionar con versiones específicas de Tomcat y de las especificaciones de Java EE o Jakarta EE. Al mantener versiones anteriores, se asegura que las aplicaciones existentes continúen operando correctamente sin requerir actualizaciones inmediatas.
* Transición Gradual: Las organizaciones necesitan tiempo para actualizar sus sistemas. Mantener múltiples versiones permite una transición gradual hacia versiones más recientes, reduciendo el riesgo de problemas asociados con actualizaciones forzadas.
* Estabilidad y Seguridad: Las versiones anteriores reciben actualizaciones de seguridad y correcciones de errores. Esto proporciona estabilidad y seguridad continuas para aplicaciones que no pueden ser actualizadas de inmediato a versiones más recientes.
* Diversidad de Necesidades: Diferentes usuarios y proyectos pueden tener necesidades distintas en cuanto a funcionalidades y soporte de especificaciones. Al ofrecer varias versiones, Apache Tomcat satisface una amplia gama de necesidades y preferencias.
* Minimización de Riesgos: La coexistencia de múltiples versiones permite a las organizaciones probar nuevas versiones en un entorno controlado antes de implementarlas completamente en producción, minimizando riesgos.

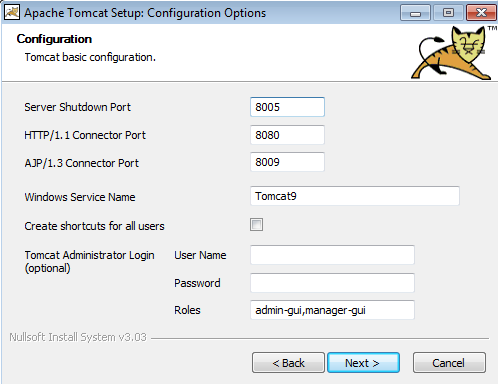
1. Vamos a instalar la versión 9. Hay que instalar previamente la versión de JDK correspondiente (es compatible con el jdk 23, el más actual). Revisarlo en la web de Tomcat y descargar el JDK de la página oficial.
2. En primer lugar, vamos a instalar Apache Tomcat mediante el archivo de instalación:
   1. Entra en la web de descarga de la versión elegida.
   2. Dentro de la sección “Core” descarga el archivo “32-bit/64-bit Windows Service Installer”
   3. En la página de configuración “Choose Componets” escoge todos los

componentes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En la pantalla de configuración puedes elegir lo siguiente:



En esta configuración cabe destacar el puerto en el que escuchará el servidor (8080).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Una vez acabada la instalación, comprobar que el servidor está arrancado. En el administrador de tareas en la pestaña de servicios debería haber un proceso llamado Tomcat9.



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

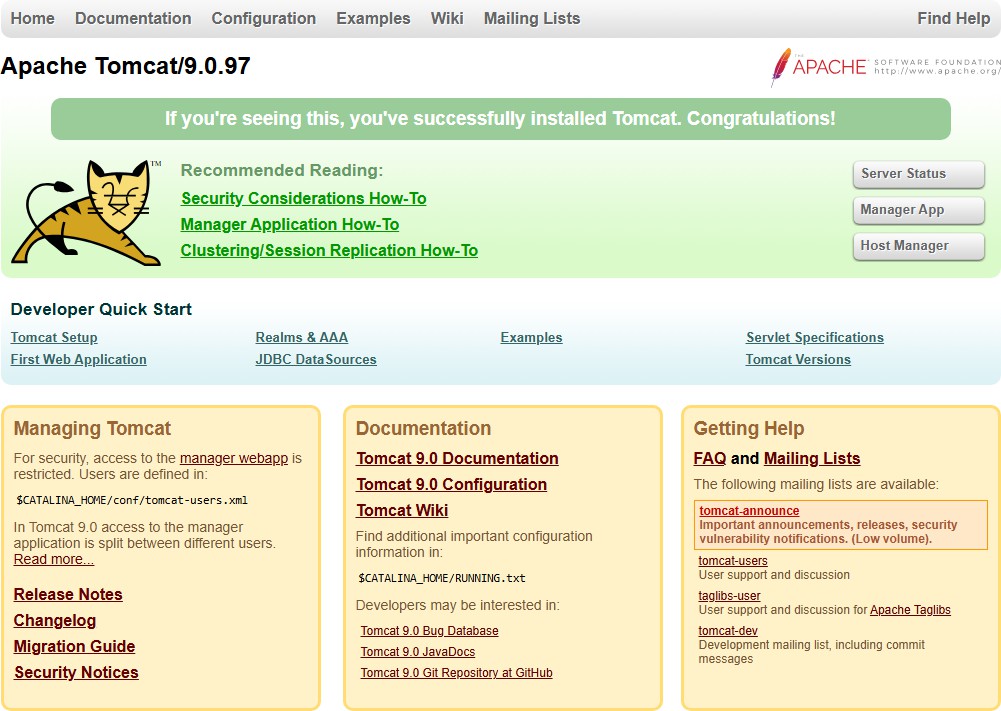
1. Consulta el contenido de la carpeta “C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Especifica los principales componentes que observas en ese directorio.
2. Para comprobar que el servicio está funcionando usa el navegador y entra

en “[http://localhost:8080](http://localhost:8080/)”



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

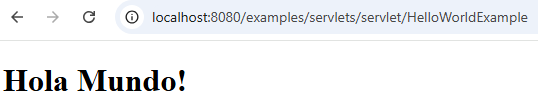
1. Uno de los componentes que utiliza Apache Tomcat son servlets pero **¿qué es un servlet?**

Un servlet es un componente de software que se ejecuta en un servidor web o de aplicaciones y que genera contenido dinámico para aplicaciones web basadas en Java. Los servlets son parte de la tecnología Java Servlet y están diseñados para manejar solicitudes y respuestas HTTP.

1. Una vez que hemos instalado y comprobado que el servidor está funcionando, vamos a ejecutar una aplicación Java en él. En concreto vamos a ejecutar un servlet.
   1. En la página principal del servidor (localhost:8080”) haz click en

Examples:

* 1. Entra en Servlets Examples.
  2. Prueba a ejecutar “Hello World” con el enlace Execute.
  3. Deberías ver en la pantalla el mensaje “Hola Mundo!”.



* 1. Prueba a ver el código de ese Hello World haciendo click en el enlace Source de la página de ejemplos.



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* 1. Prueba otros ejemplos de Servlets.

Interfaz de usuario gráfica, Texto

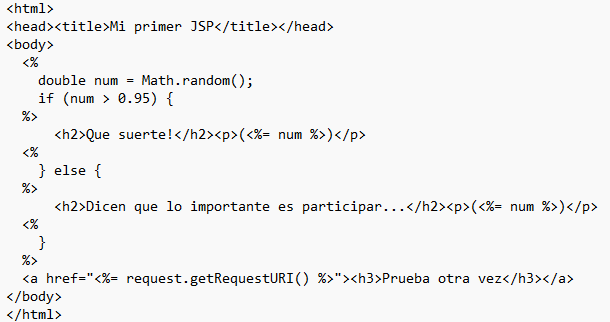
Descripción generada automáticamente con confianza media

1. Vamos a ver el funcionamiento de un JSP, pero ¿qué es un JSP? ¿Qué diferencias encontramos entre JSP y servlet?

JSP (JavaServer Pages) es una tecnología que permite crear contenido dinámico en páginas web mediante el uso de Java. JSP facilita la creación de aplicaciones web interactivas al combinar HTML con fragmentos de código Java incrustado, que se procesan en el servidor antes de enviar la página al cliente.

Las diferencias entre JSP y Servlets se pueden resumir de la siguiente manera:

1. **Propósito**:
   * JSP está diseñado principalmente para la presentación, permitiendo crear páginas web dinámicas mediante una combinación de HTML y código Java.
   * Los Servlets se centran en la lógica de negocio y el manejo de solicitudes y respuestas HTTP.
2. **Sintaxis**:
   * JSP permite incrustar código Java dentro de HTML, lo que lo hace más accesible para diseñadores que trabajan en la interfaz de usuario.
   * Los Servlets están completamente escritos en Java, lo que los hace más adecuados para desarrolladores que manejan lógica compleja.
3. **Facilidad de Uso**:
   * JSP es más fácil de usar para quienes están familiarizados con HTML, ya que pueden insertar pequeños fragmentos de código Java en el contenido.
   * Los Servlets requieren un conocimiento más profundo de Java y su uso puede resultar menos intuitivo para diseñadores.
4. **Separación de Lógica y Presentación**:
   * JSP facilita la separación entre la lógica de negocio y la presentación, lo que resulta en un código más limpio y fácil de mantener.
   * En Servlets, la mezcla de lógica y presentación puede hacer el código más difícil de mantener.
5. **Conversión**:
   * Los JSP se convierten automáticamente en Servlets en tiempo de ejecución, lo que les permite aprovechar el modelo de Servlets para el manejo de solicitudes.
   * Los Servlets son la base subyacente y no requieren conversión.
6. Crea un archivo de texto plano llamado “ejemplo.jsp”. Guárdalo en la carpeta “C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ejemplo\ejemplo.jsp”



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Una vez guardado el archivo podemos comprobar su funcionamiento abriendo en el navegador:

<http://localhost:8080/ejemplo/ejemplo.jsp>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Crea tú mismo un JSP con alguna funcionalidad básica.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente