



## Sesión 5: Introducción a Java EE

# Índice



- **Cliente-servidor y protocolo HTTP**
- **El servidor Tomcat**
- **Introducción a las aplicaciones web Java EE**

# Introducción a Java EE



- **Cliente-servidor y protocolo HTTP**
- **El servidor Tomcat**
- **Introducción a las aplicaciones web Java EE**

# Cliente-servidor



- **Toda aplicación web se basa en el protocolo *cliente-servidor***
  - **El cliente (navegador) proporciona al usuario unas interfaces (formularios, enlaces, etc) para pedir páginas y acciones a un servidor**
  - **El servidor ejecuta las peticiones del cliente y muestra los resultados relacionados (listado de búsqueda, alta en un registro, etc)**

# Protocolo HTTP



- **En este proceso interviene una serie de factores**
  - **La *petición* que realiza el usuario o cliente**
  - **Las *cabeceras de petición* que envía el cliente (normalmente añadidas automáticamente por el navegador), con información sobre la petición**
  - **La *respuesta* que le envía el servidor (p. ej, la página solicitada)**
  - **Las *cabeceras de respuesta* que envía el servidor, con información sobre la petición**
  - **Un *código de estado* enviado por el servidor, indicando si la petición se ha podido atender bien**
    - p. ej, el código 404 indica un tipo de error
- **Estos elementos se envían mediante el protocolo HTTP**

# Protocolo HTTP: peticiones



- La estructura general de una petición es:

Comando URI Protocolo

Cabeceras de petición

Datos adicionales

# Protocolo HTTP: peticiones



## ■ Ejemplos:

```
GET /dir/cargaPagina.php?id=21&nombre=Pepe HTTP/1.1
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0 (compatible;MSIE5.0;Windows 98)
```

```
POST /dir/cargaPagina.php HTTP/1.1
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0 (compatible;MSIE5.0;Windows 98)
```

```
id=21&nombre=Pepe
```

# Protocolo HTTP: respuestas



- La estructura general de una respuesta es:

Protocolo Código Mensaje

Cabeceras de respuesta

Datos de la respuesta



# Introducción a Java EE



- **Cliente-servidor y protocolo HTTP**
- **El servidor Tomcat**
- **Introducción a las aplicaciones web Java EE**

# Tomcat



- **Tomcat es un servidor web que soporta gran parte de la arquitectura Java EE**
- **No es tan potente como un servidor de aplicaciones pero puede dar soporte a aplicaciones web bastante completas**
- **Más información en**

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

# Instalación



- **Ejecutable Windows**: ejecutar el instalador y seguir los pasos
- **Fichero ZIP** (Windows o Linux): descomprimir en el lugar deseado
- **Se deben definir dos variables (con el instalador no es necesario):**
  - **JAVA\_HOME**: con el directorio de instalación de Java
  - **CATALINA\_HOME**: con el directorio de instalación de Tomcat

# Ejecución



- **Ejecutable Windows: *Inicio – Programas – Apache Tomcat 5.5 – Monitor Tomcat***
- **Fichero ZIP (Windows o Linux): ejecutables *startup* o *shutdown* de la subcarpeta *bin***
- **Una vez arrancado, abrir desde un navegador:**
  - ***`http://localhost:8080`***



# Estructura



- Carpeta *bin*: ejecutables
- Carpeta *common*: clases y librerías compartidas
- Carpeta *conf*: configuración:
  - *server.xml*: fichero principal de configuración
  - *web.xml*: configuración general para aplicaciones web
  - *tomcat-users.xml*: lista de usuarios y contraseñas
  - *catalina.policy*: política de permisos
- Carpeta *webapps*: donde irán las aplicaciones web
- Carpeta *logs*: donde se volcarán los mensajes de log del servidor
- ... etc

# Introducción a Java EE



- **Cliente-servidor y protocolo HTTP**
- **El servidor Tomcat**
- **Introducción a las aplicaciones web Java EE**

# Elementos a considerar



- **En el servidor:**
  - La aplicación debe recoger parámetros y peticiones del cliente, procesarlos y devolver un resultado
  - Podrá valerse (de forma **TRANSPARENTE** para el cliente) de herramientas externas (servlets, JSP, PHP, ASP...)
- **En el cliente:**
  - HTML es estático y no permite más que dar formato y estructura a la información
  - Para cosas más complejas (validar formularios, mostrar evoluciones de datos, etc) podemos valernos de herramientas como Javascript, Flash, Applets, etc.

# Estructura de una aplicación Java EE



- **Directorio raíz /:** desde él podemos colgar páginas HTML o JSP (estructuradas en subcarpetas si se quiere)
- **Carpeta */WEB-INF*:** contiene la información web relevante para la aplicación:
  - Fichero descriptor */WEB-INF/web.xml*
  - Librerías en */WEB-INF/lib/*
  - Clases Java en */WEB-INF/classes/*
- **El resto de elementos (imágenes, etc) podemos ponerlos como queramos (aunque fuera de WEB-INF)**
- **CUALQUIER servidor web Java EE soporta esta estructura**
  - Sólo hace falta copiar la carpeta con todo en el servidor correspondiente



# Contextos



- Cada aplicación Java EE es un contexto, una unidad con sus recursos, clases y configuración
- Cada contexto o aplicación web tendrá asociada una ruta en el servidor web
  - Por ejemplo, para acceder a la aplicación *aplic* en nuestra máquina local:
    - <http://localhost/aplic/index.html>
- Podemos empaquetar la aplicación en un fichero WAR (con el comando *jar*) para que sea más fácil de distribuir o de llevar de un sitio a otro.