

# JavaServer Pages

Sistemas Uni

# JSP

- Es tecnología de J2EE
- Permite generar contenido HTML en forma dinámica
- Muy similar a la tecnología Active Server Pages de Microsoft

# JSP JavaServer Pages™

La tecnología JSP es una manera sencilla de implementar toda la potencia de Java en aplicaciones WEB.

Es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente.

Es una tecnología basada en Java que simplifica el proceso de desarrollo de páginas y sitios web dinámicos.

Hay que pensar de JSP como un tipo de lenguaje que se encuentra en el lado del servidor.

# JSP JavaServer Pages™

Esta tecnología acelera el proceso de desarrollo de páginas web dinámicas en diferentes formas:

- ✓ Separando el contenido de generación y el contenido de presentación.  
Esto ayuda a autores a proteger su código propietario mientras que asegura la portabilidad completa para cualquier web browser basado en HTML.
- ✓ Enfatizando la reutilización de componentes.
- ✓ Simplifica el desarrollo de páginas mediante el empleo de tags.

# Beneficios JSP

- ✓ **Portabilidad:** La tecnología JSP es independiente de la plataforma, en el código de la pagina JSP, del servidor web y de sus componentes.
- ✓ Reusabilidad de Componentes y de Librerías de etiquetas.
- ✓ Separación de Contenido Estático – Dinámico.
- ✓ Soporte para la Escritura de Scripts y Acciones

# Beneficios de JSP

- ✓ Por ser tecnología Java se heredan todos los beneficios del lenguaje
- ✓ Tiene mejor tiempo de respuesta que CGI
- ✓ Facilita la distribución del trabajo en equipos de Desarrollo

# Performance

- ✓ Cuando un usuario hace request de CGI, CGI genera un nuevo proceso
- ✓ Las páginas JSP son llevadas a Servlets

# Performance

- ✓ Los request son derivados a un Servlet Container
- ✓ Un container puede manipular múltiples request a la vez
- ✓ Java genera threads para mejorar el servicio

# Performance

- ✓ Algunos Servlets containers son capaces de trabajar in-process con el servidor Web
  - IBM WebSphere
  - iPlanet 4.0 de Netscape

# ¿Qué es una Pagina JSP?

JSP es un lenguaje que queda inmerso dentro de una página WEB, este lenguaje de programación es Java. Puede trabajar con tres clases de archivos Java: applets, beans y servlets.

# Requerimientos

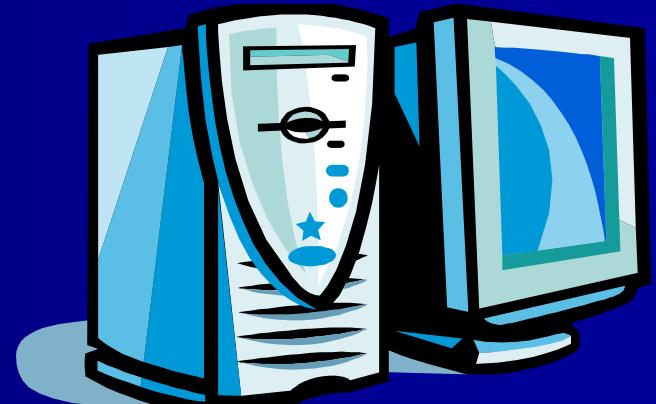
- ✓ Sevidor (Web Server)



- ✓ JDK (Java Server Web Development Kit)

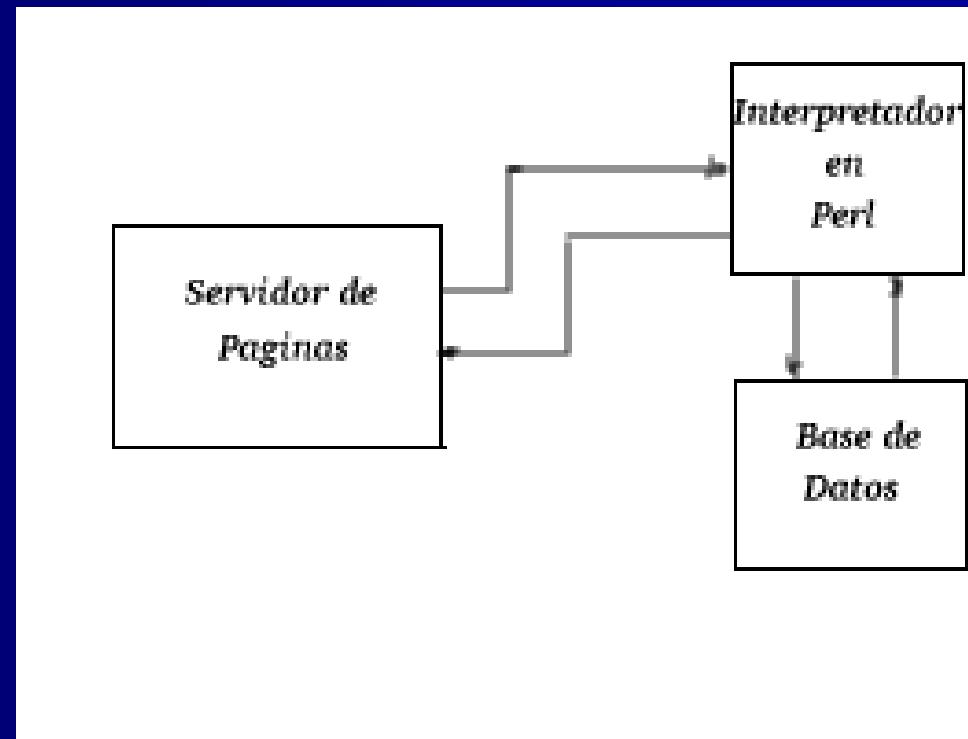
# Servidor

- ✓ Parte Primordial
- ✓ Genera y envía información
- ✓ Tipos de servidores
  - \* Páginas
  - \* Aplicaciones



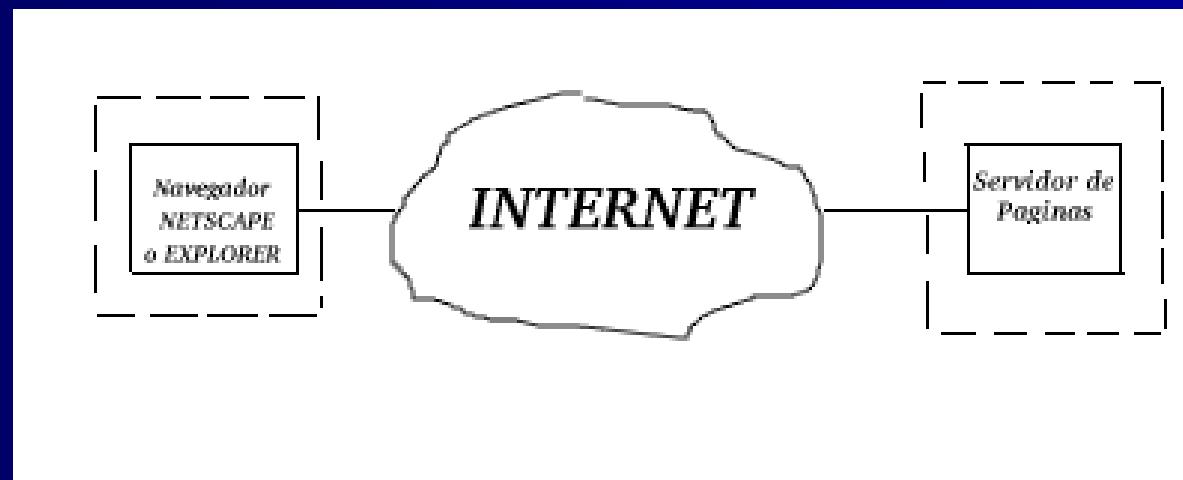
# Servidores de Páginas

- ✓ Enviar datos
- ✓ Interpretador
- ✓ Sitios Pequeños



# Servidores de Aplicaciones

- ✓ Mayor Tecnología (Procesadores, Memoria)
- ✓ Complejidad en el diseño
- ✓ Genera Acciones
- ✓ Ejecuta código



# JSP

JSP : documento que describe como crear un objeto respuesta (Response Object) a partir de un objeto petición (Request Object) mediante un protocolo determinado, creando y/o usando otros objetos.

# Traducción y ejecución

- ✓ JSP pages = componentes textuales
- ✓ Pasan por 2 fases:
  - ✓ Fase de traducción (translation phase).
  - ✓ Fase de petición (request phase).
- ✓ Fase de traducción una vez por página.
- ✓ Fase de petición cada que hay una petición.

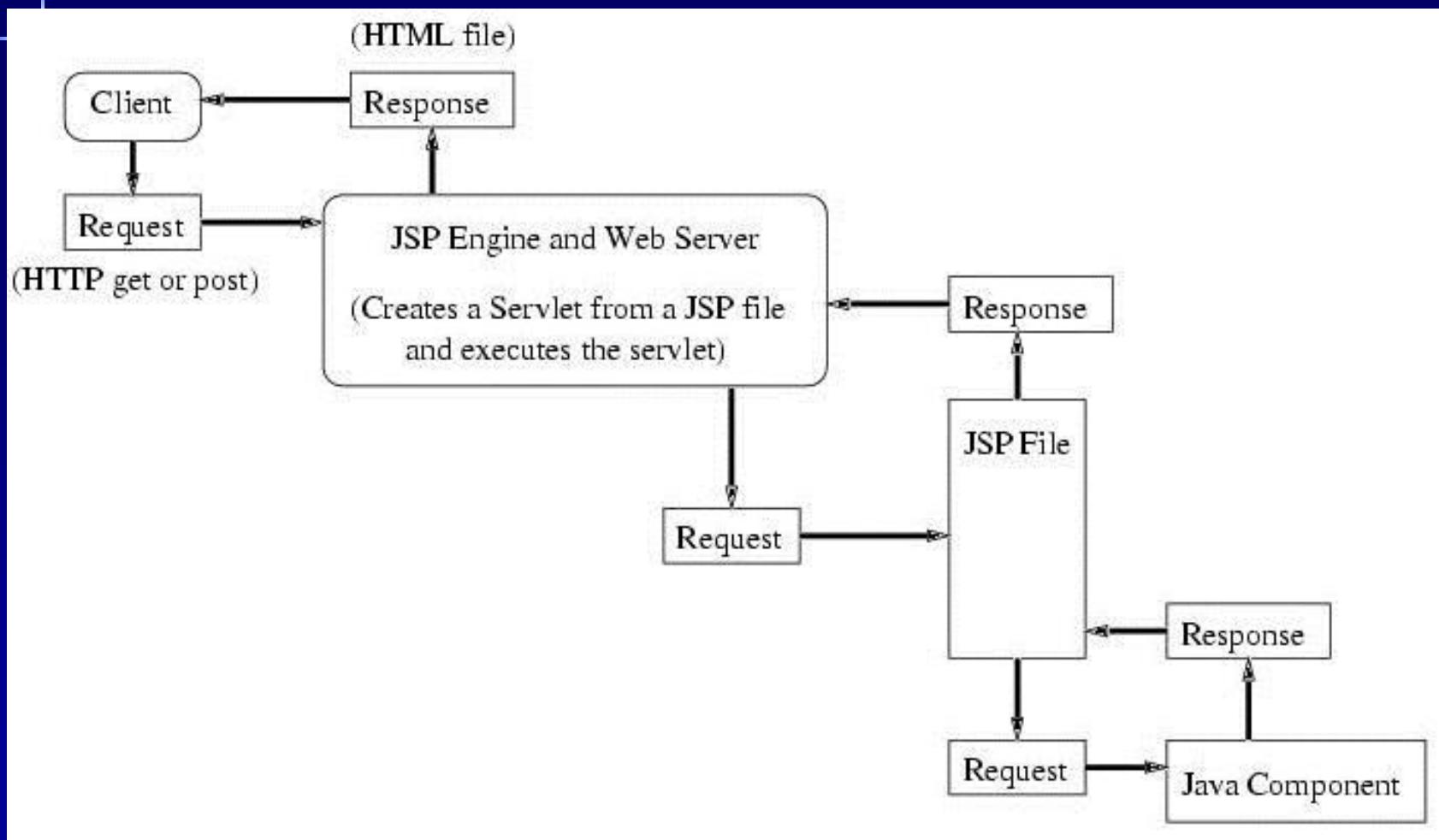
# Traducción y ejecución

- ✓ El resultado de la fase de traducción es la creación de una clase Servlet
  - ✓ JSP Page class
  - ✓ JSP Page object
- ✓ Durante la fase de ejecución el contenedor JSP entrega los eventos al objeto de implementación

# Traducción y ejecución

El contenedor JSP también se encarga de instanciar los objetos de petición y respuesta

# Ejemplo



# Servidores JSP

- ✓ Java Web server
  - ✓ Disponible para windows y unix
  - ✓ Costoso
- ✓ iPlanet
  - ✓ Costoso
  - ✓ Disponible para windows, linux, y otras variantes de unix.
- ✓ Tomcat
  - ✓ Disponible para múltiples plataformas
  - ✓ Gratuito
- ✓ GNUJsp
  - ✓ Disponible para linux
  - ✓ Gratuito
  - ✓ Funciona con Apache, Jigsaw y Vqserver

# Hablando en HTTP

## ✓ GET

- Método de HTTP que permite enviar datos a través de la URL
- Limitado en tamaño

## ✓ POST

- Método de envío de datos a través del cuerpo del mensaje

# XML-Tags

- Los tags de JSP siguen las reglas del XML – Bien formado
- Se debe utilizar el namespace **jsp**

# Tags JSP

- Jsp posee cuatro tipos de Tags
  - Directivas
    - Proporciona datos al procesamiento de la página
  - Elementos de Scripting
    - Instrucciones del lenguaje de Scripting
  - Comentarios
    - Comentarios al código fuente
  - Acciones
    - Diversos tipos de acciones, son procesados en cada request de la página.
    - Pueden proporcionar control entre páginas, con applets e interactuar con Java Beans del lado del servidor.

# Directivas JSP – page

- Proporcionan información al contenedor de la página para su procesamiento
- Ejemplo:

```
<%@page info="este es un ejemplo de directivas"
%>

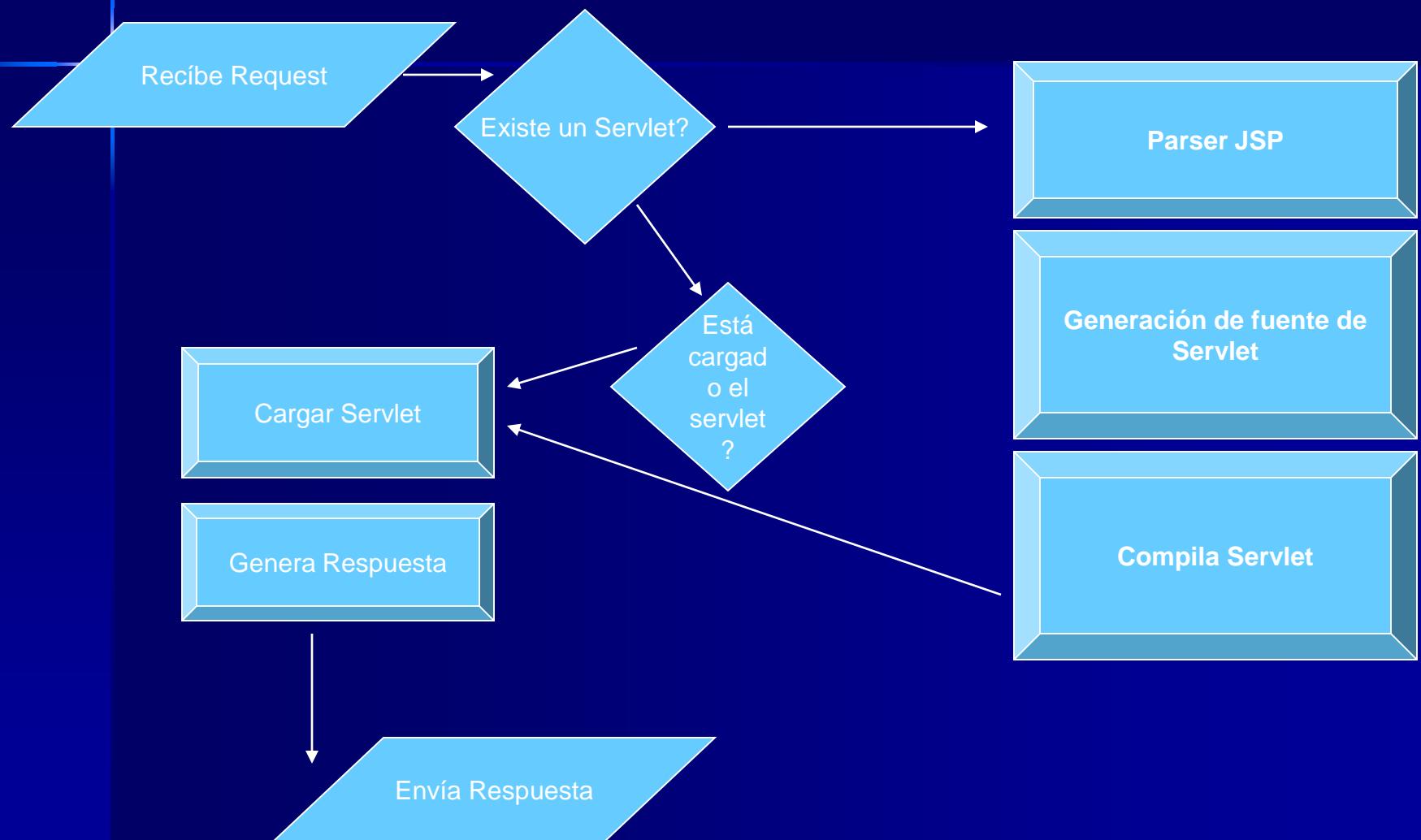
<%@page language="java" import="java.net.*
%>

<%@page import="java.util.List,
java.util.ArrayList" %>
```

# Directivas JSP – Atributo contentType

- Especifica el formato en el cual se deberá procesar la página
- Ejemplo:
  - <%@page contentType="text/xml" %>
  - <%@page contentType="text/html; charset=ISO-8859-1"%>

# Proceso de Creación de JSP



# Scripting de JSP - Directivas

- Las directivas son instrucciones especiales que indican formas de procesamiento de información
- Ejemplos:
  - <@ page contentType="text/xml" %>
  - <@ page contentType="text/xml";  
charset="ISO-8859-1" %>

# Directivas JSP - extends

- El atributo `extends` especifica la superclase que se considerará para el procesamiento de la página JSP.
- Ejemplo:
  - `<%page  
extends="com.taglib.wdjsp.myJspPage"  
%>`

# Elementos de Scripting

## Declaraciones

- Para la definición de variables se utiliza
- <%! Declaraciones %>
- En versión xml
  - <jsp:declaration>declaraciones</sp:declaration>

# Controlar el ciclo de Vida

```
<!public void jspInit(){  
    //código de inicialización  
}  
Public void jspDestroy(){  
}  
%>
```

# Declaración explícita de salida

- Se indica con <%= expresión%>
- En xml
  - <jsp:expression> expresión</jsp:expression>

# Código de Script

- Se declara <% script %>
- En xml
  - <jsp:scriptlet> scriptlet</jsp:scriptlet>
- Soporta cualquier expresión Java

# Comentarios

- Se indican con `<! - - comentario - - >` en la parte HTML
- En JSP
  - `<% -- comentario - -%>`

# Objetos Implícitos

Objeto	
page	javax.servlet.jsp.HttpJspPage
config	javax.servlet.ServletConfig
request	javax.servlet.http.HttpServletRequest
response	javax.servlet.http.HttpServletResponse
out	javax.servlet.jsp.JspWriter
session	javax.servlet.http.HttpSession
application	javax.servlet.ServletContext
pageContext	javax.servlet.jsp.pageContext
exception	Java.lang.Throwable

# Request

- Representa la data que llega desde la página
- Ejemplo:

```
<%String xStr = request.getParameter("num");  
try{  
    Long x=Long.parseLong(xStr);%>  
    Factorial result : <%= x %> != <%= fact(x) %>  
}catch(NumberFormatException e) {%->
```

Sorry, the **num** parameter not specify an integer value  
<%}%>

# Request – Métodos

## **javax.servlet.http.HttpServletR equest**

- getParameterName()
- getParameter(name)
- getParameterValues(name)
- getHeaderNames()
- getDateHeader()
- getMethod()
- getRemoteHost()
- Para otros métodos ver documentación de j2se.

# Response

- Representa el resultado para una página

# Response – Métodos

## `javax.servlet.http.HttpServletResponse`

- `setContentType()`
- `getCharacterEncoding()`
- `addCookie(cookie)`
- Para otros métodos ver documentación de j2se.

# Out

- Objeto incorporado y activo en JSP
- Representa a la clase  
javax.servlet.jsp.JspWriter
- Ejemplo:

```
<%int count = 0;  
While(carton.has.next()){  
    count++;  
    out.print(".");  
}%>
```

# Session

- Representa la sesión individual de un usuario
- Guarda información asociada a la sesión del usuario

# Session

- getId()
- getCreationTime()
- getLastAccessedTime()
- getMaxInactiveInterval()
- setMaxInactiveInterval()
- isNew()
- invalidate()
- setAttribute(attribute)
- getAttribute()

# Application

- Permite mantener datos a nivel del contexto de aplicación
- Es visible por todas las sesiones

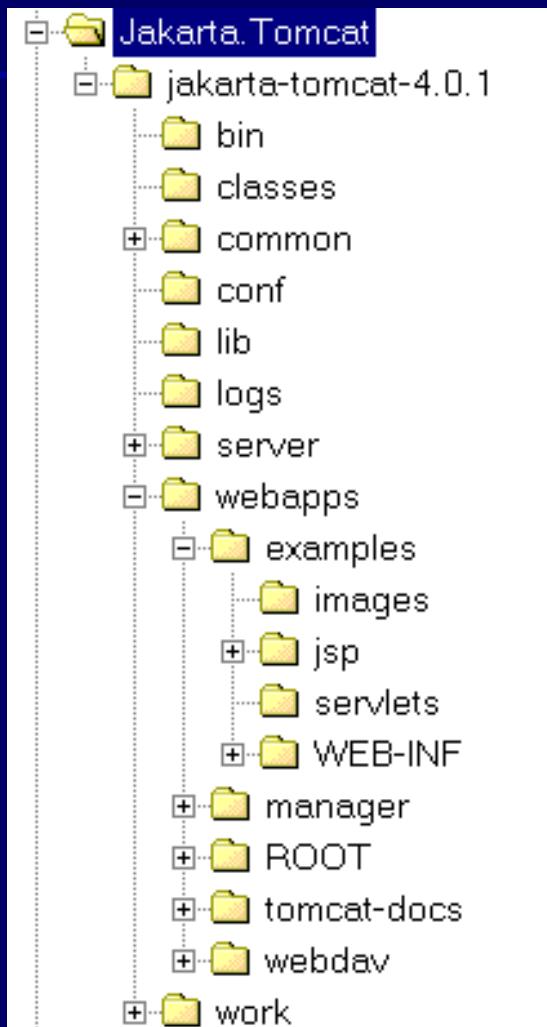
# Cookies

- Para utilizar Cookies debe instanciar la clase Cookies
- Ejemplo:
  - Cookie cookie=new Cookie("Favorite","chocolate chips");

# Ejemplo Cookies

```
<html>
<head>
<title>Cookie Page</title>
</head>
<%@page import="java.util.Date"%>
<%@page import="java.net.*"%>
<%String cookieName="RedCookie";
  Date now=new Date();
  String timestamp=now.toString();
  Cookie cookie=new Cookie(cookieName,URLEncoder.encode(timestamp));
  cookie.setDomain("tichile.cl");
  cookie.setPath("/webdev");
  cookie.setMaxAge(7*24*60*60);//Una Semana
  response.addCookie(cookie);
%>....
```

# Deployment



Directorio de Aplicaciones web

Esta estructura se puede comprimir en un archivo jar con extensión war y ser instalada en forma directa en un servidor Web

web.xml

# Primera página JSP

```
<%@page import="java.util.*"%>
<%! String cadena="Bienvenidos a JSP"; %>
<html>
    <body>
        <%= cadena %>
        <br>
        <% out.println(new Date()); %>
    </body>
</html>
```