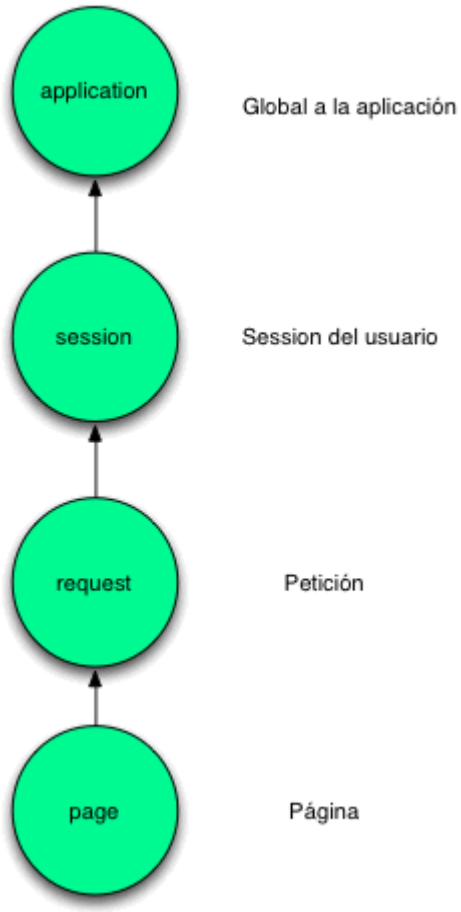
Cuando usamos JSP tenemos varios objetos implícitos que nos permiten acceder a las distintos scopes o ámbitos que una página tiene. En concreto disponemos de los siguientes :



El scope de Application es compartido por todos los elementos de la aplicación , el scope de Session pertenece a las variables que almacena cada usuario. Por otro lado el scope de petición tiene un ciclo de vida mucho más corto ya que pertenece a las páginas que están ligadas a la misma petición. Por último el scope de page que hace referencia a variables que se encuentran definidas en la propia página. Vamos a verlos todos en acción a través del uso del siguiente servlet.

```
package com.arquitecturajava;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
* Servlet implementation class ServletsScopes
*/
@WebServlet(&quot;/ServletsScopes&quot;)
public class ServletsScopes extends HttpServlet {
private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

ServletContext aplicacion= request.getServletContext();
aplicacion.setAttribute(&quot;variableAplicacion&quot;,
&quot;holaAplicacion&quot;);
HttpSession session=request.getSession(true);
session.setAttribute(&quot;variableSession&quot;,
&quot;holaSession&quot;);
request.setAttribute(&quot;variablePeticion&quot;,
&quot;holaPeticion&quot;);
```

```
RequestDispatcher despachador=
request.getRequestDispatcher(&quot;/scopes.jsp&quot;);
despachador.forward(request, response);

}

}
```

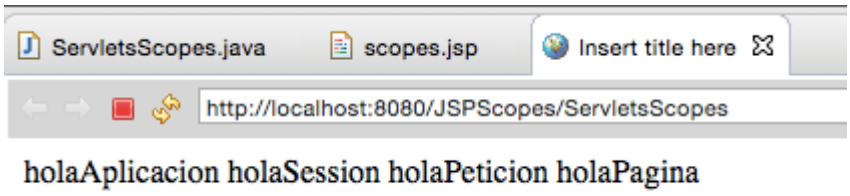
El Servlet se encarga de almacenar los valores en cada uno de los ámbitos para finalmente redireccionarnos a una página JSP que contiene el siguiente código:

```
&lt;%@ page language=&quot;java&quot; contentType=&quot;text/html;
charset=UTF-8&quot;
pageEncoding=&quot;UTF-8&quot;%&gt;
&lt;%@taglib uri=&quot;http://java.sun.com/jsp/jstl/core&quot;
prefix=&quot;c&quot; %&gt;
&lt;!DOCTYPE html PUBLIC &quot;-//W3C//DTD HTML 4.01
Transitional//EN&quot;
&quot;http://www.w3.org/TR/html4/loose.dtd&quot;&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta http-equiv=&quot;Content-Type&quot; content=&quot;text/html;
charset=UTF-8&quot;&gt;
&lt;title&gt;Insert title here&lt;/title&gt;
&lt;/head&gt;
&lt;c:set var=&quot;variablePagina&quot;
value=&quot;holaPagina&quot;/&gt;
&lt;body&gt;
${applicationScope.variableAplicacion}
```

```
${sessionScope.variableSession}
${requestScope.variablePeticion}
${pageScope.variablePagina}

&lt;/body&gt;
&lt;/html&gt;
```

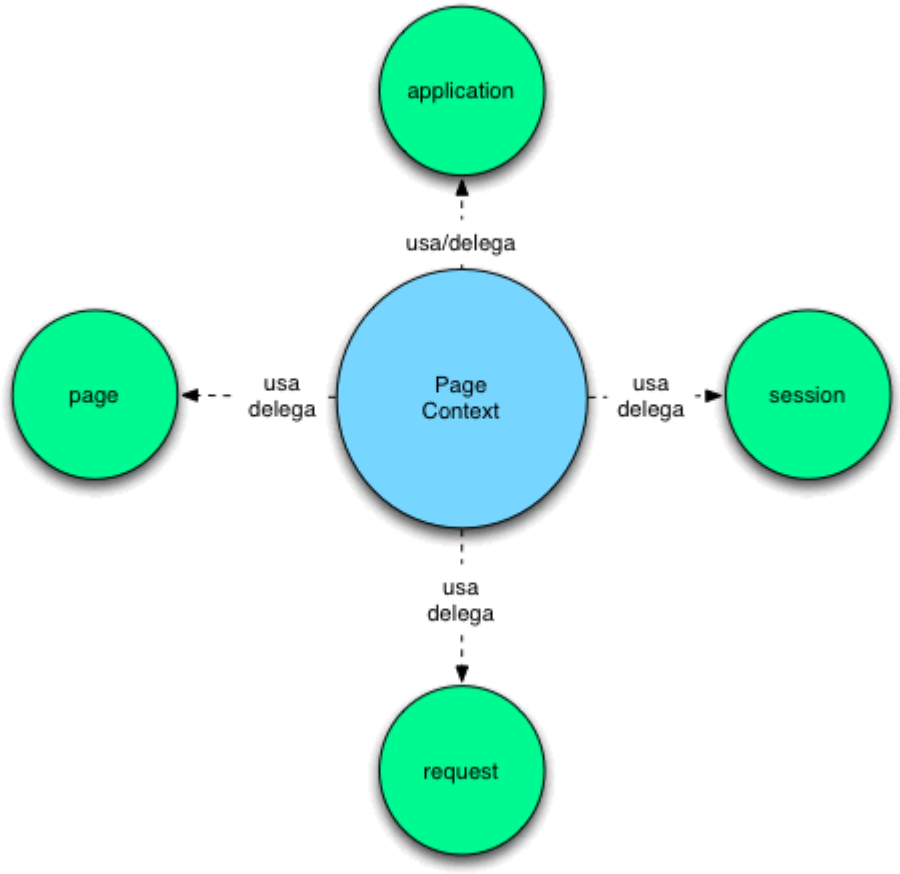
En la página JSP podemos ver como accedemos a cada una de las variables que se encuentran en los distintos ámbitos e imprimimos su valor por pantalla. Hemos añadido además a través del uso de las librerías de etiqueta de JSTL una variable para rellenar el ámbito de página,el resultado será el siguiente:



Se han almacenado sin problemas las diferentes variables en los diferentes ámbitos.

¿Qué es el PageContext?

Una de las dudas más habituales es si PageContext es también un ámbito. La respuesta es NO, PageContext es una clase que aglutina un conjunto de operaciones que se ejecutan en los diferentes ámbitos. En ese sentido funciona como una fachada/adaptador a la hora de darnos acceso a diferentes características de la aplicación.



Podríamos acceder a los mismos datos utilizando el PageContext

```
${pageContext.servletContext.getAttribute(&amp;quot;variableAplicacion&amp;quo
t;)}
${pageContext.session.getAttribute(&amp;quot;variableSession&amp;quot;)}
${pageContext.request.getAttribute(&amp;quot;variablePeticion&amp;quot;)}
${variablePagina}
```

De igual forma podemos acceder a información adicional como por ejemplo:

```
${pageContext.servletContext.contextPath}
${pageContext.session.id}
```

Otros artículos relacionados : Usando ServletContext , ServletContextListener ,ServletLifeCycle