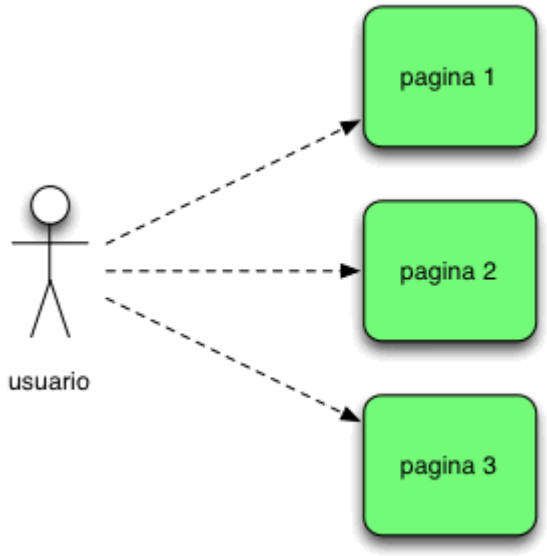
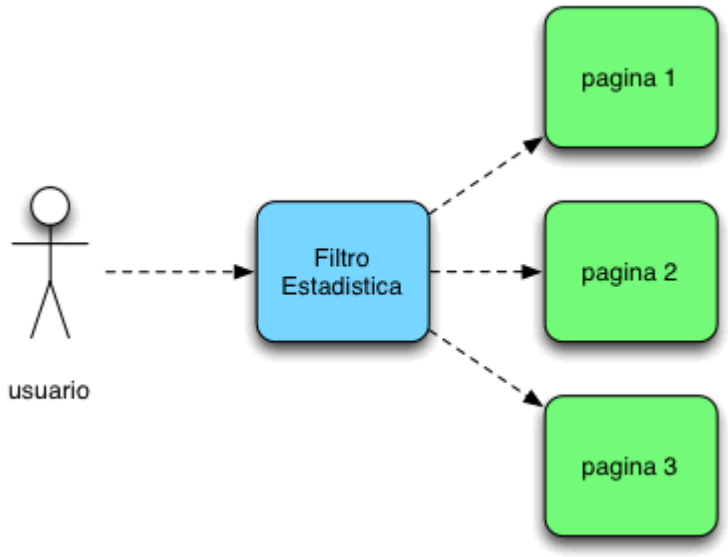
Cuando empezamos a trabajar un Java EE se nos presentan muchos conceptos nuevos ,a veces demasiados pero uno de los más clásicos es el concepto de java filter o filtro java .Practicamente todos los frameworks de capa de presentación los usan de una forma o de otra para aportar nuevas funcionalidad extendiendo de las ya disponibles . Ahora bien ¿como funcionan? y sobre todo para que sirven. Vamos a partir de la idea que tenemos una aplicación web con tres páginas.



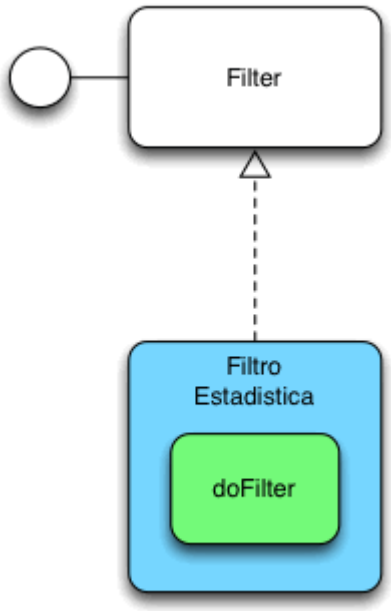
Java Filter

Un filtro de java se encarga de añadir una nueva funcionalidad a la aplicación colocandose entre el usuario y las páginas .Por ejemplo en nuestro caso vamos a implementar un sencillo filtro que construye unas estadisticas sobre cuantas veces se ha visitado cada página



Java Filter (Construcción)

Definir un filtro en Java es bastante sencillo sobre todo a partir de Servlet 3.0 que nos podemos apoyar en el sistema de anotaciones. Un Filtro implementa la interface Filter de JEE que aporte el método doFilter.



Vamos a ver él código de este filtro :

```
</pre>
ackage com.arquitecturajava;

import java.io.IOException;
import java.util.HashMap;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
```

```
@WebFilter(filterName="/FiltroEstadistica",urlPatterns="/*" )
public class FiltroEstadistica implements Filter {

 public void destroy() {
 // TODO Auto-generated method stub
 }
 @SuppressWarnings("unchecked")
 public void doFilter(ServletRequest request, ServletResponse
response, FilterChain chain) throws IOException, ServletException {
 // TODO Auto-generated method stub
 // place your code here

HttpServletRequest peticion= (HttpServletRequest)request;
 ServletContext contexto= request.getServletContext();

 HashMap<String, Integer> urls;
 if (contexto.getAttribute("estadistica")==null) {
 //creamos un objeto en el contexto
 urls= new HashMap<String,Integer>();

 urls.put(peticion.getRequestURL().toString(), 1);
 contexto.setAttribute("estadistica", urls);
 }else {

 // actualizamos claves e incrementamos
 urls=(HashMap<String,Integer>)contexto.getAttribute("estadistica");

 if (urls.get(peticion.getRequestURL().toString())==null) {

 urls.put(peticion.getRequestURL().toString(), 1);
 }else {
```

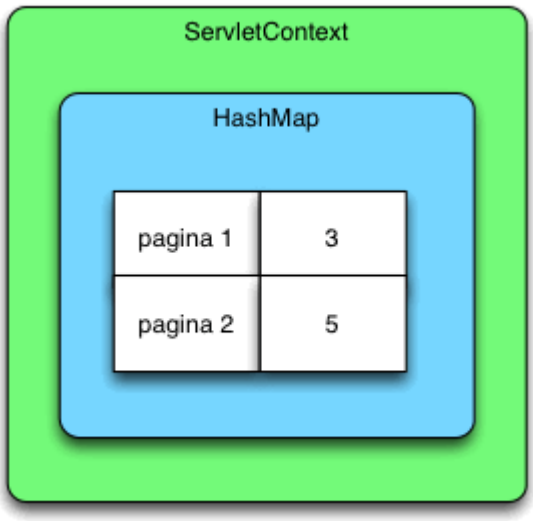
```
 urls.put(peticion.getRequestURL().toString(),
urls.get(peticion.getRequestURL().toString())+1);
 }
 }

 chain.doFilter(request, response);
 }

/**
 * @see Filter#init(FilterConfig)
 */
 public void init(FilterConfig fConfig) throws ServletException {
 // TODO Auto-generated method stub
 }

}
<pre>
```

El filtro es relativamente sencillo .Se encarga de almacenar un HashMap o diccionario en el contexto de de aplicación (ámbito de aplicación). Este mapa contendrá el nombre de la página y las veces que hemos accedido a ella .



De esta forma será luego muy sencillo construir un JSP básico que accede al contexto (ServletContext) y se encargue de imprimir por pantalla la lista de páginas ya acceso que ha guardado . Vamos a ver su código a traves del uso de scriptlet de JSP que aunque no es la mejor práctica sirve para clarificar .

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%>
 <%@page import="java.util.HashMap" %>

<%@page import="java.util.Set" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
 <html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <title>Insert title here</title>
 </head>
```

```
 <body>
 <%

HashMap<String,Integer>
mapa=(HashMap<String,Integer>)application.getAttribute("estadistica");

Set<String> conjunto=mapa.keySet();

for (String clave :conjunto) {

out.println("pagina :" +clave +" visitas :"+mapa.get(clave)+"<br/>");

}
 %>
 </body>
 </html>
```

Realizada esta operación el JSP nos mostrará lo siguiente :



El código : JavaFiltros