

Práctica 3  
Macedo Borbolla Eduardo Sai  
Filtros  
Web Application Development  
18 de Septiembre del 2018

### Resumen

Reporte del desarrollo de un Filtro de IP, hora y fecha.

**Teoría:** Los filtros son clases Java que nos permiten interceptar una petición y modificarla antes de que sea procesada por el Servlet, o interceptar una respuesta antes de que la misma sea enviada al cliente, para crear estas clases es necesario implementar la interface `javax.servlet.Filter`, la cual define los siguientes tres métodos que debemos implementar:

- init(FilterConfig cfg):** Es invocado por el contenedor una vez que el filtro entra en funcionamiento.
- Destroy():** Este método lo llama el contenedor cuando el filtro de Servlet deja de funcionar y se deja al recolecto de basura de Java.
- doFilter(ServletRequest req, ServletResponse res, FilterChain chain).** El contenedor llama a este método para cada solicitud de Servlet correlacionada con este filtro antes de invocar el Servlet. El Servlet original solicitado se ejecuta cuando el filtro llama al método `chain.doFilter(...)`.

### Desarrollo:

1.-Implementar un filtro que intercepte cualquier petición que se haga a la aplicación web e imprima en log: `Filter: IP fecha y hora método recurso.`

## Código:

```
package puntos;

import java.io.IOException;
import java.util.Date;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletResponse;

public class FiltroIP implements Filter {

    public FiltroIP() {
        super();
    }

    public void destroy() {
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
chain) throws IOException, ServletException {
        String ip= request.getRemoteAddr();
        Date date= new Date();
        HttpServletResponse httpResponse = (HttpServletResponse) response;
        System.out.println("Filter:"+ip+ " " + date);

        chain.doFilter(request, response);
        //      httpResponse.sendRedirect("/practica3");

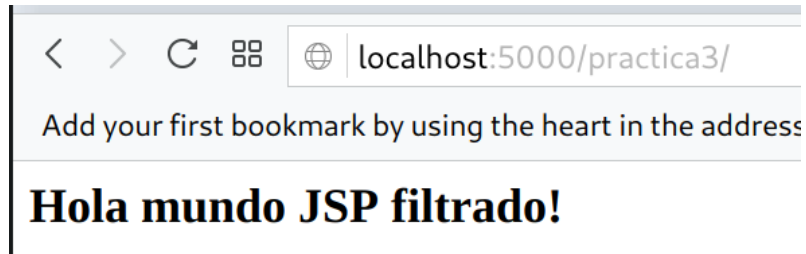
    }

    public void init(FilterConfig fConfig) throws ServletException {
    }

}
```

Al desarrollar este filtro solo me encontré con la confusión de recibir la IP 0:0:0:0 que al no recordar que era la local dado que la petición es local me tuvo pensando un rato.

Al ejecutar en nuestro entorno tenemos la siguiente vista del lado del cliente.



Y el resultado del filtro impreso en consola del servidor.

```
Filter:0:0:0:0:0:0:0:1 Mon Oct 01 10:07:59 CDT 2018
```

### **Conclusiones:**

El desarrollo de filtros en la arquitectura Java EE permite el monitoreo de acceso y gestión de recursos de todo tipo desde el despliegue de la aplicación lo cual nos permite ahorrar trabajo como desarrolladores Web ya que no necesitamos crear instancias de nuevas clases (Servlets por ejemplo) con las cuales sería más complicado programar las funciones que vienen con una implementación sencilla en la interfaz Filter.

### **Bibliografía:**

**Curso Web Application Development 2019-1. M. en C. Hermes Montes Casiano**