

Text("Hello".i18n)

"Hola"

"Salut"

"Hallo"

"Olé"



i18n, Filtros y Eventos

- añadiendo capacidades a mi aplicación -

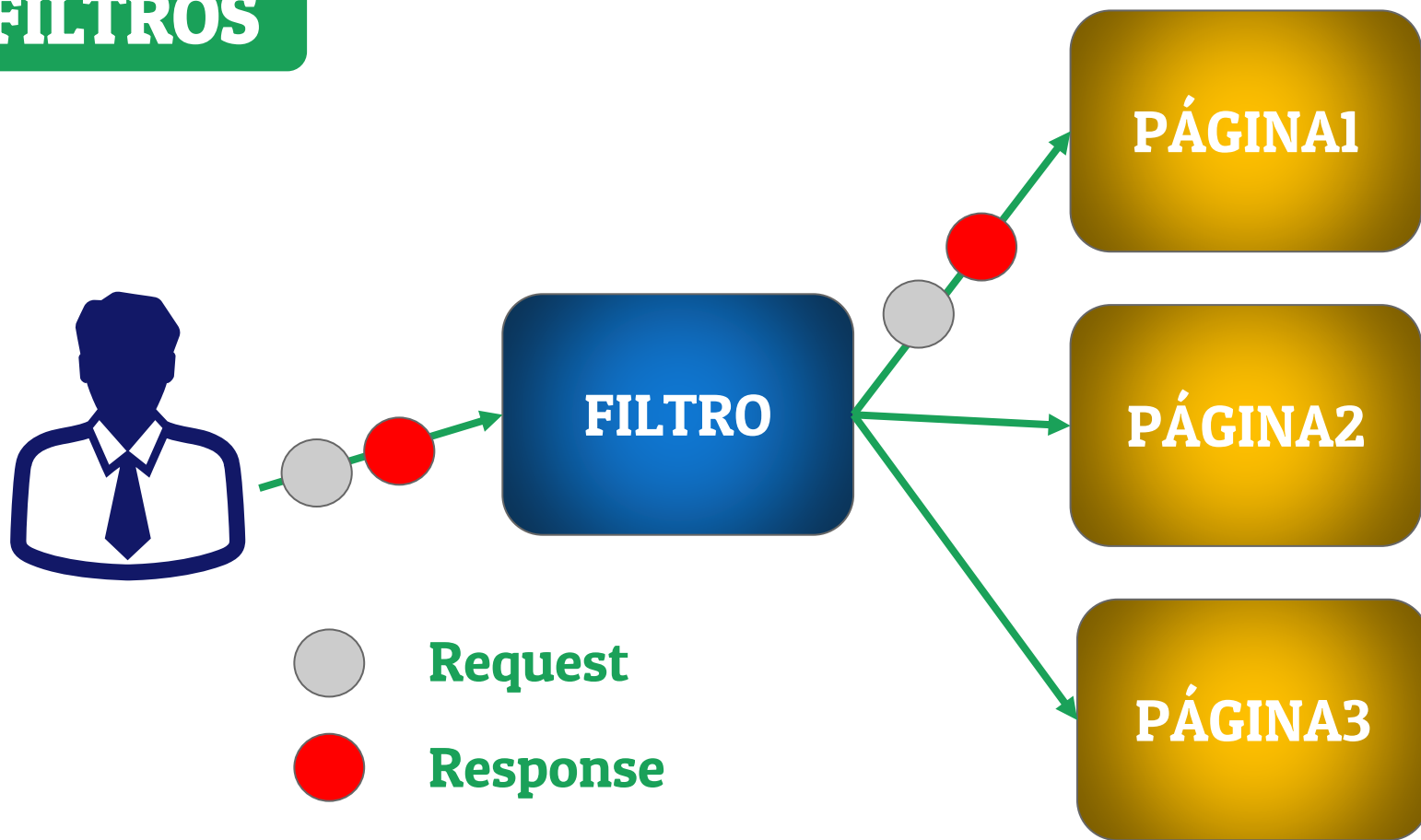
ÍNDICE DE CONTENIDOS

- **Filtros.**
- **Eventos.**
- **il8n.**
- **Ocultando los ficheros JSP.**
- **Referenciando imágenes.**

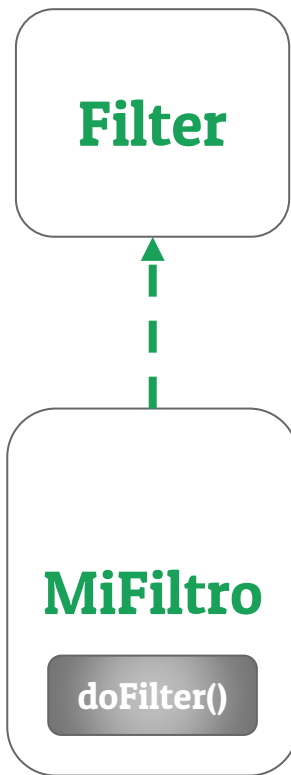
FILTROS

“ **Funcionalidad** que se coloca **ENTRE** el usuario y las páginas de mi aplicación de tal forma que **antes** de “acceder” a esas páginas se ejecutan ciertas **operaciones.** ”

FILTROS



FILTROS



```
@WebFilter(filterName="nombreFiltro",urlPatterns="...")
public class MiFiltro implements Filter {

    @Override
    public void doFilter(request, response, FilterChain chain) {
        chain.doFilter(request, response);
    }
}
```

Atributos de @WebFilter (sin orden)

Nombre	Tipo	Requerido	Descripción
filterName	String	Optional	Nombre del filtro
value / urlPatterns	String[]	Required	Patrón para las urls
servletNames	String[]	Optional	Nombres de los servletes a los que se va a aplicar
displayName	String[]	Optional	Nombre que se va a mostrar el servlet
description	String	Optional	Descripción

Atributos de @WebFilter

Nombre	Tipo	Requerido	Descripción
asyncSupported	boolean	Optional	Si soporta modo asíncrono
initParams	WebIniParam[]	Optional	Parmámetros de inicialización
smallIcon	String	Optional	Icono
largeIcon	String	Optional	Icono
dispatcherTypes	DispatcherType[]	Optional	SI se aplica a Request o Response

EJEMPLOS de @WebFilter

- `@WebFilter("/admin")`
- `@WebFilter("/*")`
- `@WebFilter(servletNames="MyServlet")`
- `@WebFilter(servletNames={"MyServlet1","MyServlet2"})`
- `@WebFilter(
 urlPatterns = "/uploadFilter",
 initParams = @WebInitParam(name = "fileTypes", value =
 "doc;xls;zip;txt;jpg/png/gif")
)`
- `@WebFilter(
 urlPatterns = "/admin/*",
 filterName = "AdminFilter",
 description = "Filter all admin URLs"
)`

Filtros en el descriptor de despliegue (Puedo especificar el orden)

```
<filter>
  <filter-name>MiFiltro</filter-name>
  <filter-class>velazquez.MiFiltro</filter-class>

  <init-param>
    <param-name>message</param-name>
    <param-value>A message from Hello Filter</param-value>
  </init-param>

  <init-param>
    <param-name>threshold</param-name>
    <param-value>1024</param-value>
  </init-param>
</filter>
```

Filtros en el descriptor de despliegue (Puedo especificar el orden)

```
<filter-mapping>  
  <filter-name>MiFiltro</filter-name>  
  <servlet-name>ServletPrincipal</servlet-name>  
</filter-mapping>
```

```
<filter-mapping>  
  <filter-name>MiFiltro</filter-name>  
  <url-pattern>/hello</url-pattern>  
  <url-pattern>/bonjour</url-pattern>  
  <url-pattern>/ciao</url-pattern>  
</filter-mapping>
```

```
<filter-mapping>  
  <filter-name>MiFiltro</filter-name>  
  <url-pattern>/*</url-pattern>  
</filter-mapping>
```

¿QUÉ PUEDO HACER CON LOS FILTROS?

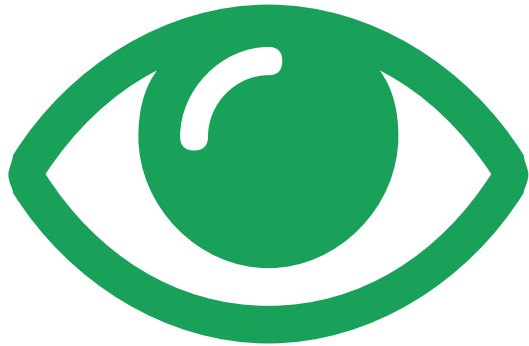
REQUEST

- Security Checks
- Modificar cabeceras y cuerpo
- Auditar y logear peticiones

RESPONSE

- Comprimir el stream.
- Añadir o modificar la respuesta.
- Crear una respuesta nueva.

FILTROS



EVENTOS

“ Un evento es la “ocurrencia” de algo.

En el ámbito de los Servlets ese algo es la modificación de algún objeto. ”

TIPOS DE EVENTOS



1. **ServletRequestEvent**

2. **ServletContextEvent**



3. **ServletRequestAttributeEvent**



4. **ServletContextAttributeEvent**



5. **HttpSessionEvent**

6. **HttpSessionBindingEvent**

EVENTOS

“ Para responder a los eventos tenemos que implementar “LISTENERS” que implementen ciertos interfaces específicos para cada evento. ”

LISTENERS DE EVENTOS

1. **ServletRequestListener**
2. **ServletRequestAttributeListener**
3. **ServletContextListener**
4. **ServletContextAttributeListener**
5. **HttpSessionListener**
6. **HttpSessionAttributeListener**
7. **HttpSessionBindingListener**
8. **HttpSessionActivationListener**

EVENTOS



“ Preparar nuestra aplicación para que tenga soporte para varios idiomas. ”

il8n - Internacionalización

PASOS PARA il8n

1. Crear los ficheros necesarios.
2. Rellenar los ficheros con las traducciones adecuadas.
3. Traducir en el Servlet o Traducir en el JSP
(depende lo que necesitemos).

Requisito: Añadir librerías al POM

il8n - Internacionalización

CREAR los ficheros necesarios

```
/Project_ROOT/  
  |----/resources/  
    |-----interface.properties  
    |-----interface_es.properties  
    |-----interface_XX.properties
```

XX-> <https://github.com/michaelwittig/node-il8n-iso-countries>

RELLENAR los ficheros

Por defecto

date= Date

title= Title

user= User

home= Home

Traducción/ES

date= Fecha

title= Título

user= Usuario

home= Inicio

Traducción/FR

date= Date

title= Titre

user= Nom

home= Page d'accueil

i18n - Internacionalización

```
Locale locale = request.getLocale();  
ResourceBundle bundle = ResourceBundle.getBundle(resourceName, locale);  
String welcome = bundle.getString("usuario");
```

TRADUCIR desde el JSP

```
<%@taglib prefix="fmt" uri="http://java.sun.com/jstl/fmt_rt" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<fmt:setBundle basename="interface" />

<fmt:message key="usuario" />
```

Ocultando los ficheros .jsp

“ **A DÍA DE HOY** nuestros ficheros jsp (las vistas) están recibiendo datos de los servlet pero siguen siendo directamente accesibles por URL ”

Ocultando los ficheros .jsp



PROBLEMA

Si accedemos directamente y no tenemos esos datos puedo obtener errores, por ejemplo valores esperados que son nulos etc...

Ocultando los ficheros .jsp



SOLUCIÓN

- Creamos la carpeta view dentro de /WEB-INF/
- Colocamos los ficheros JSP dentro de esa carpeta.
- Modificamos las rutas de los JSP añadiendo WEB-INF

**NO PUEDO ACCEDER DE MANERA DIRECTA A LOS JSP
DENTRO DE WEB-INF**



END



jgardur081@g.educaand.es