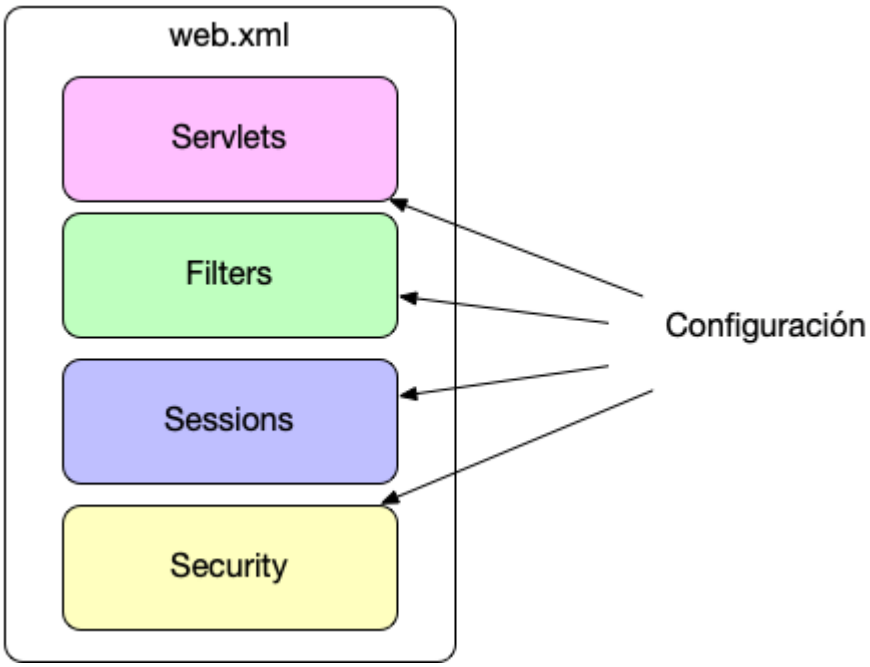
Tabla de Contenidos

- Web.xml y paginas por defecto
- Servlets y mapeos
- Filtros y Web.xml
- Seguridad en entornos Java EE con web.xml
- Otros artículos relacionados

El web.xml o web deployment descriptor es el fichero de configuración que usa Java EE para el despliegue de aplicaciones web. Esto nos puede sonar un poco raro pero la realidad es que simplemente es el fichero que se encarga de configurar una aplicación web cuando la desplegamos sobre un Tomcat , WebSphere etc.



Es muy común tener que realizar alguna modificación en el para el correcto funcionamiento de la aplicación. El fichero esta lleno de secciones.

Web.xml y paginas por defecto

Una de las configuraciones más sencillas y típicas es la de las páginas por defecto:

```
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>
```

Estas vienen delimitadas por la etiqueta welcome-file-list y marcan que página se mostrará por defecto al arrancar la aplicación .

Servlets y mapeos

Otra de las secciones más clásicas es aquella que da de alta Servlets y mapeos de estos para su posterior ejecución.

```
  <servlet>
        <servlet-name>miservlet</servlet-name>
        <servlet-class>com.arquitecturajava.MiServlet</servlet-class>
        <init-param>
            <param-name>contador</param-name>
            <param-value>5</param-value>
        </init-param>
    </servlet>
```

Esta configuración es un poco peculiar ya que se divide en dos secciones. La primera define el servlet y los parámetros de acceso a este. La segunda por otra parte define su mapeo.

```
 <servlet-mapping>
        <servlet-name>miservlet</servlet-name>
        <url-pattern>/miservlet</url-pattern>
```

```
    </servlet-mapping>
```

Recordemos que hoy por hoy muchos de los mapeos de Servlets se hacen a través de las anotaciones @WebServlet y podemos omitirlos del web.xml

Filtros y Web.xml

Otra de las cosas que el web.xml configura de forma muy habitual son filtros de acceso a determinadas urls.

```
    <filter>
        <filter-name>mifiltro</filter-name>
        <filter-class>com.arquitecturajava.mifiltro</filter-class>
    </filter>
```

De igual manera esta dividido en dos bloques:

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

Seguridad en entornos Java EE con web.xml

Otra de las secciones más típicas es la sección dedicada a temas de seguridad en la cual se definen los roles y urls que están protegidos para los diferentes usuarios.

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>zona1</web-resource-name>
        <url-pattern>/zona1</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
```

```
    </web-resource-collection>
    <auth-constraint>
        <role-name>Administrador</role-name>
    </auth-constraint>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Todas estas etiquetas y muchas más definen la configuración completa de una aplicación web con el web.xml y nos ayudan en el correcto arranque de la aplicación . Recordemos que este fichero siempre se ubica en /WEB_INF y es opcional para las ultimas versiones de Java EE.

Otros artículos relacionados

- Java WebApp y su estructura
- El concepto de EJB in WAR y su uso
- Java Packages vs JARs y su reutilización
- fichero de despliegue