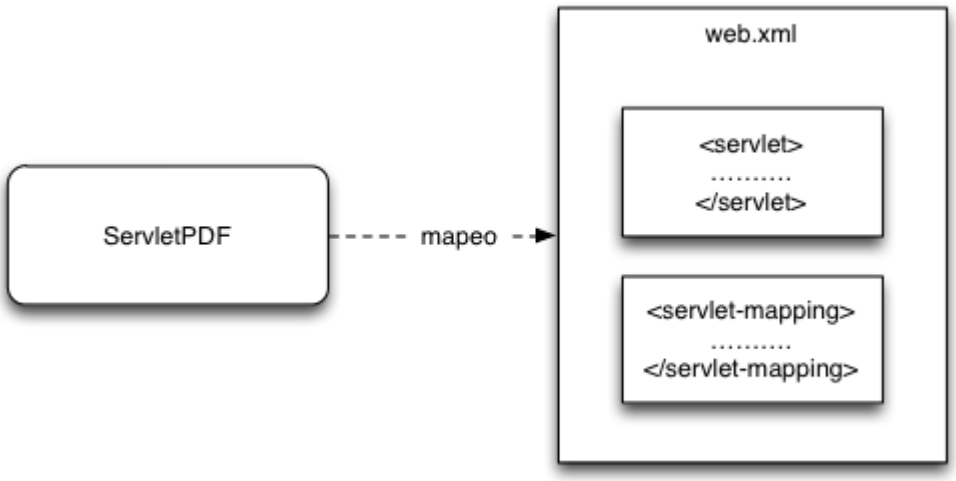
En muchas ocasiones hemos construido servlets para desarrollar una u otra funcionalidad .Para ello hemos construido una clase que extiende de HttpServlet sobrecargando métodos como doGet . Hecho esto hemos mapeado el servlet a nivel de web.xml

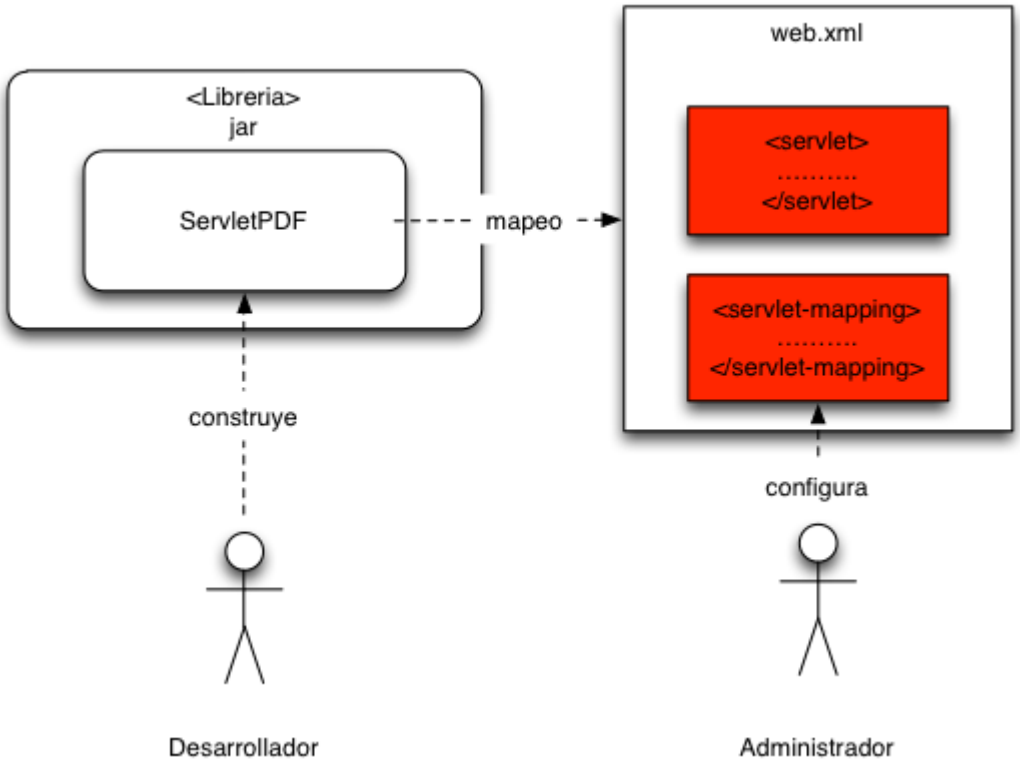


Aunque se trata de una operación sencilla siempre ha tenido algunos problemas añadidos . Por ejemplo imaginemonos que deseamos que nuestro servlet (ServletPDF) sea un Servlet reutilizable en varias aplicaciones .Para ello ubicaremos el servlet en una libreria (.jar) de tal forma que la libreria pueda desplegarse en varios proyectos.

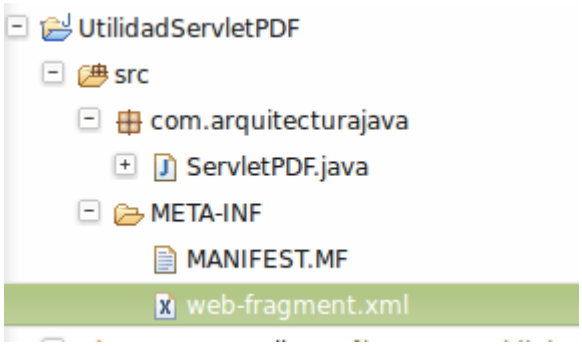


Lamentablemente aunque deseemos que esta libreria se despliegue de forma automatica en varias aplicaciones web tenemos un problema .Deberemos modificar el fichero web.xml para dar de alta al servlet tarea que normalmente realiza un administrador. Por lo tanto no

estamos ante un proceso automatico ni mucho menos.



A partir de Servlet 3.0 estos problemas se pueden resolver usando un nuevo concepto denominado WebFragment que hace referencia un nuevo tipo de fichero denominado web-fragment.xml que se puede configurar en el directorio META-INF de nuestra libreria JAR. A continuación se muestra una imagen de un proyecto de libreria (UtilidadServletPDF) que incluye el servlet y el fichero web-fragment.xml

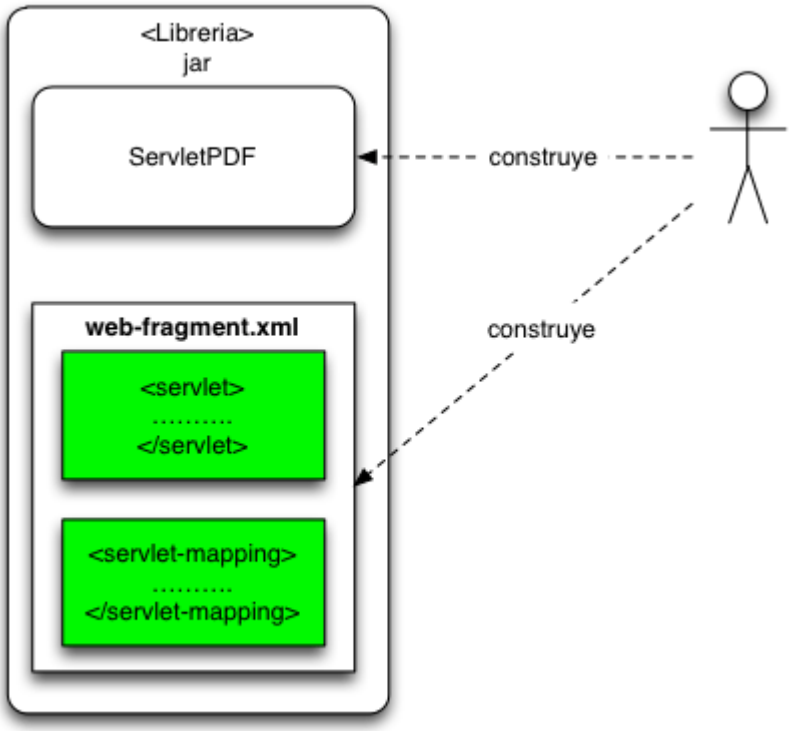


Una vez tenemos clara la estructura de este proyecto. Vamos a ver el contenido del fichero web-fragment.xml.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-fragment xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-fragment_3_0.xsd"
 version="3.0">
 <servlet>
 <servlet-name>ServletPDF</servlet-name>
 <servlet-class>com.arquitecturajava.ServletPDF</servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>ServletPDF</servlet-name>
 <url-pattern>/ServletPDF</url-pattern>
 </servlet-mapping>
</web-fragment>
```

Como podemos ver es un fichero prácticamente idéntico al web.xml pero que su etiqueta principal se denomina <web-fragment> y se encuentra ubicado en la propia libreria jar que

el desarrollador construye.



Una vez construido el web-fragment.xml vamos a ver el contenido del servlet .

```
package com.arquitecturajava;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletPDF extends HttpServlet {
```

```
/**
 *
 */
 private static final long serialVersionUID = 1L;

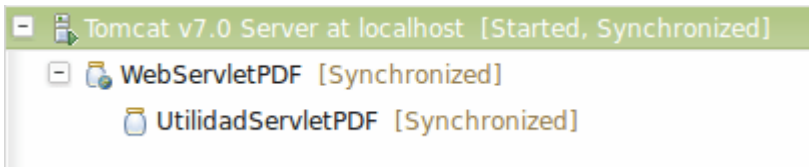
@Override
 protected void doGet(HttpServletRequest req, HttpServletResponse
resp)
 throws ServletException, IOException {

 PrintWriter pw= resp.getWriter();

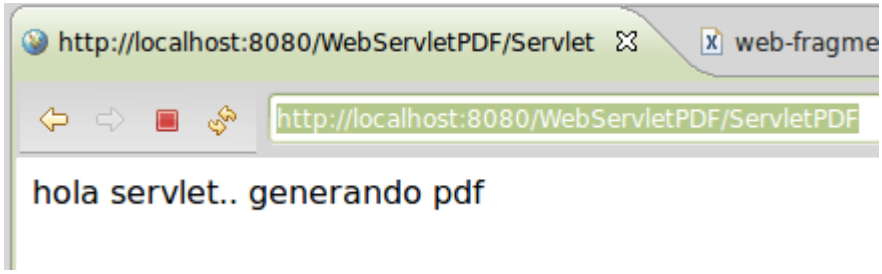
 pw.print("<html>");
 pw.print("<body>");
 //pseudocodigo
 pw.print("hola servlet.. generando pdf");
 pw.print("</body>");
 pw.print("</html>");
 pw.close();

 }
<pre>
```

Ahora podemos añadir esta libreria a un proyecto web y desplegarlo en el Tomcat



Sin tener que modificar el fichero web.xml para nada podremos acceder al Servlet ya que la librería contenía el web-fragment necesario.


http://localhost:8080/WebServletPDF/Servlet
web-fragme
http://localhost:8080/WebServletPDF/ServletPDF
hola servlet.. generando pdf