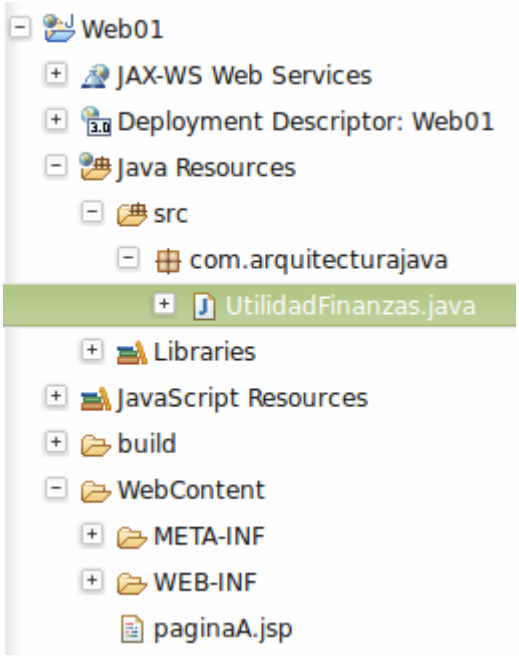
Cuando trabajamos con eclipse en muchas ocasiones nos encontramos utilizando un conjunto de clases de apoyo . Muchas veces se las denomina utilidades ,helpers o algo similar .Son las tipicas clases que vamos construyendo en nuestro proyecto web pero que nunca acabamos refactorizando en un JAR y que probablemente nos serian utiles en otros proyectos .Para solventar esta problemática Eclipse nos provee de los “Utility proyects” .Vamos a ver a continuación como utilizarlos . Para ello partiremos de un proyecto Web que tiene una clase de utilidad denominada “UtilidadFinanzas” y una página web “paginaA.jsp”



Vamos a mostrar a continuación el código fuente de esta clase.

```
package com.arquitecturajava;
public class UtilidadFinanzas {

private double iva;

public UtilidadFinanzas() {
```

```
 iva = 21;
 }

public double getIva() {
 return iva;
 }

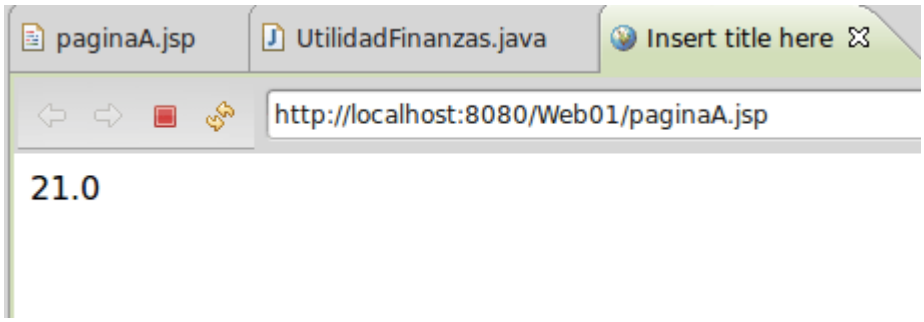
public void setIva(double iva) {
 this.iva = iva;
 }
 }
```

Esta clase que acabamos de definir es usada por “paginaA.jsp”

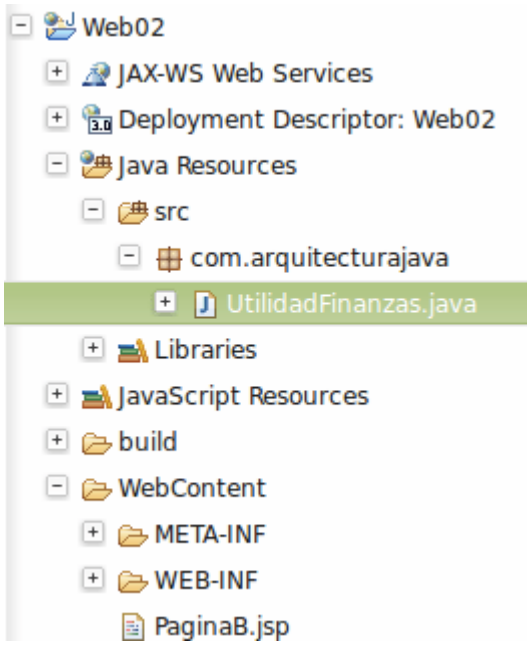
<%@ page language=”java” contentType=”text/html; charset=UTF-8″
pageEncoding=”UTF-8″%>

<!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>
<jsp:useBean id=”utilidad” class=”com.arquitecturajava.UtilidadFinanzas”>
<jsp:getProperty name =”utilidad” property = “iva” />
</jsp:useBean>
</body>
</html>

Si invocamos la pagina web a traves de un navegador nos imprimirá el iva por pantalla

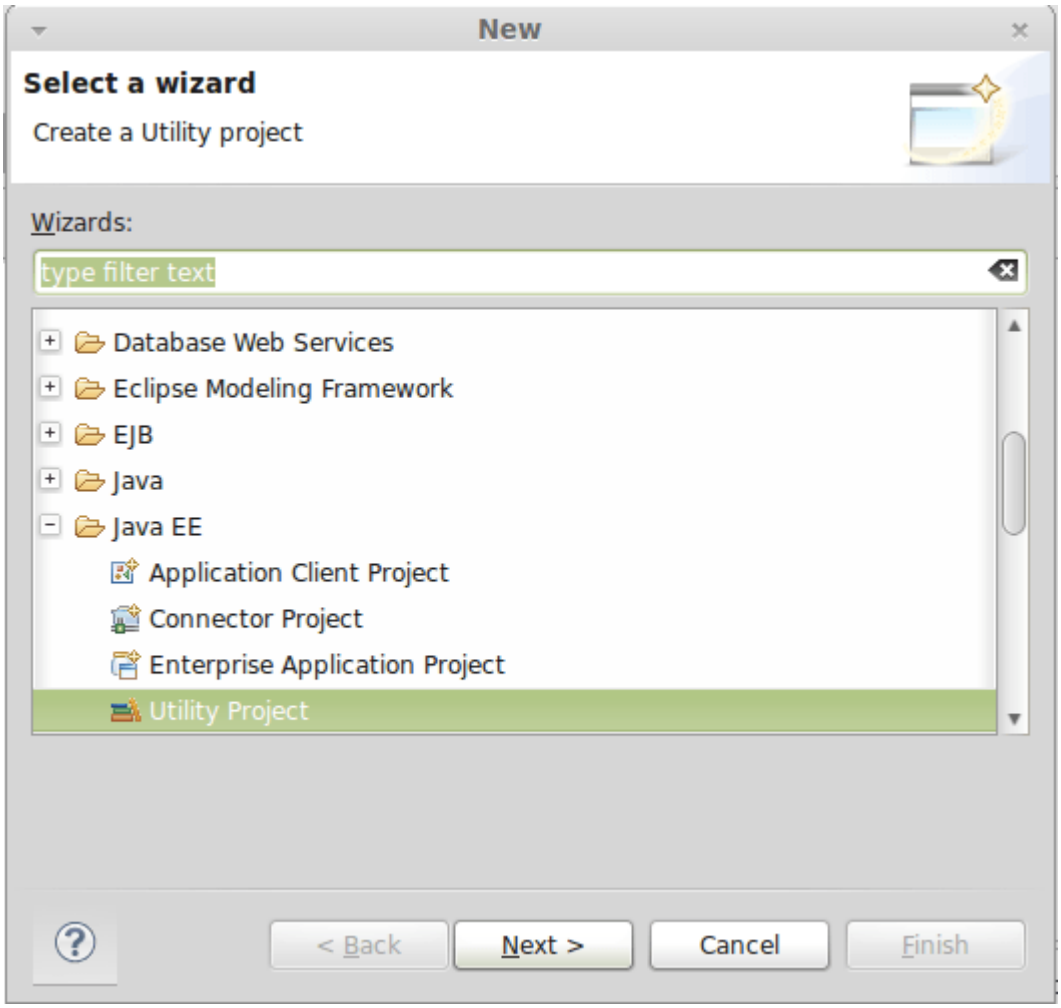


Ahora bien si tuvieras un proyecto Web02 y quisieras usar la mismas clase nos encontrariamos con una situación en la que deberemos volver a añadir la clase al nuevo proyecto.

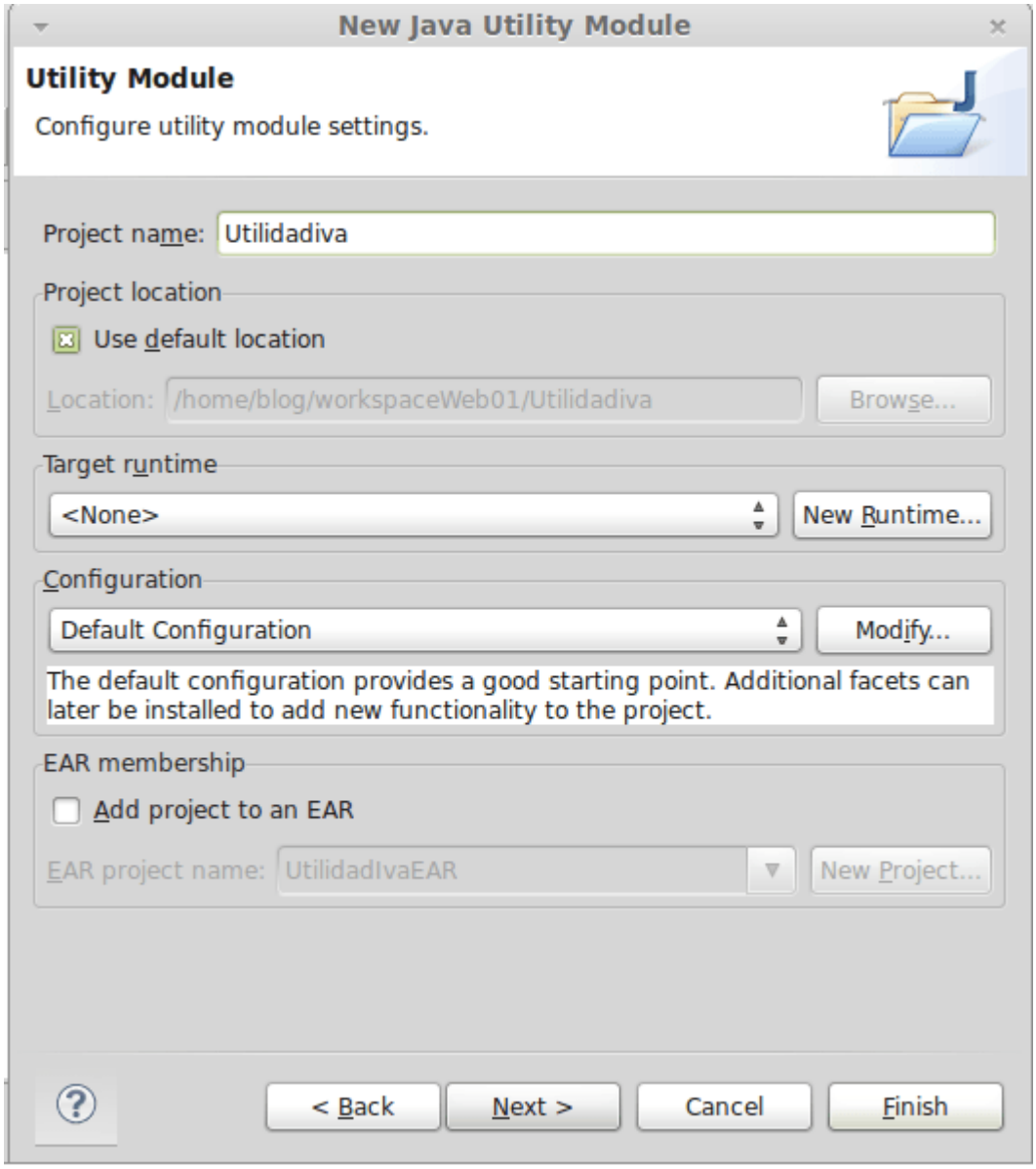


Esto es un problema ya que estamos repitiendo código en dos proyectos y si cambiamos la clase en uno habrá que actualizarla en el otro. Existen muchas soluciones a este problema . Una de las mas clásicas es construir un proyecto JAR y referenciarlo pero tiene algunos problemas . En este caso voy a usar un enfoque mas practico y voy a crear un “Utility

Proyect".El primer paso será seleccionar este proyecto a traves del asistente de eclipse.



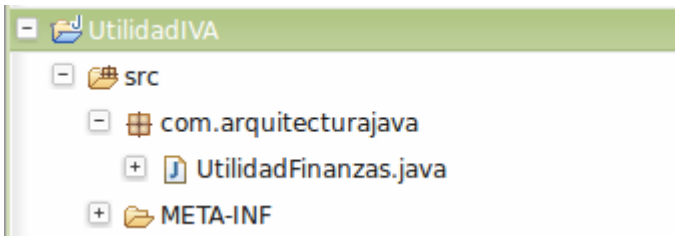
Una vez seleccionado el proyecto le asignamos un nombre.



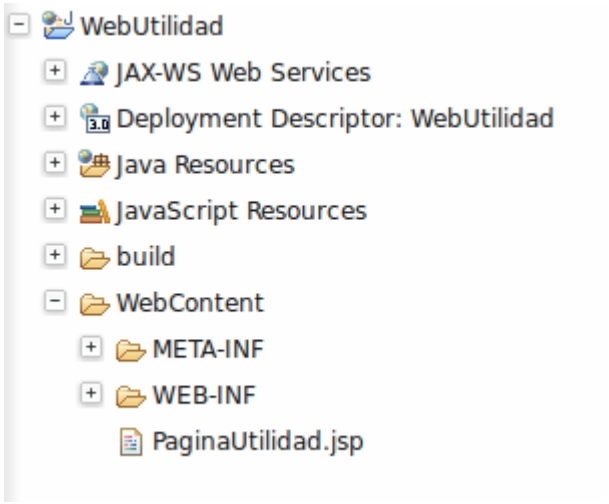
Una vez realizado este paso pulsamos sobre “Finish” y nos aparecera un nuevo proyecto en la pestaña de proyect explorer .



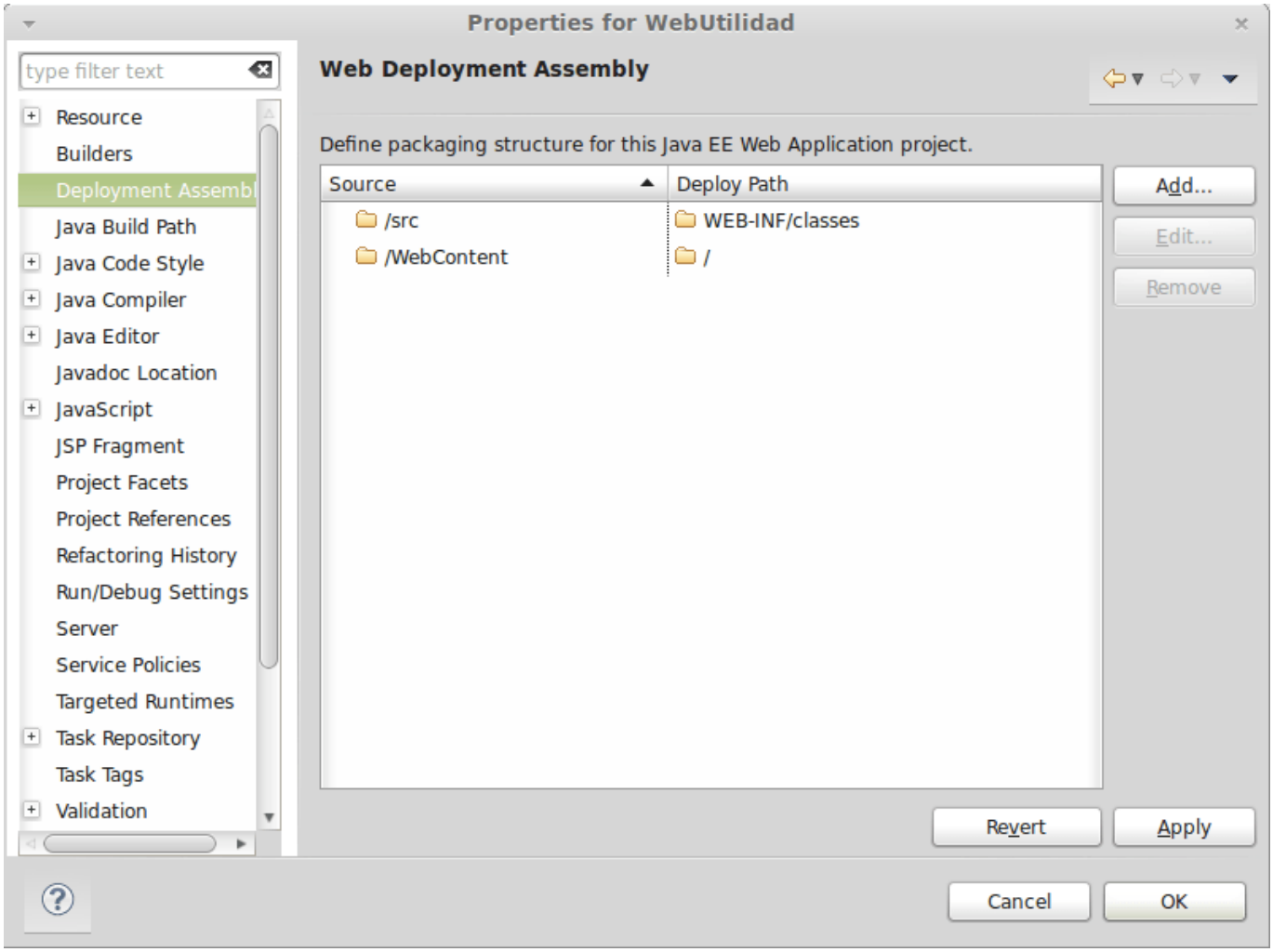
A continuación definiremos la clase UtilidadFinanzas dentro de este proyecto.



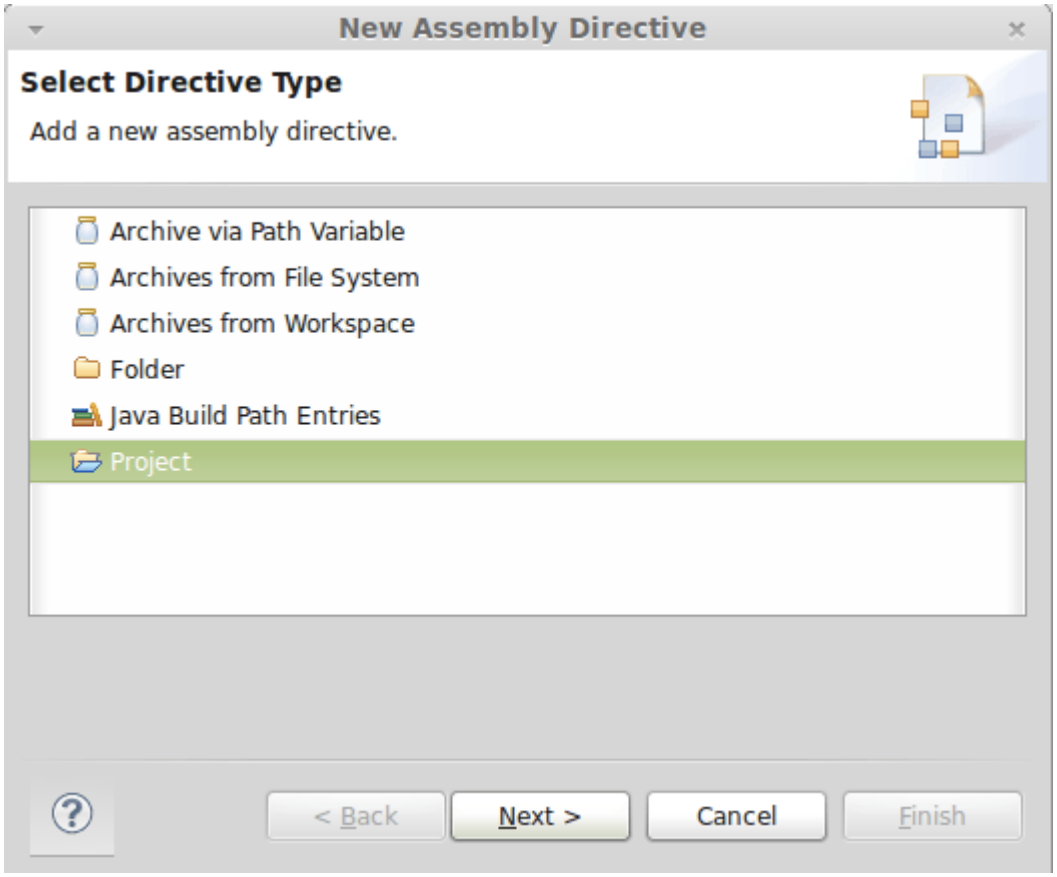
Una vez disponemos de un utility proyect vamos a construir un nuevo proyecto Web que haga uso de este proyecto de utilidades.



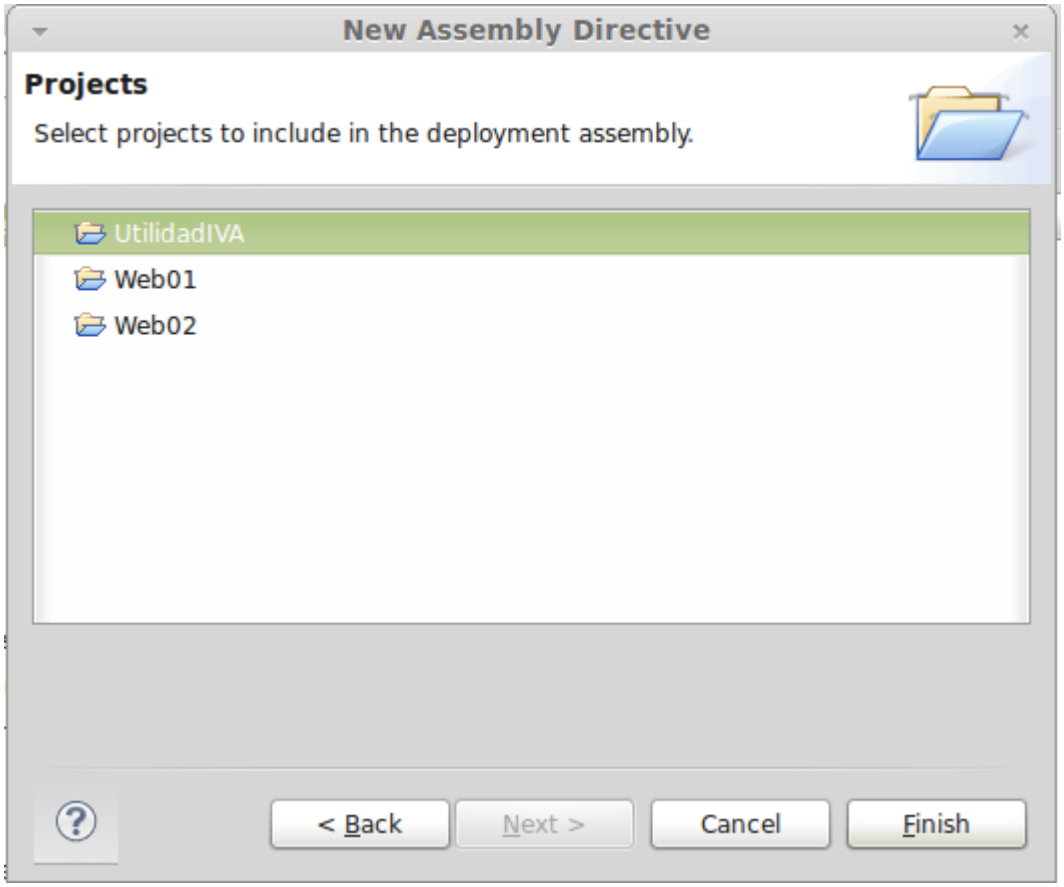
Vamos a configurar este proyecto para que se apoye en el proyecto de utilidades . Para ello pulsamos boton derecho propiedades sobre nuestro nuevo proyecto web y accedemos al siguiente menú.



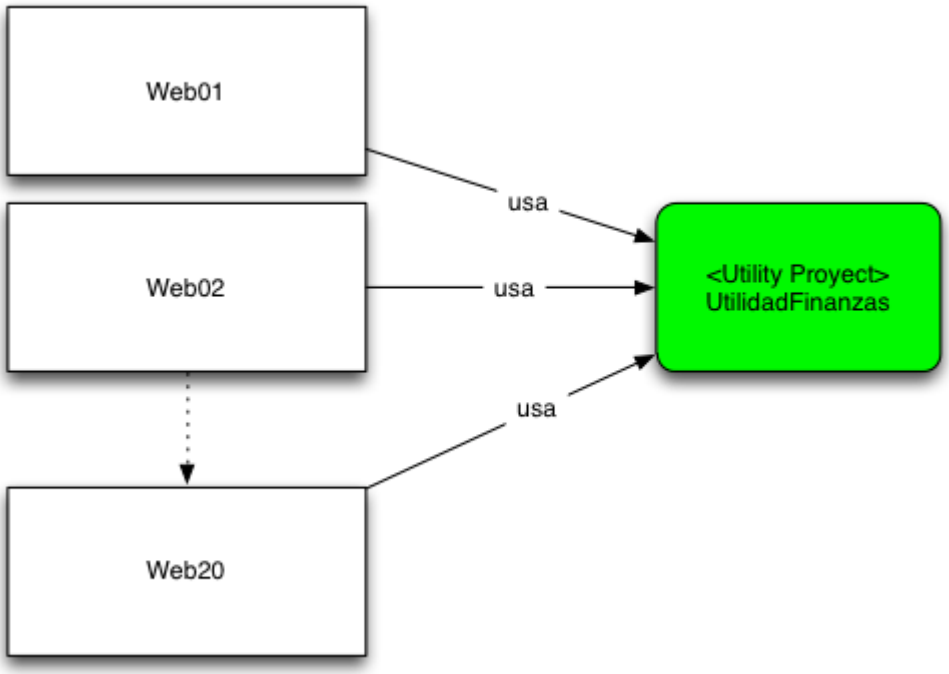
En el menú elegimos la opción de “Deployment Assembly” y pulsamos sobre “Add” una vez hecho esto nos aparecerá la siguiente ventana.



En ella seleccionamos la opción de proyect y pulsamos sobre “Next”. Una vez hecho esto elegimos nuestro proyecto de utilidades entre las elecciones que se nos presentan y pulsamos “Finish”



Realizada este operación dispondremos de un proyecto Web correctamente ligado a las librerías que utiliza. Pudiendo usar el mismo “Utility Proyect” en varios proyectos de forma simultanea.



Realizadas estas operaciones si exportamos nuestro proyecto Web como WAR podremos ver que el “Utility Project” que habiamos definido ha sido exportado como un JAR y ubicado en el directorio lib de nuestra aplicación web de forma correcta.