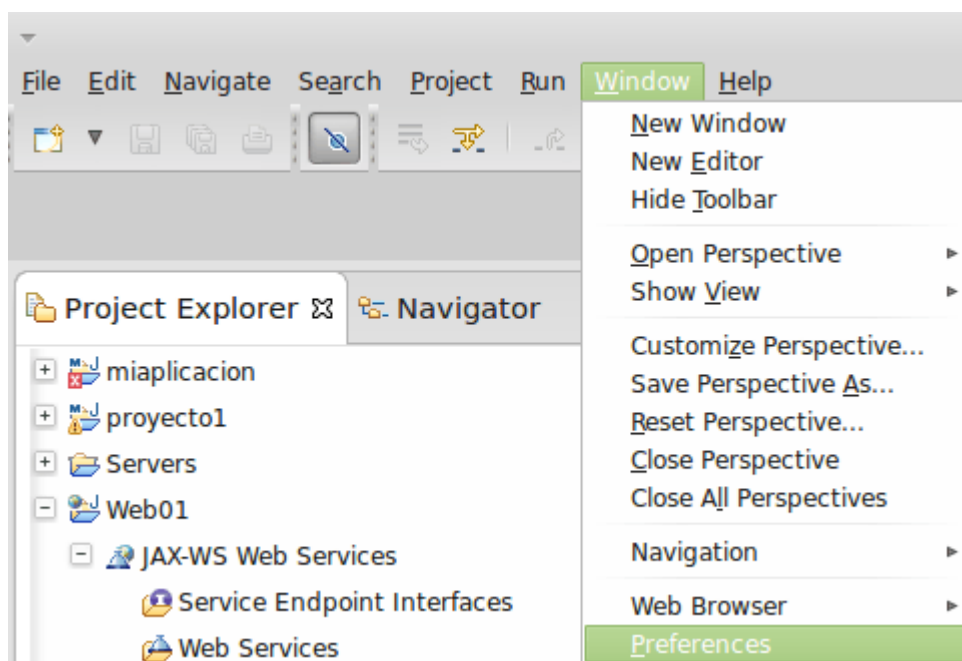
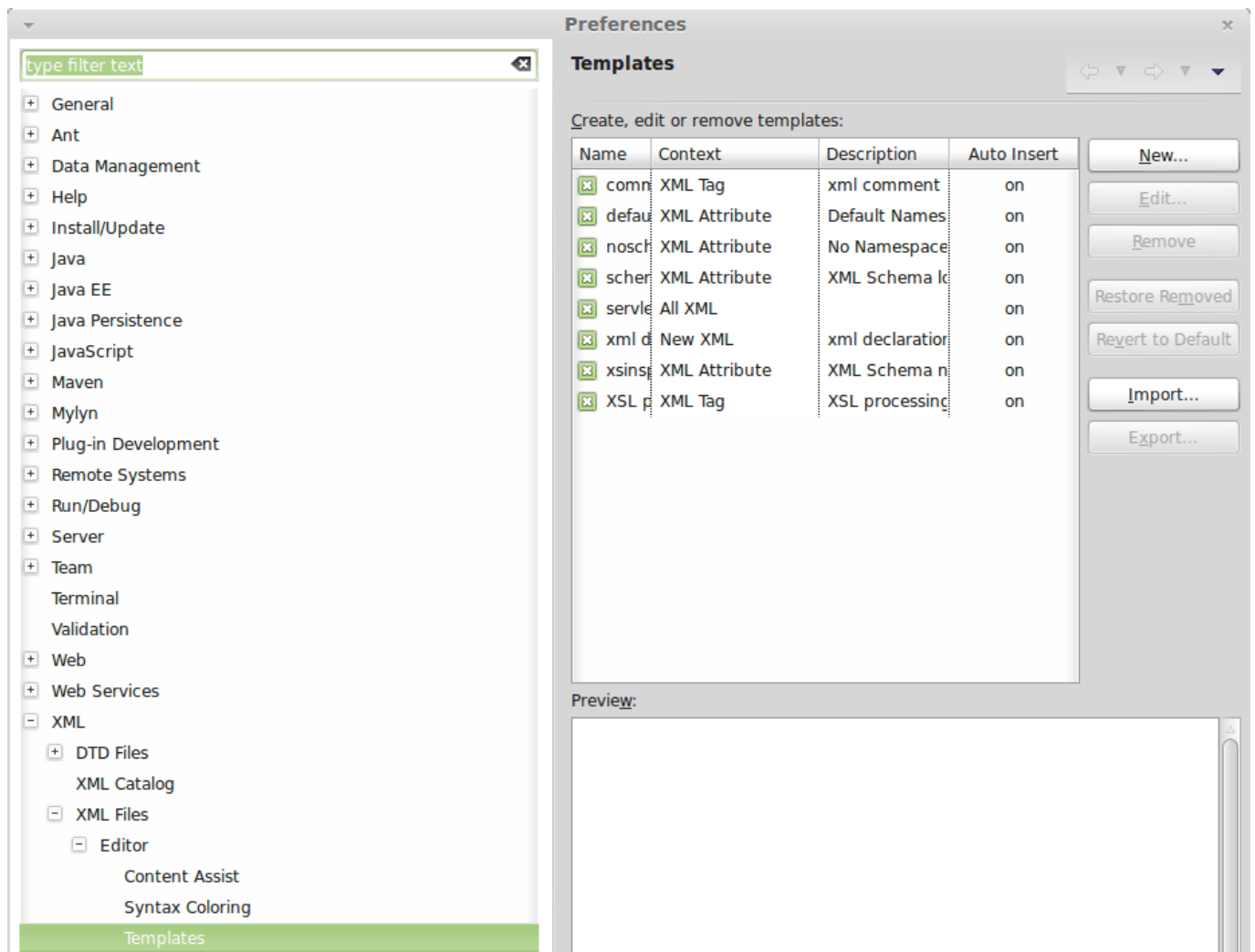


En muchas ocasiones nos encontramos programando con Eclipse y realizando tareas repetitivas. Una gran parte de estas tareas nos las resuelve el framework como por ejemplo generar constructores o metodos set/get .Sin embargo en ocasiones tenemos otras tareas que el IDE no nos ayuda como por ejemplo generar un servlet , un bean de spring etc. En estos casos podemos instalar plugins adicionales (Eclipse JEE, Spring Tools) o podemos personalizar Eclipse y definir nuestra propia plantilla que nos sirva de apoyo , esto es lo que vamos a hacer en este POST. En nuestro ejemplo vamos a crear una plantilla para definir un servlet.El primer paso será ir al menu de preferencias (Windows->Preferences).



Una vez estamos ubicados en preferencias accedemos al menu de XML->XML Files->Templates ya que vamos a construir una plantilla para ficheros XML.



Una vez realizado este paso pulsamos sobre el boton de New (nueva plantilla) para crear una plantilla nueva.

**Edit Template**

Name:  Context:  ☒ Automatically insert

Description:

Pattern: 

```
<servlet>
    <servlet-name>${nombre}</servlet-name>
    <servlet-class>com.arquitecturajava.${clase}</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>${nombre}</servlet-name>
    <url-pattern>/${nombre}</url-pattern>
</servlet-mapping>
</web-app>
```

Nos aparece un formulario en el que debemos dar nombre a la plantilla y descripción . En este caso asignamos el nombre de “servlet” y la descripción . Hecho esto definimos el propio código de la plantilla.

```
<servlet>
```

```
<servlet-name>${nombre}</servlet-name>
```

```
<servlet-class>com.arquitecturajava.${clase}</servlet-class>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
<servlet-name>${nombre}</servlet-name>
```

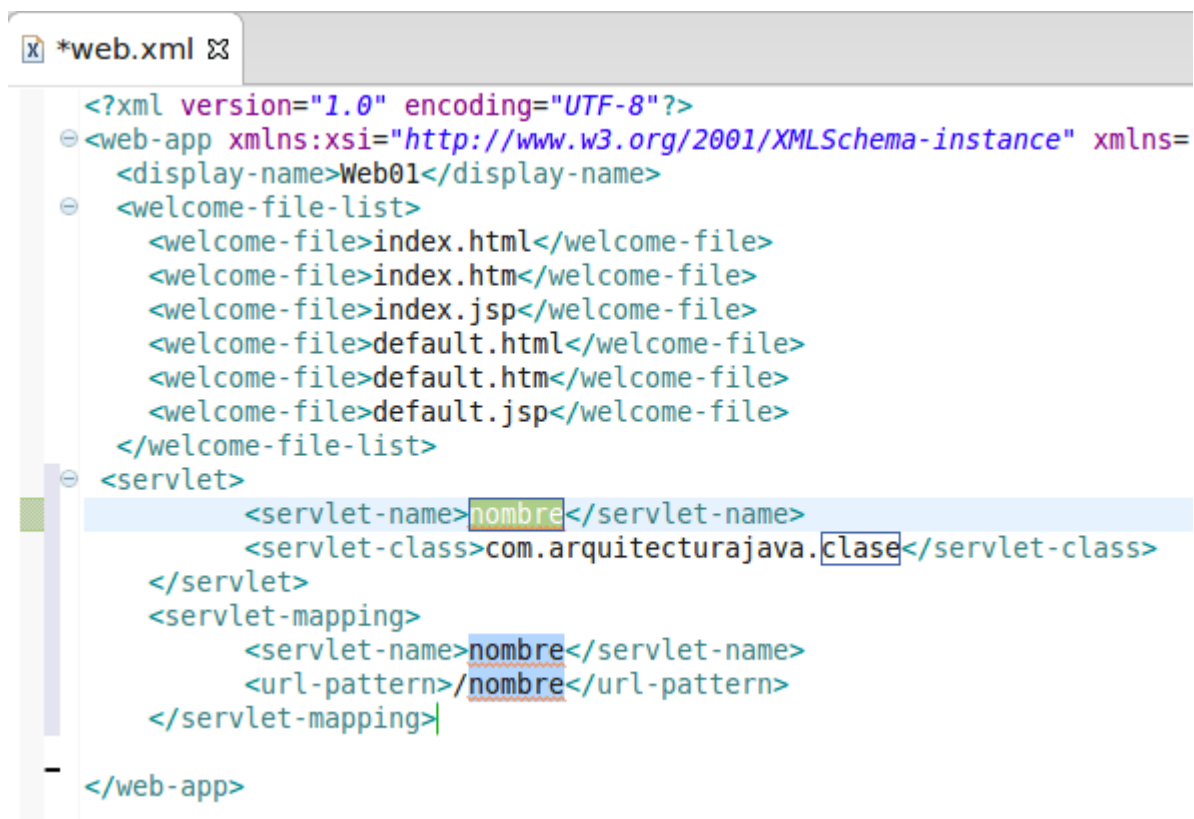
```
<url-pattern>/${nombre}</url-pattern>
```

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

Como podemos ver incluye dos variables `${nombre}` y `${clase}` las cuales podremos substituir cuando utilicemos la plantilla en nuestro código .Construida la plantilla pulsamos aceptar y la guardamos. A partir de ahora tendremos una nueva plantilla que podremos usar en nuestros ficheros xml . En nuestro caso nos interesa para el web.xml . Así pues abrimos el fichero web.xml pulsamos Ctrl+ Space y empezamos a escribir la palabra “servlet” que es como hemos denominado a nuestra plantilla.



Seleccionamos la plantilla y eclipse automaticamente nos generará el código.



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
  <display-name>Web01</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>nombre</servlet-name>
    <servlet-class>com.arquitecturajava.clase</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>nombre</servlet-name>
    <url-pattern>/nombre</url-pattern>
  </servlet-mapping>
</web-app>
```

Nada mas generar el código podremos cambiar el nombre y la clase por las que nosotros necesitemos. De esta forma las plantillas nos ahorraran mucho trabajo y nos generarán una gran flexibilidad ya que muchas veces el eclipse o los plugins que instalamos no cubren completamente nuestras necesidades.