

Programa : Java Developer
Curso : Java API for XML Web Services

Laboratorio N° 3: consumidor de Web Services Público

Duración:
60 minutos

Objetivo:

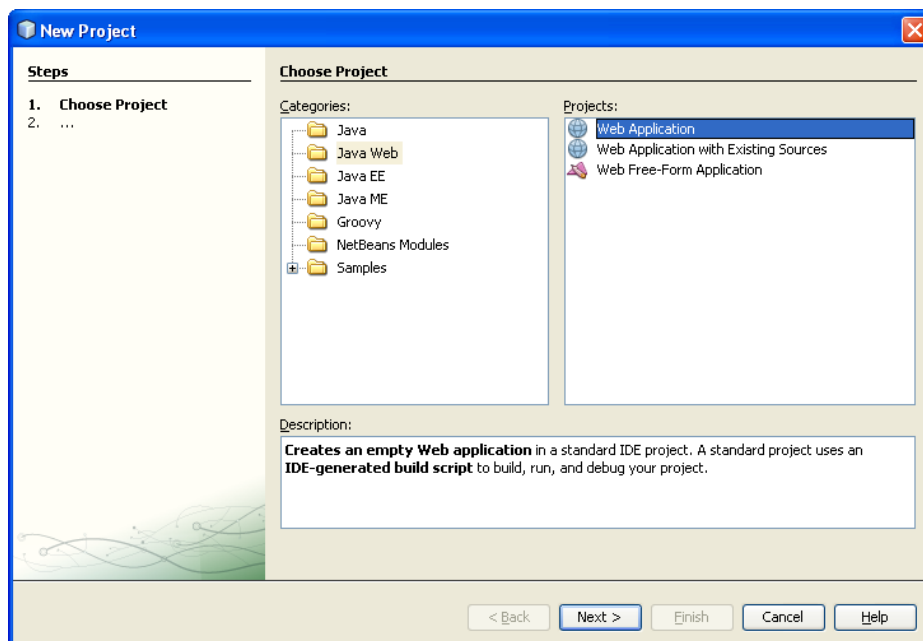
- Crear un Aplicación consumidor de Web Services Público.

Descripción:

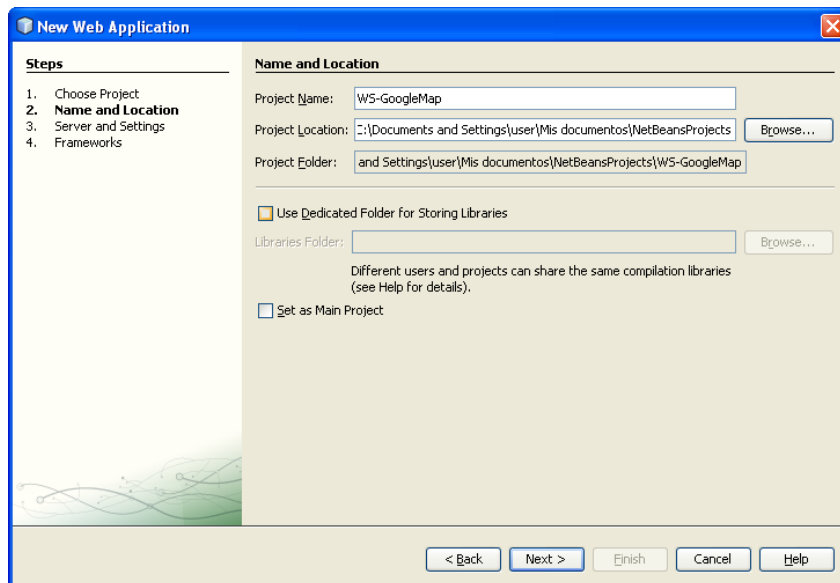
El alumno aplica información recibida en clases sobre **consumidor de Web Services Público** para desarrollar este laboratorio.

A continuación, realice los siguientes pasos.

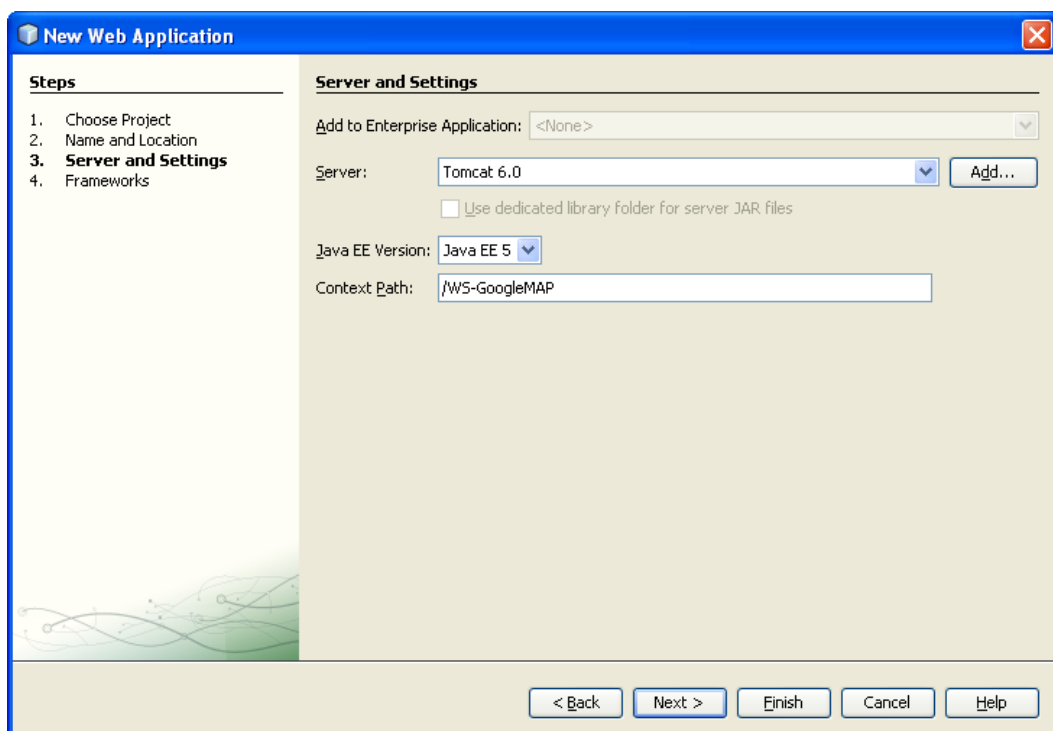
- 1º Ingrese a **File**; luego, en **New Project**, seleccione **Web Application**.



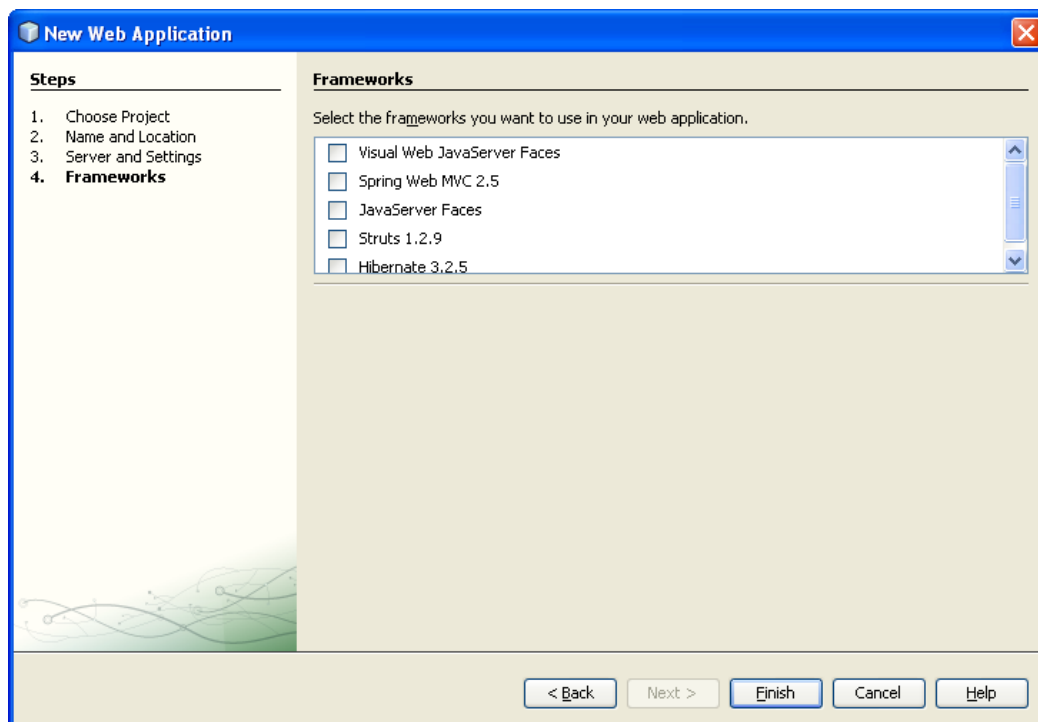
- 2º Ingrese el nombre del proyecto y la ubicación.



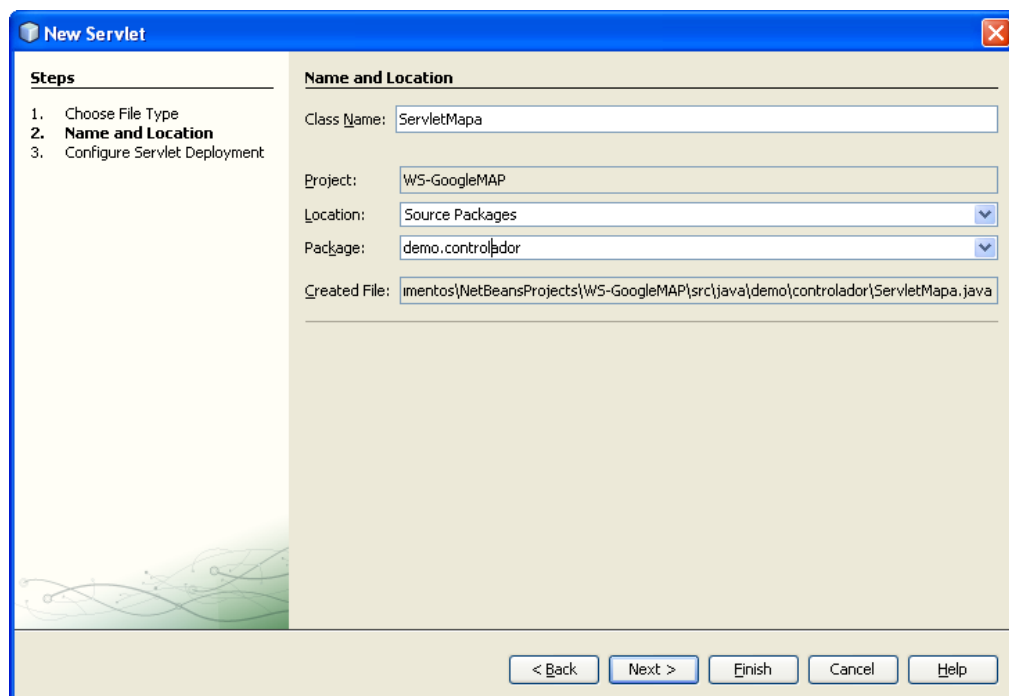
- 3º Seleccione **Tomcat** como servidor. En caso que no esté configurado, haga clic en **Add** para configurarlo. Luego, ingrese la ruta de instalación, usuario y contraseña de administración.



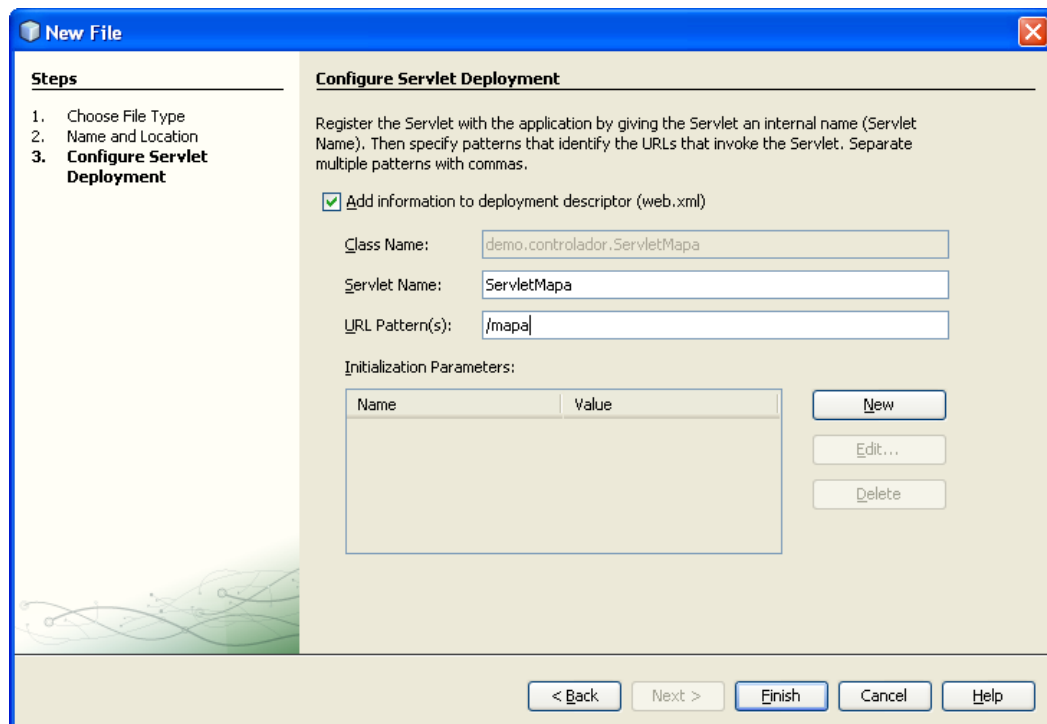
4º Recuerde que no se utilizarán frameworks en este laboratorio.



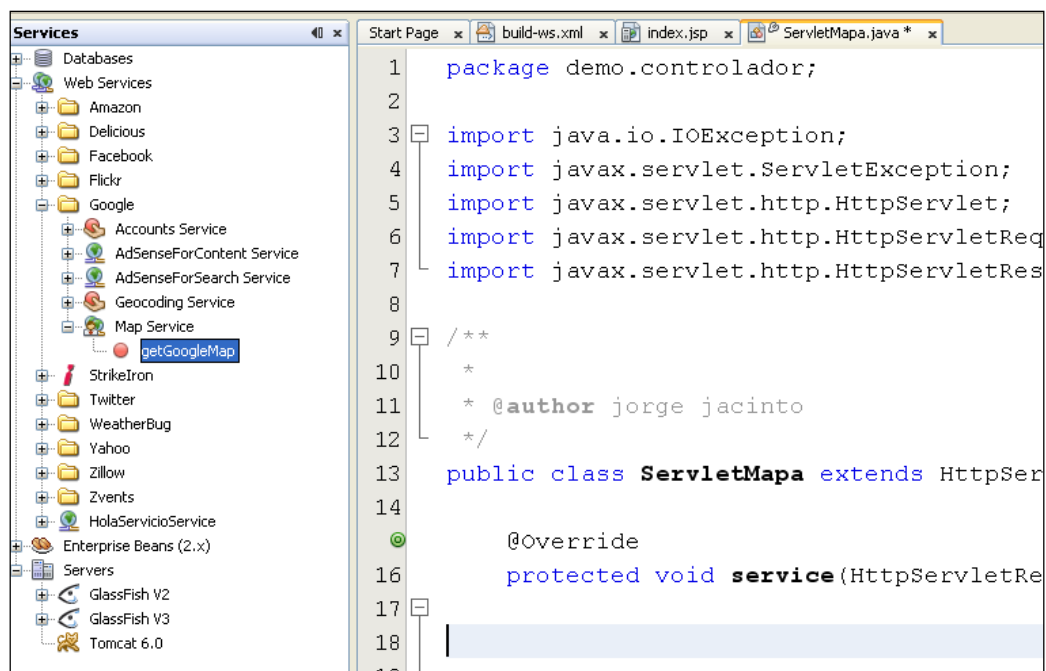
5º Cree un Servlet de nombre **ServletMapa**.



6º El alias a usar del Servlet es **mapa**.



7º Arrastre la referencia del servicio Google al Servlet creado.



8º Note que se mostrarán los valores por defecto.

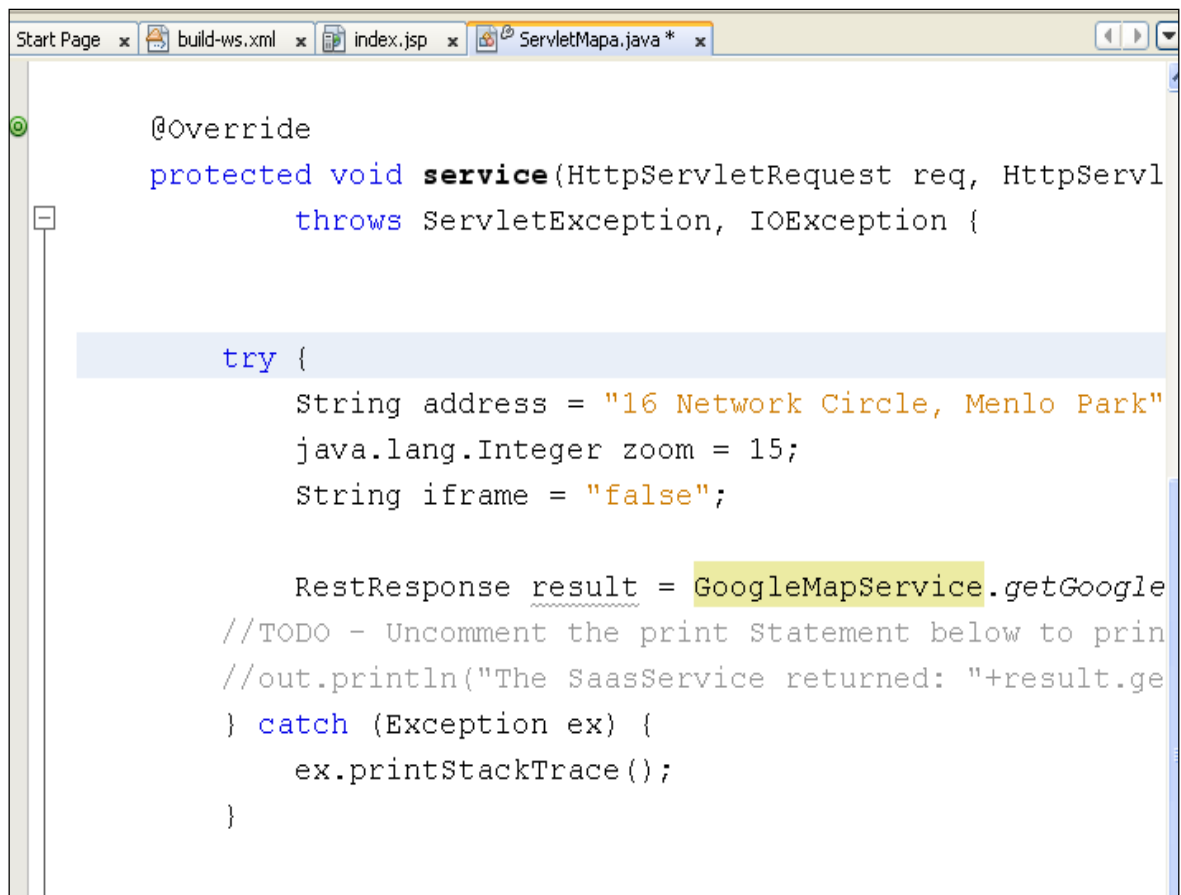
Customize getGoogleMap SaaS Service

Input Parameters:

Name	Type	Default Value
address	java.lang.String	16 Network Circle, Menlo Park
zoom	java.lang.Integer	15
iframe	java.lang.String	false

OK Cancel

9º Ahora, agregue el siguiente código.



```
@Override
protected void service(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {

    try {
        String address = "16 Network Circle, Menlo Park"
        java.lang.Integer zoom = 15;
        String iframe = "false";

        RestResponse result = GoogleMapService.getGoogle
//TODO - Uncomment the print Statement below to print
//out.println("The SaasService returned: "+result.ge
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

10º Constate que el Servlet quede de la siguiente manera.

```

package demo.controlador;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.netbeans.saas.google.GoogleMapService;
import org.netbeans.saas.RestResponse;

/**
 *
 * @author jorge jacinto
 */
public class ServletMapa extends HttpServlet {

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

        String address = req.getParameter("address");
        Integer zoom = Integer.parseInt(req.getParameter("zoom"));
        String iframe = req.getParameter("iframe");

        RestResponse result = GoogleMapService.getGoogleMap(address, zoom, iframe);

        req.setAttribute("mapaTexto", result.getDataAsString());
        req.getRequestDispatcher("/index.jsp").forward(req, resp);

    }
}

```

11º Modifique el JSP **index.jsp**, para que quede de la siguiente manera.

```

<%@page contentType="text/html" 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>JSP Page</title>
  </head>
  <body>
    <form action="mapa">
      Address : <input type="text" name="address" value=""><br>
      Zoom : <input type="text" name="zoom" value="15"><br>
      Iframe : <input type="text" name="iframe" value="false"><br>
      <input type="submit" value="buscar"><br>
    </form>
    ${requestScope.mapaTexto}
  </body>
</html>

```

- 12º Finalmente, se muestra el JSP de la búsqueda de una localidad, por medio del Web Service Google Map de Google.

