

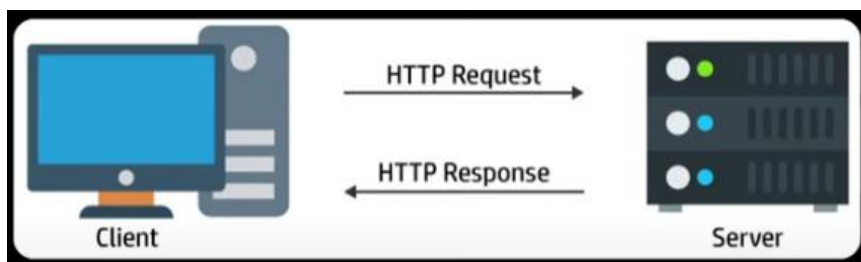
Laboratorios 6

Objetivos:

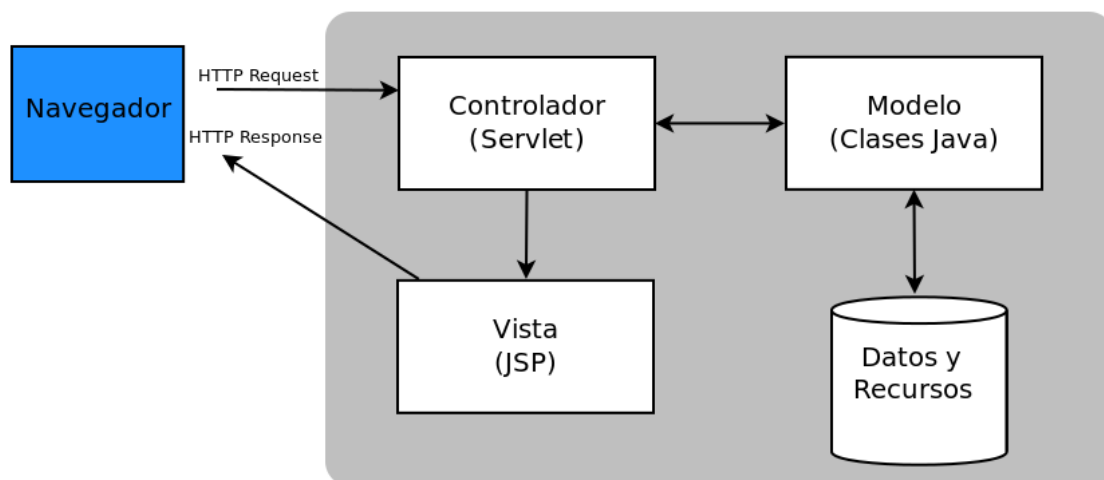
- ✓ Comprender la relación entre Servlets y JSP en la creación de aplicaciones web dinámicas.
- ✓ Aplicar los conceptos de Servlets y JSP para desarrollar aplicaciones web dinámicas que respondan a las solicitudes del cliente.

¿Qué es un Servlet?

- ✓ Básicamente es una "clase" especial de Java que se utiliza como punto intermedio entre una página JSP y el servidor web donde está alojada la lógica de una aplicación.
- ✓ Un servlet se encarga de recibir peticiones o request desde un cliente, tratarlas y analizar si es necesario realizar alguna solicitud en particular o brindar una determinada respuesta o response.



- ✓ . Para poder tratar cada una de las peticiones, utiliza una serie de métodos donde dependiendo del verbo por el cual se reciba la petición (GET, POST, PUT, DELETE, etc)





¿Métodos de un Servlet?

Los servlets tienen diferentes métodos que pueden ser utilizados dependiendo del tipo de solicitud que reciban por parte del cliente.

Los dos más usados son:

- **doGet():** Es el método encargado de recibir las solicitudes que provienen mediante GET.
- **doPost():** Es el método encargado de recibir las solicitudes que provienen mediante POST.

Actividad



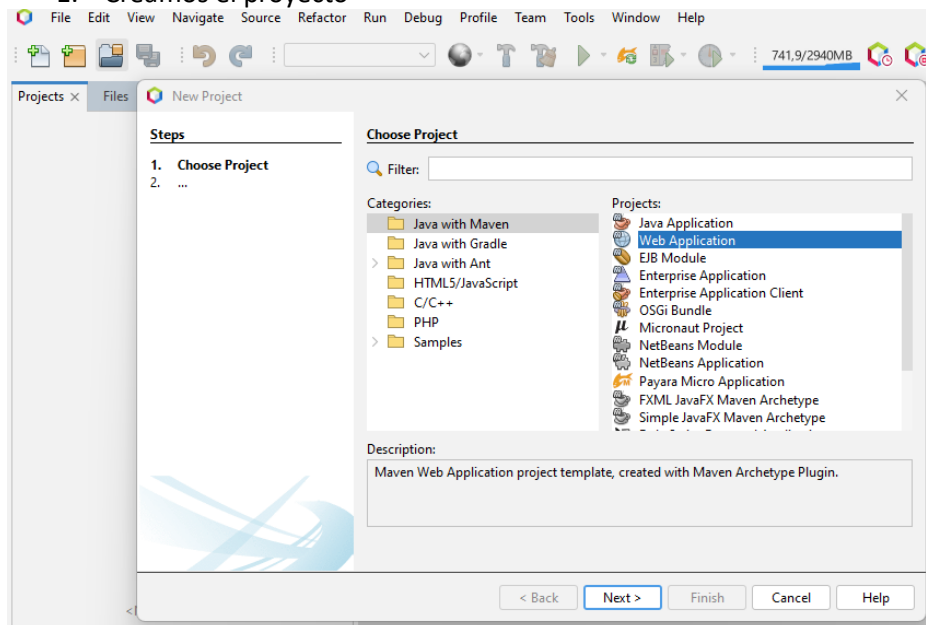
1- Crea un Proyecto y trabaja con Servlet y JSP

Crea un proyecto servlet01, con Maven y te tipo Web Application.

Group Id: py.una.pol.par

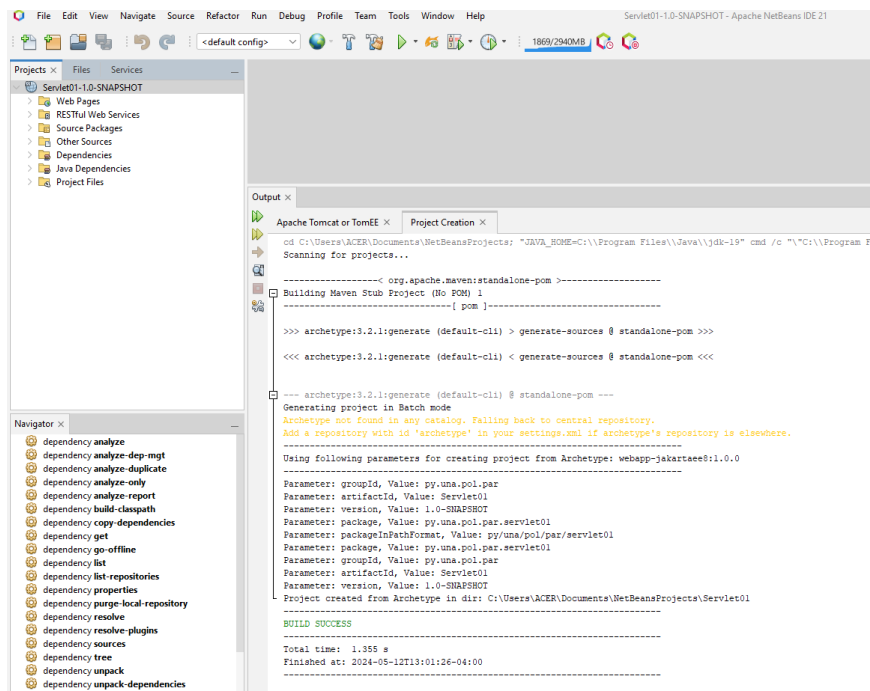
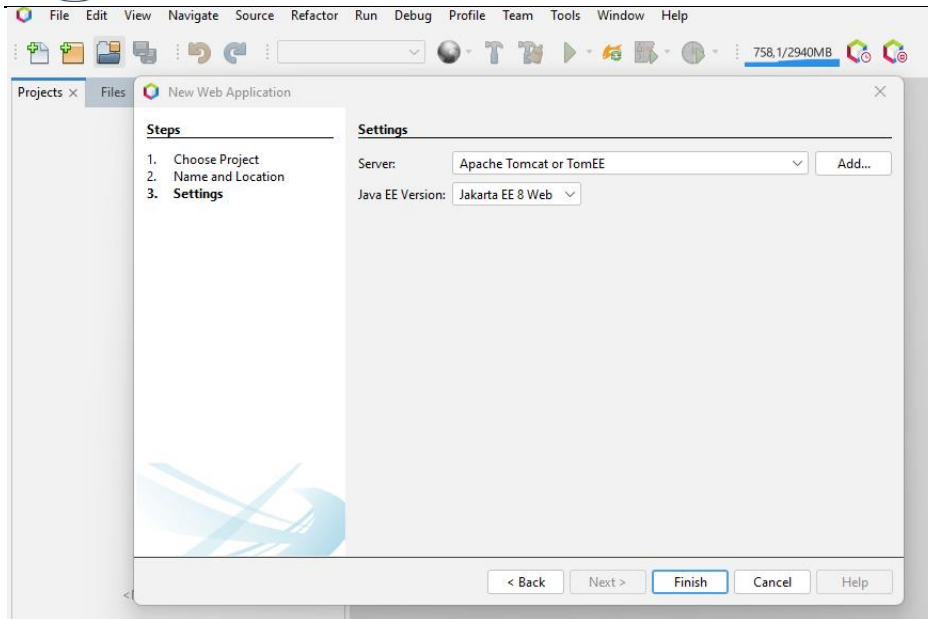
Servidor: Apache Tomcat

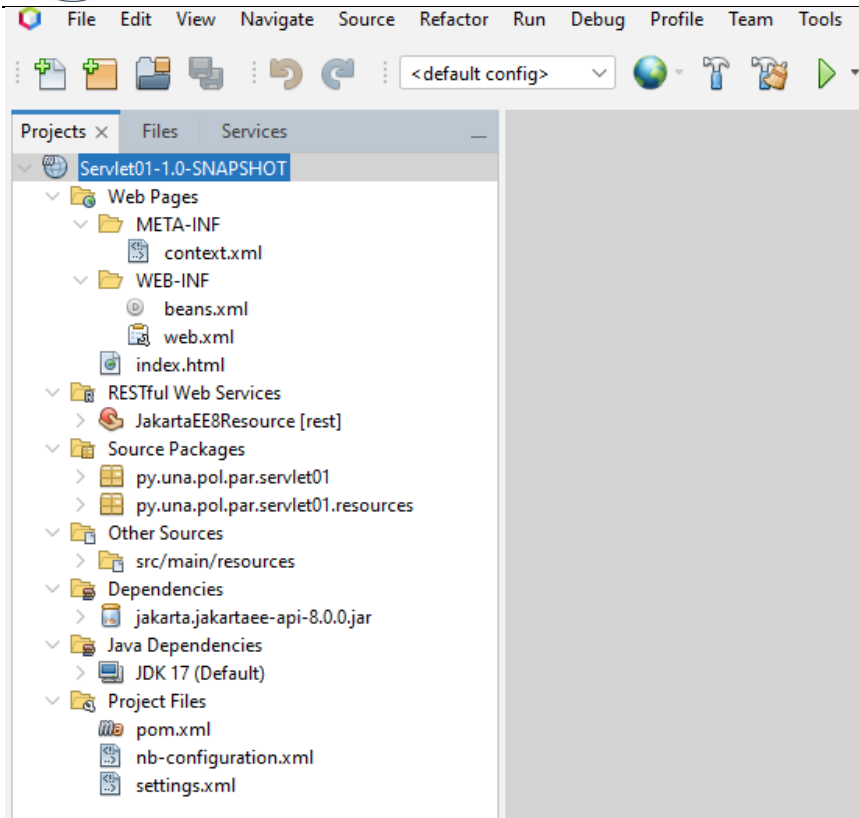
1. Creamos el proyecto



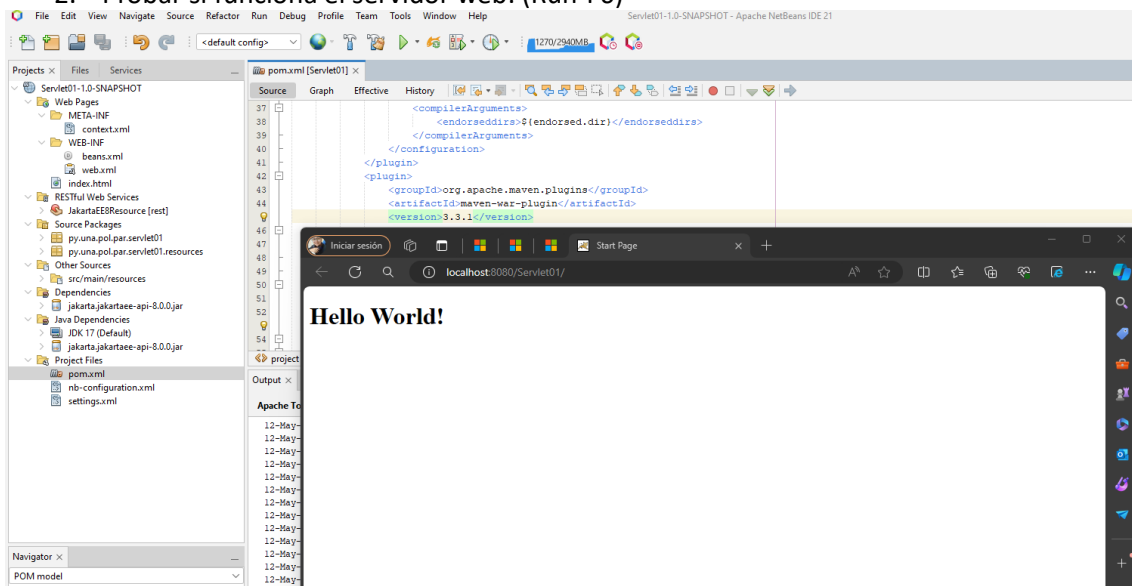


Universidad Nacional de Asunción
Facultad Politécnica
Licenciatura en Ciencias Informáticas
Programación de Aplicaciones en Redes – Prácticas de Laboratorio [LAB06]





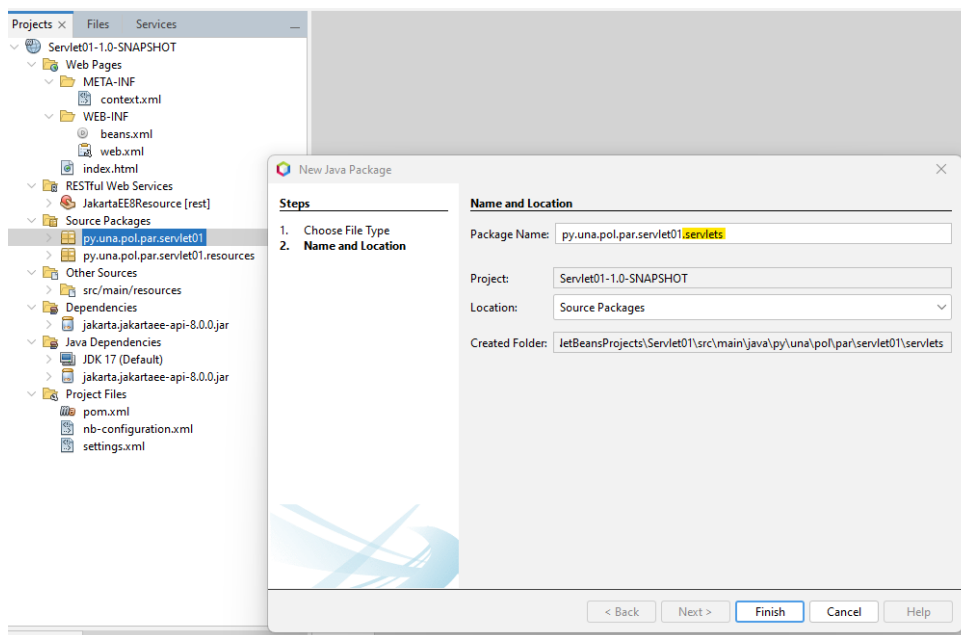
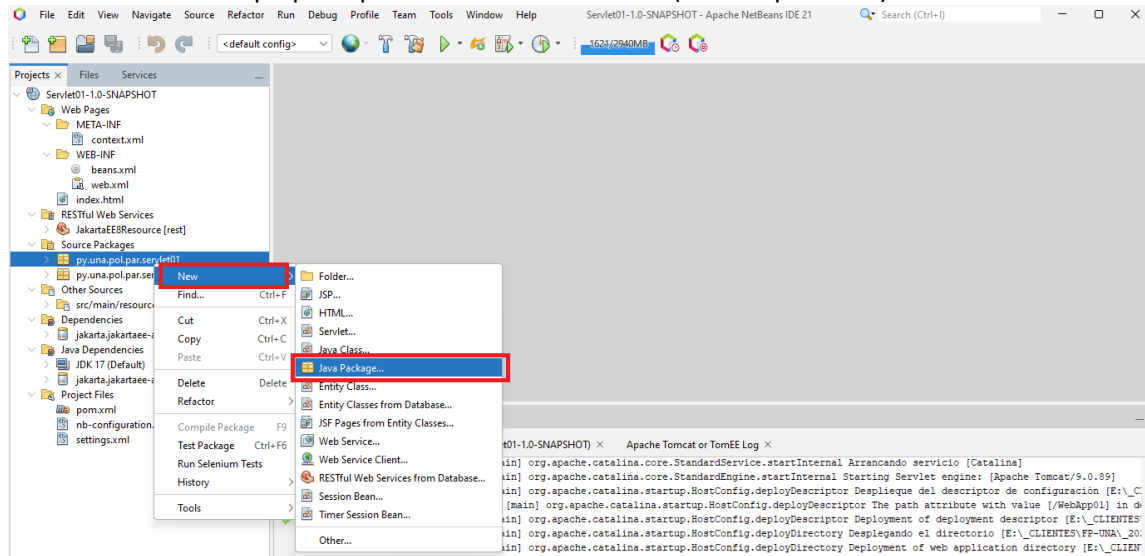
2. Probar si funciona el servidor web: (Run-F6)

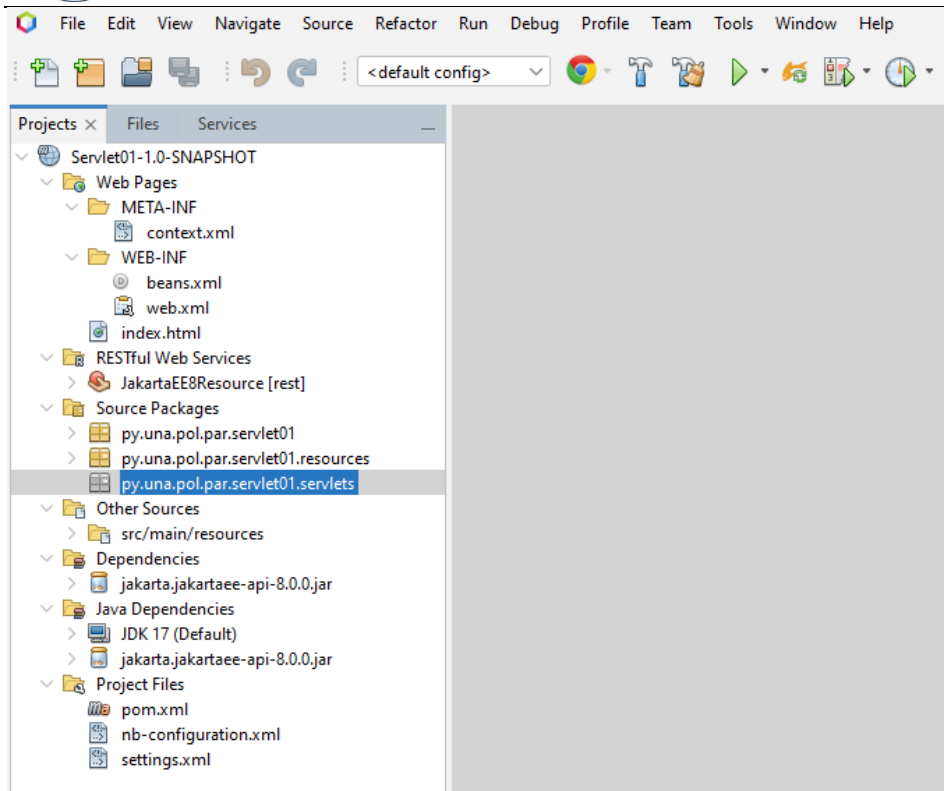


3. Creamos el primer Servlet

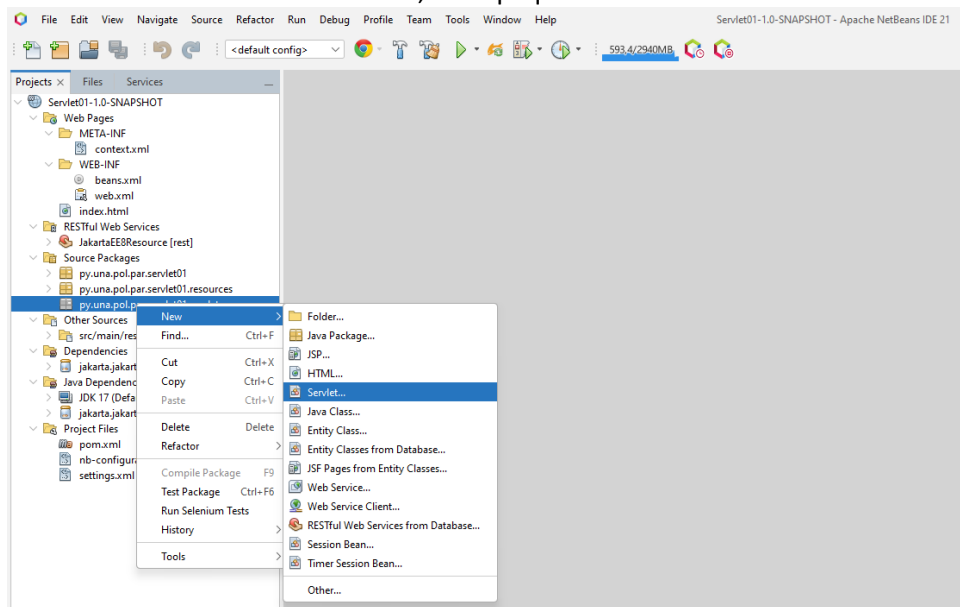


3.1. Creamos un paquete para contener los servlets (buenas prácticas)



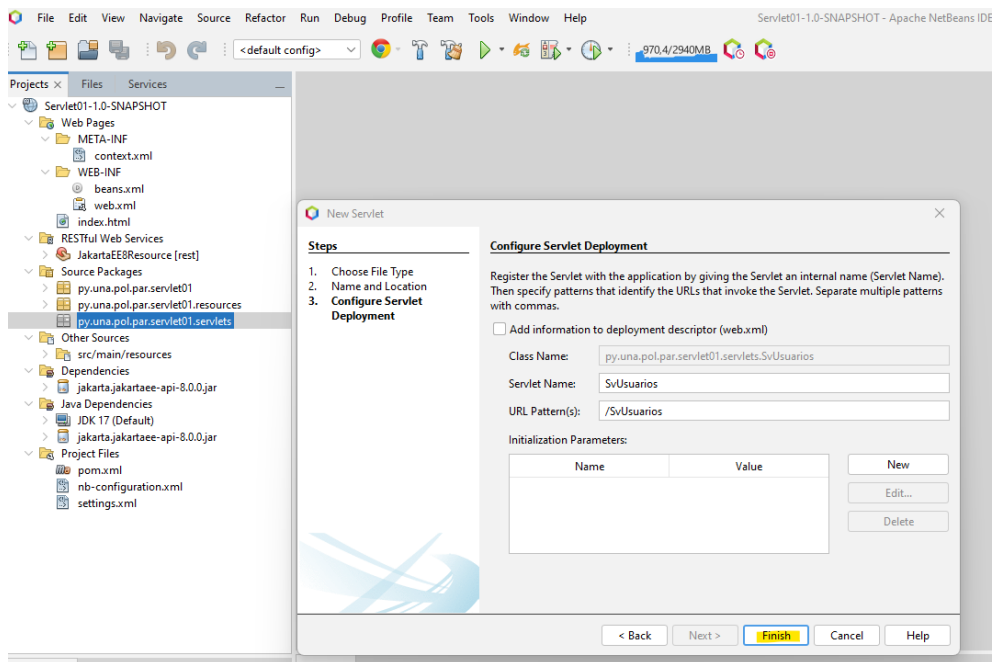
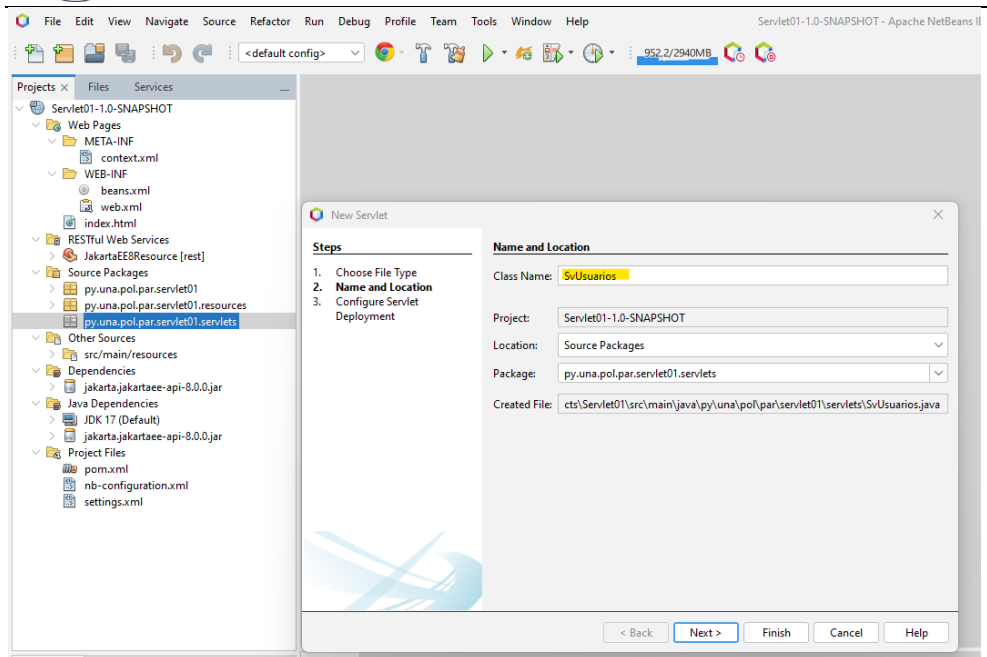


3.2. Creamos servlets como tal, en el paquete creado





Universidad Nacional de Asunción
Facultad Politécnica
Licenciatura en Ciencias Informáticas
Programación de Aplicaciones en Redes – Prácticas de Laboratorio [LAB06]





Universidad Nacional de Asunción
Facultad Politécnica
Licenciatura en Ciencias Informáticas
Programación de Aplicaciones en Redes – Prácticas de Laboratorio [LAB06]

```
1  /**
2   * Click https://nhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click https://nhost/SystemFileSystem/Templates/JSP\_Servlet/Servlet.java to edit this template
4   */
5  package py.una.pol.par.servlet01.servlets;
6
7  import java.io.IOException;
8  import java.io.PrintWriter;
9  import javax.servlet.ServletException;
10 import javax.servlet.annotation.WebServlet;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15 /**
16  *
17  * @author Opalacios
18  */
19 @WebServlet(name = "SvUsuarios", urlPatterns = {"/SvUsuarios"})
20 public class SvUsuarios extends HttpServlet {
21
22     /**
23      * Processes requests for both HTTP GET and POST
24      * methods.
25      *
26      * @param request servlet request
27      * @param response servlet response
28      * @throws ServletException if a servlet-specific error occurs
29      * @throws IOException if an I/O error occurs
30      */
31     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
32         throws ServletException, IOException {
33         response.setContentType("text/html;charset=UTF-8");
34         try (PrintWriter out = response.getWriter()) {
35             /* TODO output your page here. You may use following sample code. */
36             out.println("<doctype html>");
37             out.println("<html>");
38             out.println("<head>");
39             out.println("<title>Servlet SvUsuarios</title>");
40             out.println("</head>");
41             out.println("<body>");
42             out.println("<h1>Servlet SvUsuarios at " + request.getContextPath() + "</h1>");
43             out.println("</body>");
44             out.println("</html>");
45         }
46     }
47
48     /**
49      * @Override
50      * @param request
51      * @param response
52      * @throws ServletException
53      * @throws IOException
54      */
55     protected void doGet(HttpServletRequest request, HttpServletResponse response)
56         throws ServletException, IOException {
57         processRequest(request, response);
58     }
59
60     /**
61      * @Override
62      * @param request
63      * @param response
64      * @throws ServletException
65      * @throws IOException
66      */
67     protected void doPost(HttpServletRequest request, HttpServletResponse response)
68         throws ServletException, IOException {
69         processRequest(request, response);
70     }
71
72     /**
73      * @Override
74      * @return
75      */
76     public String getServletInfo() {
77         return "Short description";
78     }
79 }
```

```
1  import javax.servlet.http.HttpServletRequest;
2  import javax.servlet.http.HttpServletResponse;
3
4  @WebServlet(name = "SvUsuarios", urlPatterns = {"/SvUsuarios"})
5  public class SvUsuarios extends HttpServlet {
6
7      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
8          throws ServletException, IOException {
9          //Esto si se quiere construir las respuestas dentro del mismo Servlet, lo ideal es que se maneje por
10         //capas con los JSP como front para interactuar con el cliente
11         /*
12          *
13          * @param request
14          * @param response
15          * @throws ServletException
16          * @throws IOException
17          */
18         response.setContentType("text/html;charset=UTF-8");
19         try (PrintWriter out = response.getWriter()) {
20             out.println("<!DOCTYPE html>");
21             out.println("<html>");
22             out.println("<head>");
23             out.println("<title>Servlet SvUsuarios</title>");
24             out.println("</head>");
25             out.println("<body>");
26             out.println("<h1>Servlet SvUsuarios at " + request.getContextPath() + "</h1>");
27             out.println("</body>");
28             out.println("</html>");
29         }
30     }
31
32     @Override
33     protected void doGet(HttpServletRequest request, HttpServletResponse response)
34         throws ServletException, IOException {
35         processRequest(request, response);
36     }
37
38     @Override
39     protected void doPost(HttpServletRequest request, HttpServletResponse response)
40         throws ServletException, IOException {
41         processRequest(request, response);
42     }
43
44     @Override
45     public String getServletInfo() {
46         return "Short description";
47     }
48
49 }
50
```




Pasar datos a SERVLET

Método

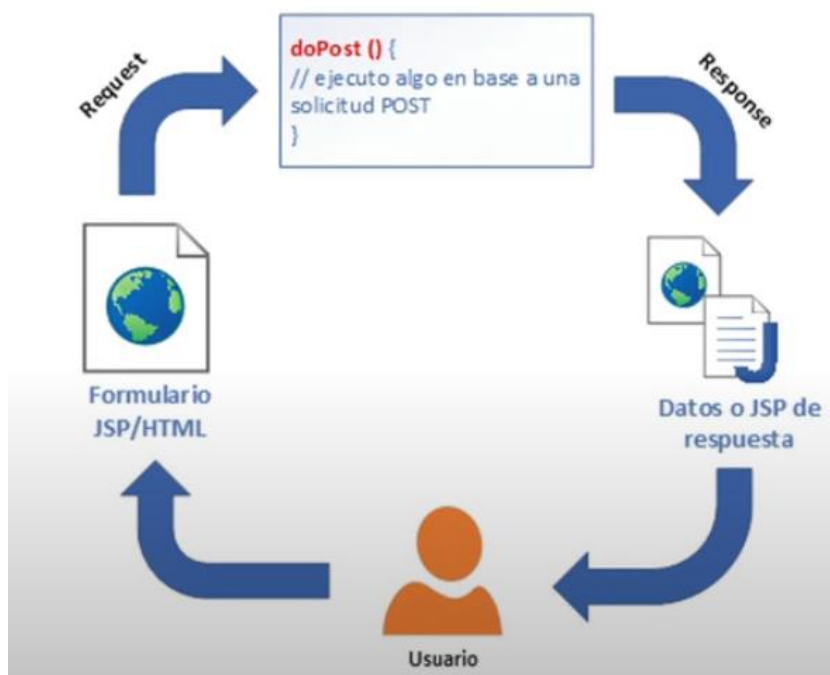
getParameter()

. Es utilizado para obtener el valor de un parametro enviado por el cliente via los metodos GET o POST.

Ejemplo de uso:

Si enviamos un dato desde un formulario en un JSP y el mismo está en una caja de texto llamada txtNombre, entonces la forma de obtener ese dato será:

String dato = request.getParameter("txtNombre");



4. Creamos index.jsp (JSP), como puerta de entrada. Nota si es necesario; elimine index.html.

index.jsp puede tener algo como:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Formulario inicial</title>
  </head>
  <body>
    <h1>Datos del Usuario</h1>
    <form action="SvUsuarios" method="POST">
      <p><label>CI:</label><input type="text" name="ci"></p>
      <p><label>Nombre:</label><input type="text" name="nombre"></p>
      <p><label>Apellido:</label><input type="text" name="apellido"></p>
      <p><label>Telefono:</label><input type="text" name="telefono"></p>
      <button type="submit"> Enviar </button>
    </form>

    <h1>Ver Usuario</h1>
    <form action="SvUsuarios" method="GET">
      <p>Para ver los datos de los usuarios cargados haga click en siguiente
      boton</p>
```



```
<button type="submit"> Mostrar usuarios </button>
</form>

</body>
</html>
```

Recuerda:

<form action="SvUsuarios" method="POST">

Esta línea de código indica que cuando se envíe el formulario, los datos se enviarán al servlet llamado "SvUsuarios" utilizando el método POST de HTTP. El servlet procesará entonces los datos del formulario y realizará las acciones necesarias en el servidor.

<form>: se utiliza para crear un formulario en una página web. Los formularios son utilizados para recopilar datos de los usuarios, como nombres, contraseñas, direcciones, etc.

action="SvUsuarios": Este atributo especifica la URL del servidor al que se enviarán los datos del formulario cuando se envíe. **En este caso, los datos del formulario se enviarán al servlet llamado "SvUsuarios"**.

method="POST": Este atributo especifica el método HTTP que se utilizará para enviar los datos del formulario al servlet.

5. Editemos el SvUsuarios.java

Puede quedar como:

```
package py.una.pol.par.servlet01.servlets;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
//import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet(name = "SvUsuarios", urlPatterns = {"/SvUsuarios"})
public class SvUsuarios extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String ci= request.getParameter("ci");
        String nombre= request.getParameter("nombre");
        String apellido= request.getParameter("apellido");
        String telefono= request.getParameter("telefono");

        //Esto para validar que estamos recibiendo, para probar paso a paso el proceso
        System.out.println("CI es: " + ci);
        System.out.println("Nombre es: " + nombre);
        System.out.println("Apellido es: " + apellido);
        System.out.println("Teléfono es: " + telefono);
    }

    @Override
```



```
public String getServletInfo() {  
    return "Short description";  
} // </editor-fold>  
  
}
```

6. Prueba la funcionalidad

7. Edita index.jsp de forma tal que despliegue una interfaz sencilla, algo como:

Datos del Usuario

CI:

Nombre:

Apellido:

Telefono:

Ver Usuario

Para ver los datos de los usuarios cargados haga click en siguiente boton

Que esto invoque al servlet SvUsuarios con el método GET.

SvUsuarios debe procesar esta petición y desplegar una lista de usuarios.

Entregas:

Lab06 – Entrega por:

1. **Plataforma EDUCA:** Documento basado en plantilla de informe de laboratorio (evidencias), en formato pdf.

2- **Repositorio:**

Dentro del repositorio creado en GitHub, con tu usuario y la descripción: **par2024**

Dentro de **laboratorios/Lab06**

Levantar la implementación realizada.

Permitir el acceso a: cnpalacios@pol.una.py.

Los códigos que no pueden ser probado, no serán considerados.