



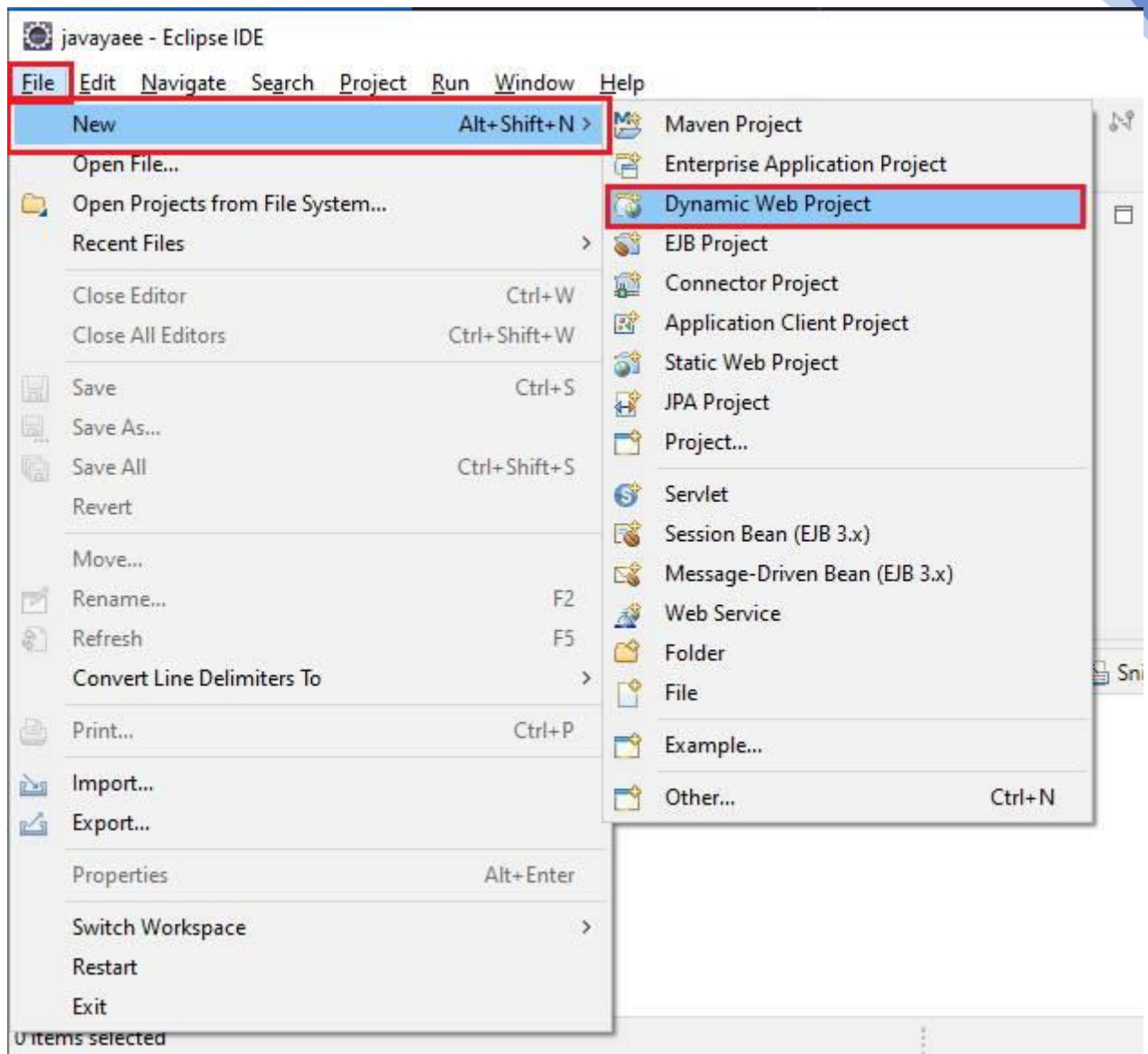
Servlet

Un servlet es una clase que se ejecuta en el contexto de un servidor web (en nuestro caso el Apache Tomcat)

Un servlet se ejecuta en un servidor web y el resultado de ejecución viaja por internet para ser visualizado en un navegador web (normalmente un servlet genera HTML, pero puede generar otros formatos de archivos)

Veremos los pasos en Eclipse para crear un servlet mínimo que nos muestre un mensaje y los números del 1 al 10000.

Desde el menú de opciones seleccionamos File -> New -> Dynamic Web Project:



En el diálogo siguiente especificamos el nombre del proyecto (en nuestro caso le llamaremos proyectoservlet1) y presionamos el botón "Finish":

New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: **proyectoservlet1**

Project location

☒ Use default location

Location: C:\javayaee\proyectoservlet1 Browse...

Target runtime

Apache Tomcat v8.0 New Runtime...

Dynamic web module version

3.1

Configuration

Default Configuration for Apache Tomcat v8.0 Modify...

A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR New Project...

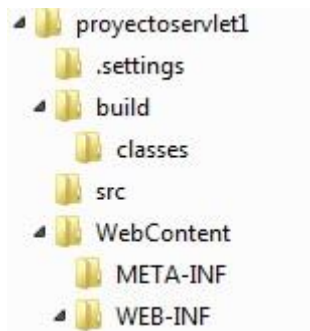
Working sets

☐ Add project to working sets New...

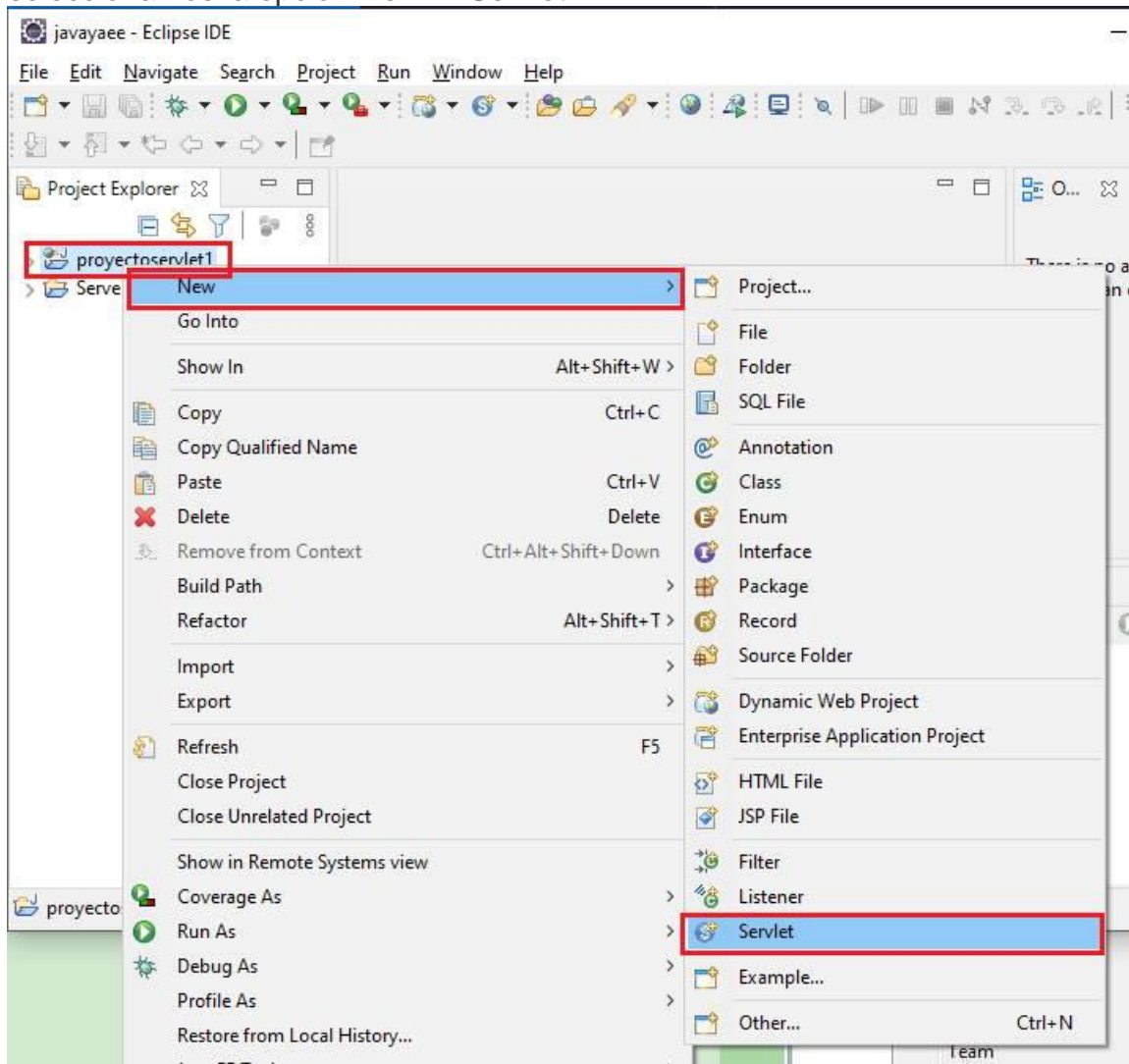
Working sets: Select...

? < Back Next > **Finish** Cancel

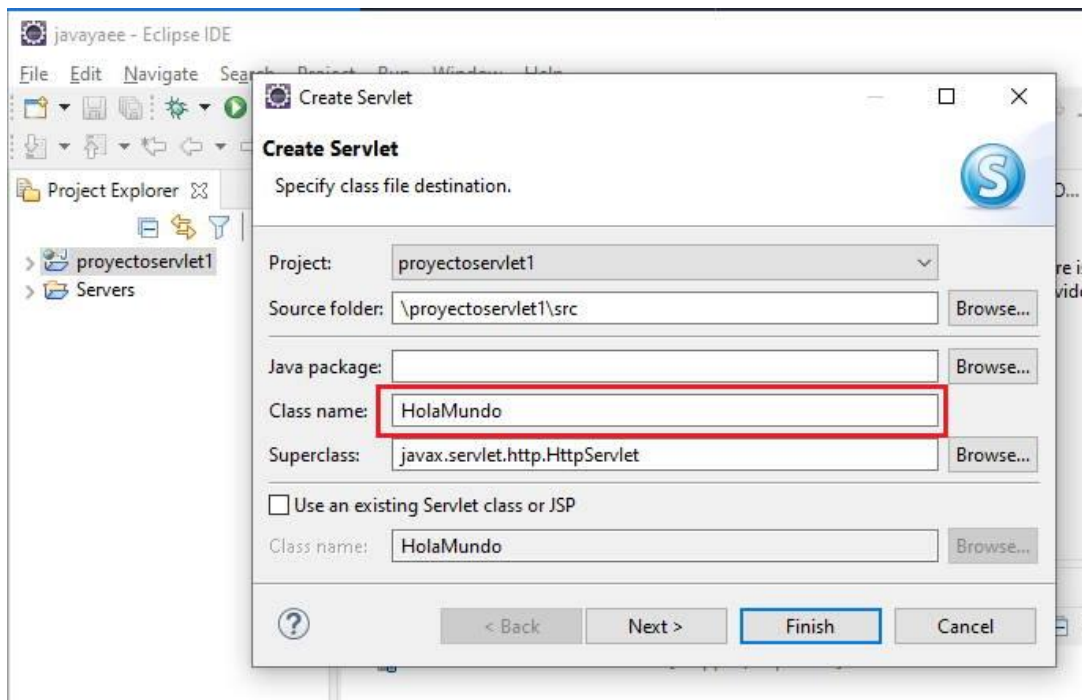
El Eclipse nos crea una serie de carpetas y archivos donde alojaremos los servlet:



Ahora presionamos el botón derecho sobre el nombre del proyecto y seleccionamos la opción New -> Servlet:



En el diálogo siguiente especificamos el nombre de nuestro servlet (en nuestro ejemplo le llamaremos HolaMundo), presionamos el botón "Finish" y ya tenemos el esqueleto básico de un servlet:



El código fuente generado es el siguiente:

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HolaMundo
 */
@WebServlet("/HolaMundo")
public class HolaMundo extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HolaMundo() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}
```

Todo servlet debe heredar de la clase HttpServlet que se encuentra en el paquete javax.servlet.http

Esta clase debe sobrescribir el método doGet o doPost (o ambos) En el protocolo HTTP las peticiones pueden ser de tipo post (cuando llamamos a una página desde un formulario HTML) y de tipo get (páginas sin formulario)

Nuestro problema es mostrar un mensaje e imprimir los números del 1 al 10000, esta actividad la haremos en el método doGet.
El algoritmo a implementar en el método doGet para dicha salida es:

```
import java.io.IOException;
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;

/**
 * Servlet implementation class HolaMundo
 */
@WebServlet("/HolaMundo")
public class HolaMundo extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HolaMundo() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head></head>");
        out.println("<body>");
        out.println("<h1>Hola Mundo</h1>");
        for(int f=1;f<=10000;f++) {
            out.println(f);
            out.println(" - ");
        }
        out.println("</body>");
        out.println("</html>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}
```

Una parte importante de la declaración del servlet que nos genera automáticamente el Eclipse es la anotación @WebServlet (esta línea registra el servlet para todas las peticiones al servidor con la sintaxis <http://localhost:8080/proyectoservlet1/HolaMundo>):

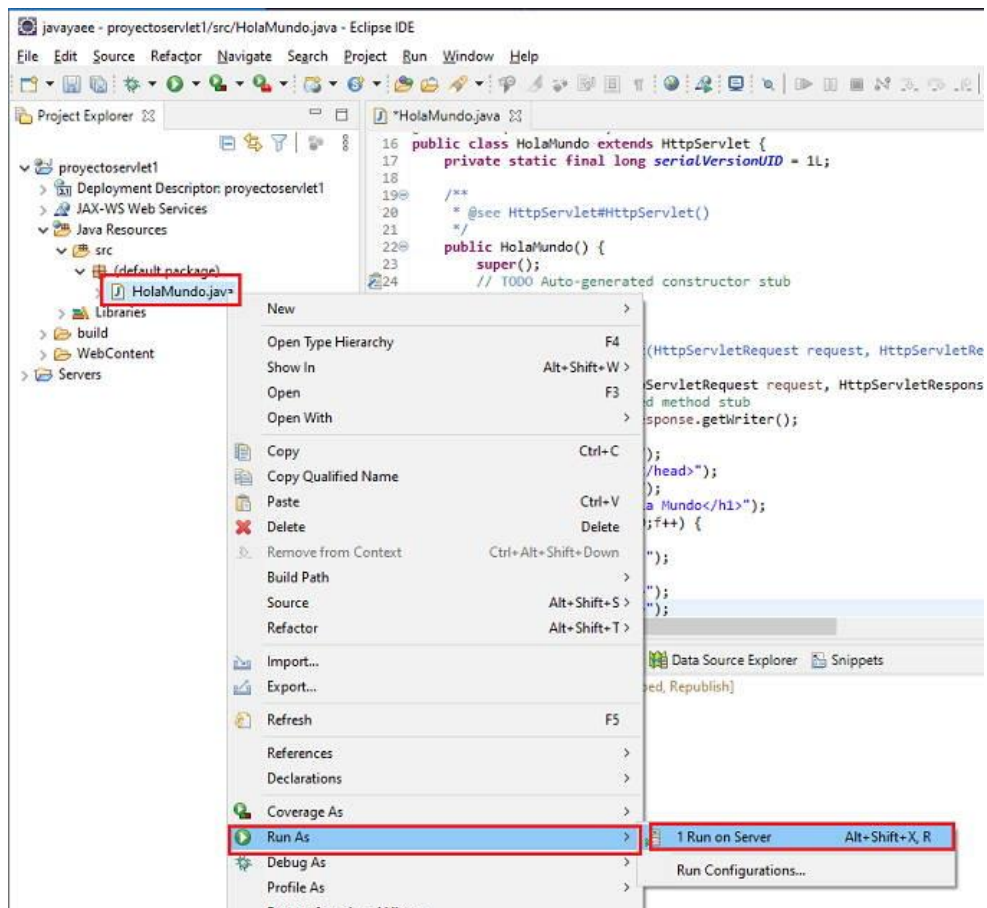
@WebServlet("/HolaMundo")

Obtenemos una referencia de un objeto de la clase PrintWriter (debemos importar la clase PrintWriter) mediante la llamada al método getWriter del objeto response que llega como parámetro al método doGet:

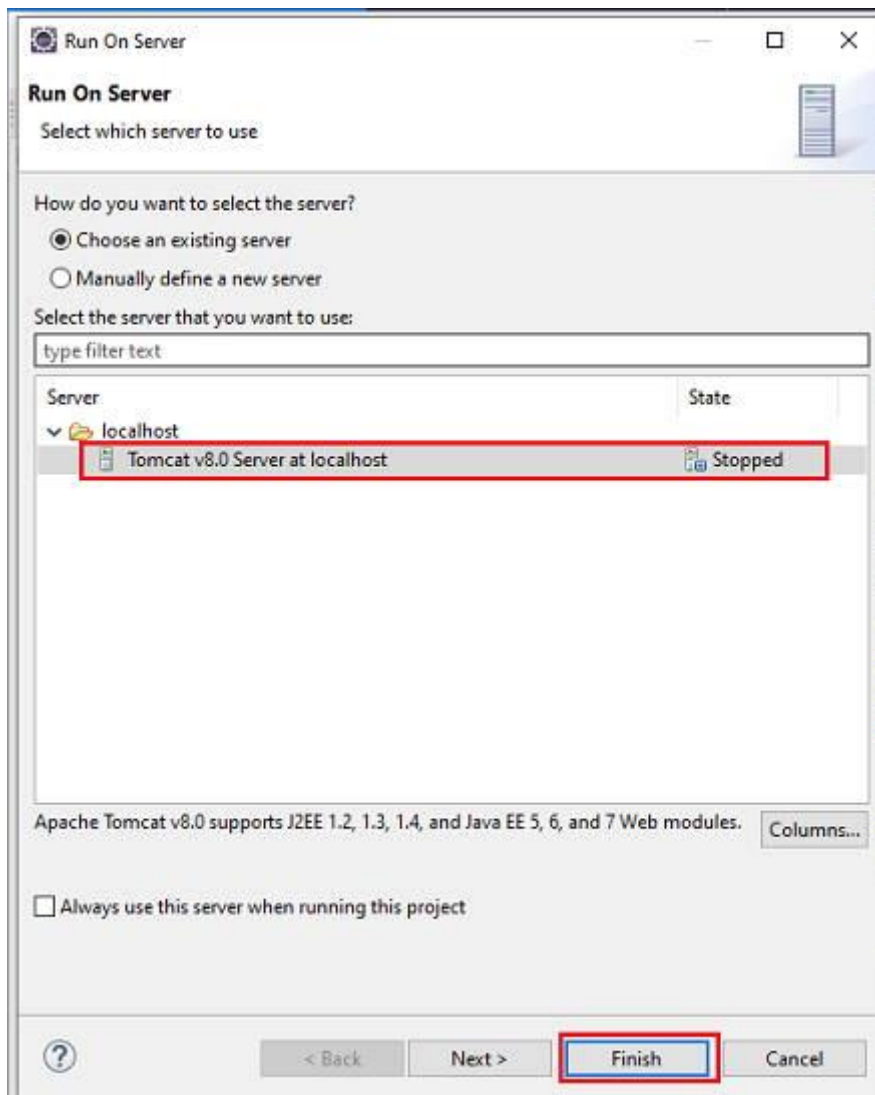
PrintWriter out = response.getWriter();

Todas las salidas son llamando al método println del objeto out de la clase PrintWriter. Como vemos generamos como salida HTML, para mostrar los números del 1 al 10000 es más conveniente utilizar una estructura repetitiva que hacer una salida secuencial.

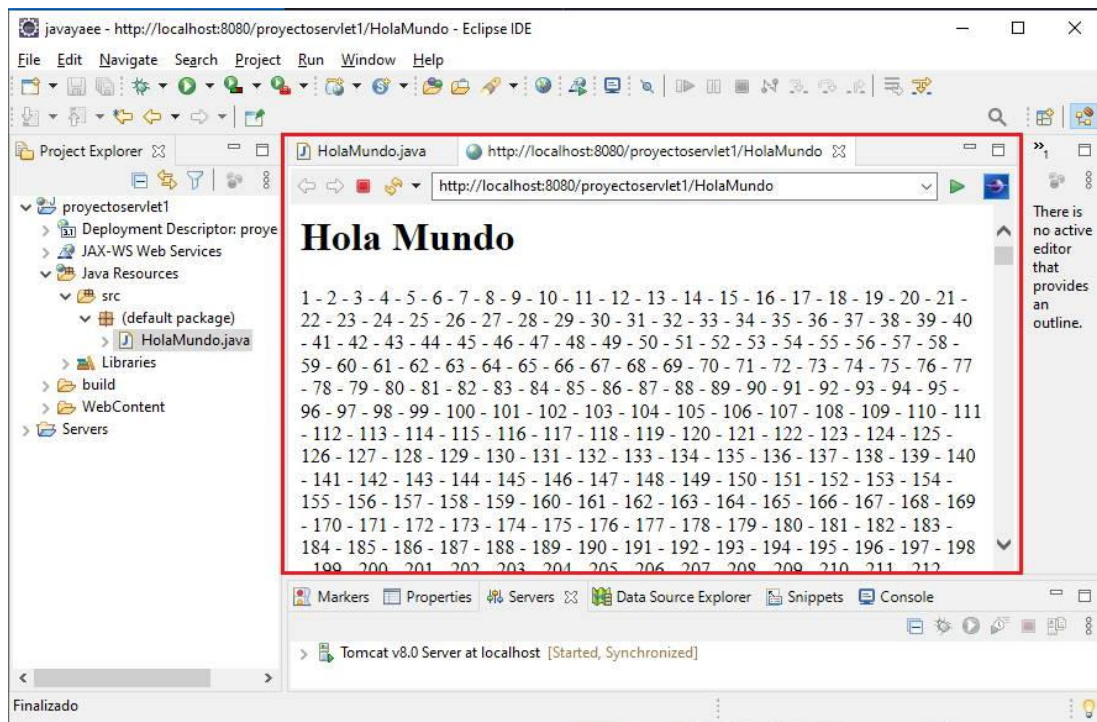
Para probar el servlet que acabamos de codificar debemos presionar el botón derecho del mouse sobre el nombre de la clase y seleccionar "Run on Server":



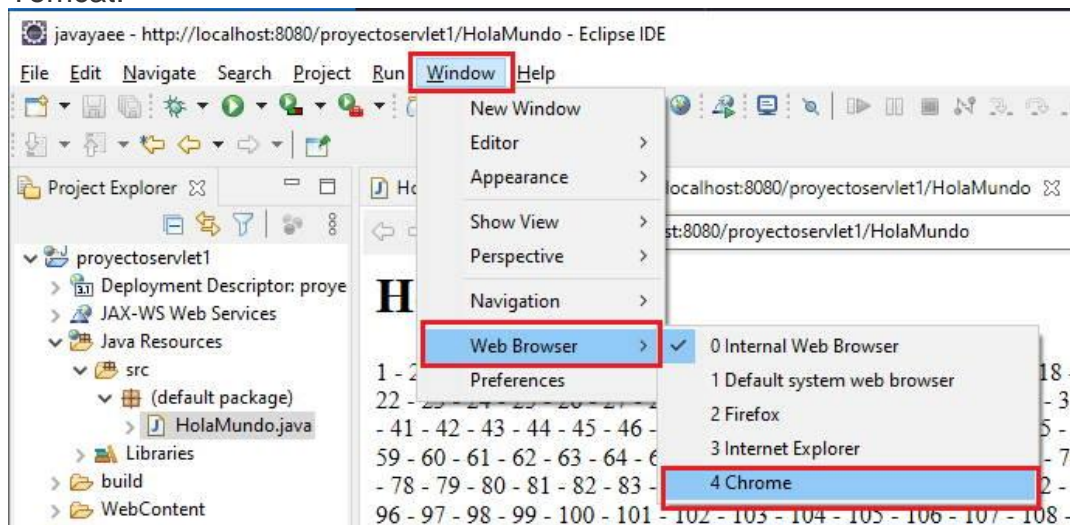
Aparece un diálogo que debemos seleccionar el botón "Finish" ya que está seleccionado el servidor "Tomcat" para ejecutar el servlet:



El resultado de la ejecución del servlet lo podemos ver dentro de una ventana dentro del mismo Eclipse:



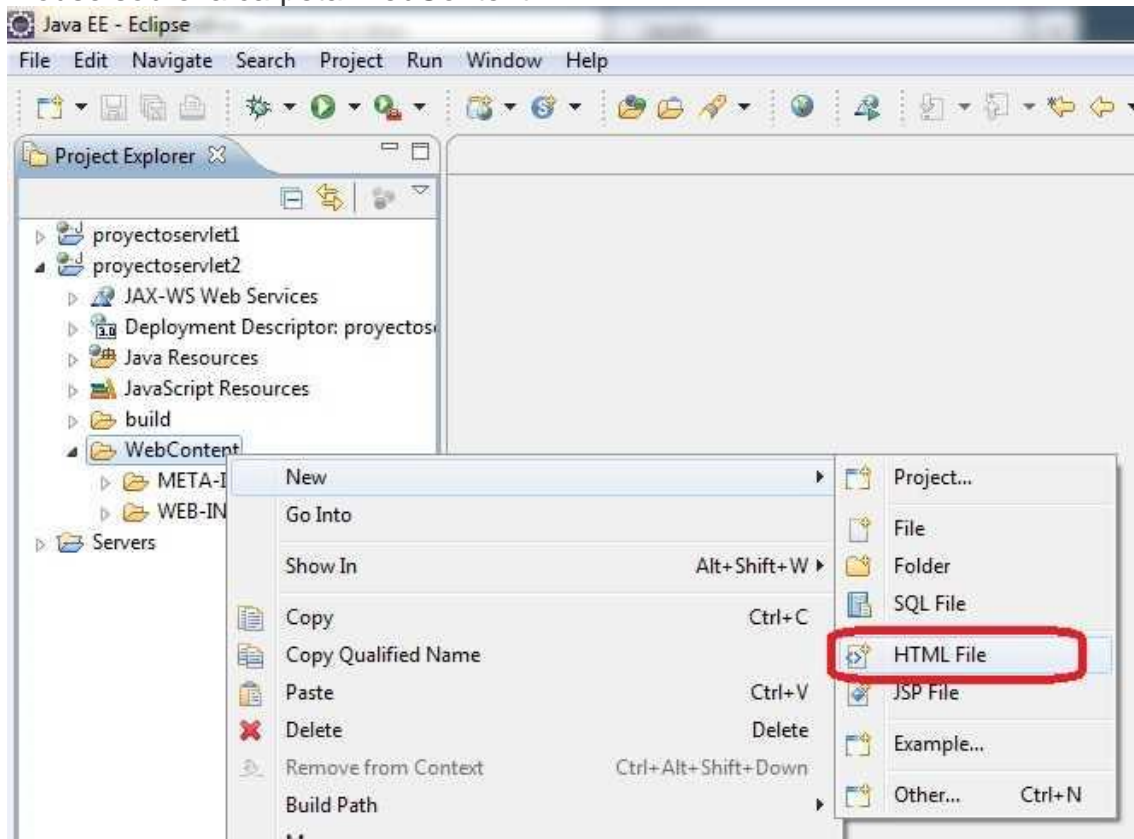
Si queremos que el resultado aparezca en otro navegador podemos configurar desde el menú de Eclipse el navegador que muestra el resultado que devuelve Tomcat:



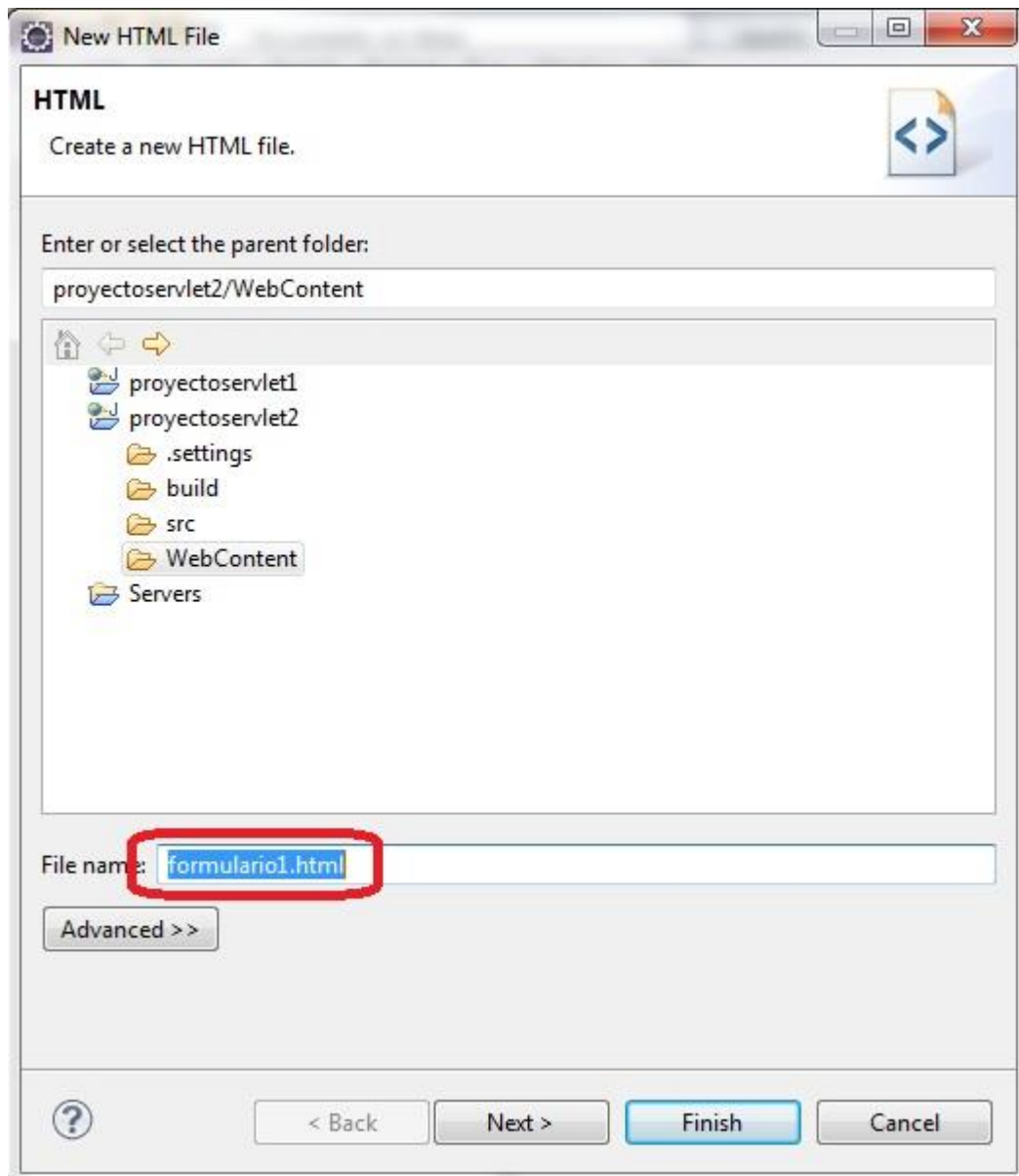
Recuperación de los datos de un formulario HTML en un servlet

Veremos ahora que un servlet puede recibir datos de un formulario HTML. Implementaremos un formulario HTML que solicite el ingreso del nombre y clave de un usuario y posteriormente recuperaremos los dos datos en un servlet y los mostraremos en otra página generada por el servlet.

Primero crearemos un proyecto llamado proyectoservlet2. Para crear el archivo html con el formulario presionamos el botón derecho del mouse sobre la carpeta WebContent:



En el diálogo siguiente especificamos el nombre del archivo html, en nuestro caso lo llamaremos formulario1.html:



Codificamos la página html con el formulario web que solicita el ingreso del nombre de usuario y su clave:

```
<!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=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="RecuperacionUsuario">
Ingrese nombre de usuario:
<input type="text" name="usuario" size="20"><br>
Ingrese clave:
<input type="password" name="clave" size="20"><br>
<input type="submit" value="confirmar">
</form>
</body>
</html>
```

Lo más importante cuando creamos el formulario web es la especificación de la propiedad action de la marca form con el nombre el servlet que recuperará los datos del formulario:

<form method="post" action="RecuperacionUsuario">

Ahora creamos el servlet que como vemos deberá llamarse "RecuperacionUsuario". El código fuente del servlet es el siguiente:

```
import java.io.IOException;
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;

/**
 * Servlet implementation class RecuperacionUsuario
 */
@WebServlet("/RecuperacionUsuario")
public class RecuperacionUsuario extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RecuperacionUsuario() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head></head>");
        out.println("<body>");

        out.println("Usuario:");
        String usu=request.getParameter("usuario");
        out.println(usu);
        out.println("<br>");
        out.println("Clave:");
        String cla=request.getParameter("clave");
        out.println(cla);

        out.println("</body>");
        out.println("</html>");
    }
}
```

Como podemos ver en el código fuente de la clase RecuperacionUsuario debemos implementar todo el código en el método doPost, ya que este se ejecuta cuando se tiene un formulario HTML y se especificó en el HTML en la propiedad method el valor post:

<form method="post" action="RecuperacionUsuario">

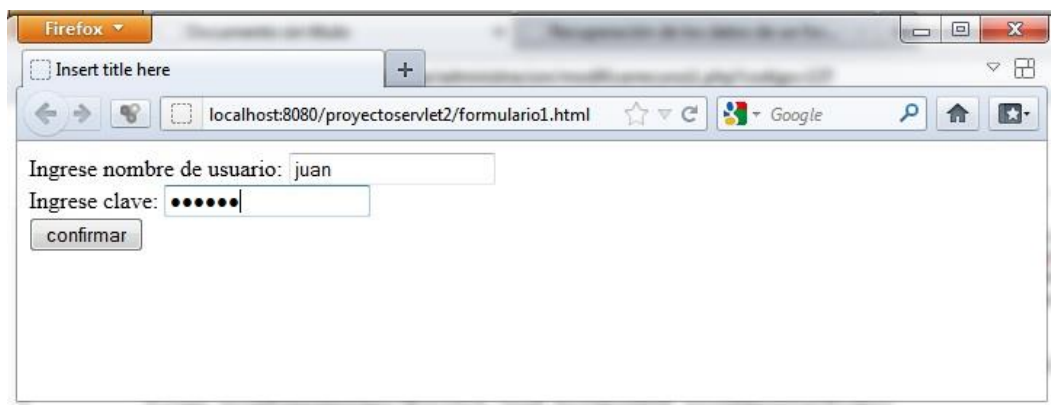
Para recuperar los datos de los controles text y password del formulario HTML el objeto request de la clase HttpServletRequest dispone de un método llamado `getParameter` que le indicamos el nombre del control a recuperar:

`String usu=request.getParameter("usuario");`

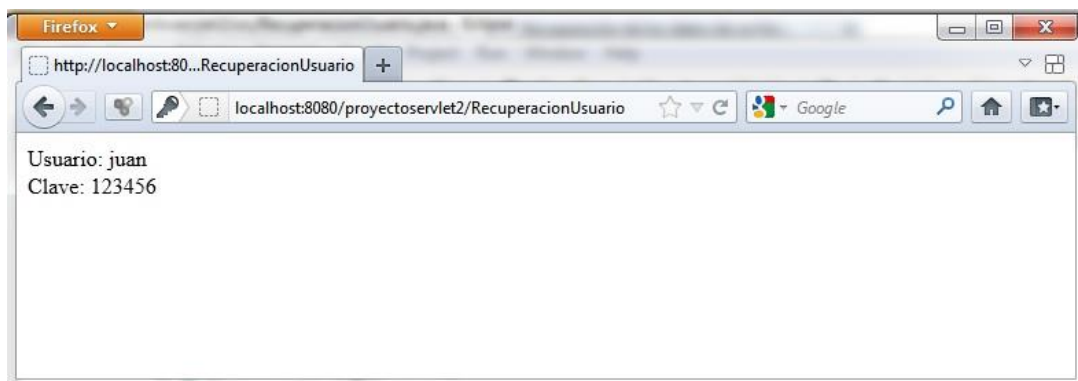
Luego de recuperarlo procedemos a mostrarlo dentro de la página generada:

`out.println(usu);`

Para probar nuestra aplicación debemos presionar el botón derecho sobre el `formulário1.html` y seleccionar "Run As" -> "Run on Server", luego el resultado de la ejecución en el navegador:



Y

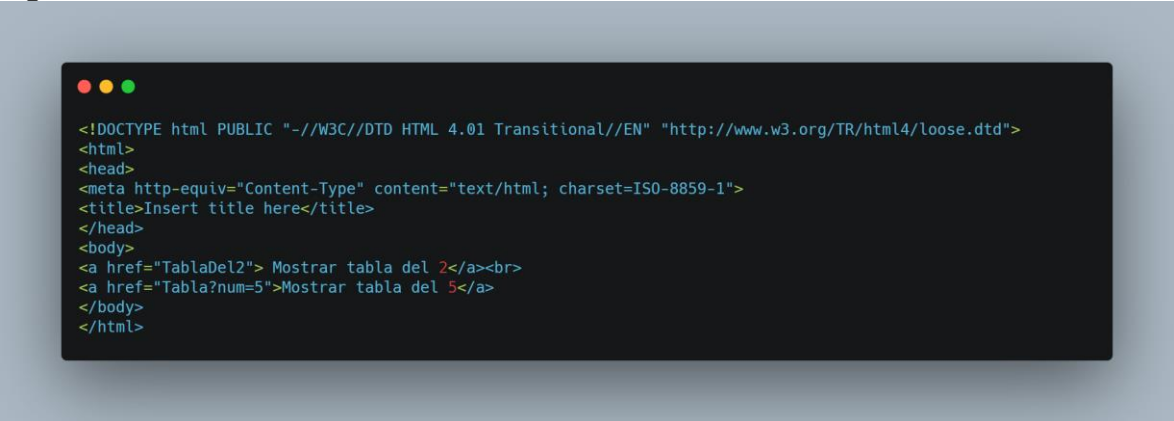


Llamada a servlet desde un hipervínculo (con y sin parámetros)

Problema:

Confeccionaremos una página HTML con dos hipervínculos a dos servlet. El primero tiene por objetivo mostrar la tabla de multiplicar del 2, el segundo servlet llega como parámetro el número del cual queremos mostrar la tabla de multiplicar.

Primero creamos un proyecto llamado proyectoservlet4. Seguidamente creamos el archivo HTML (presionamos el botón derecho del mouse sobre la carpeta WebContent y creamos un archivo HTML llamado menu.html) y codificamos lo siguiente:



```
<!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=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="TablaDel2">Mostrar tabla del 2</a><br>
<a href="Tabla?num=5">Mostrar tabla del 5</a>
</body>
</html>
```

El primer hipervínculo en la propiedad href indicamos el nombre del servlet a ejecutar:

**Mostrar tabla del 2
**

El segundo hipervínculo llama al servlet Tabla y pasa un parámetro num con el valor 5 (este valor se rescatará posteriormente desde el servlet):

Mostrar tabla del 5

Ahora crearemos un servlet llamado TablaDel2:


```

import java.io.IOException;
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;

/**
 * Servlet implementation class TablaDel2
 */
@WebServlet("/TablaDel2")
public class TablaDel2 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public TablaDel2() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head></head>");
        out.println("<body>");
        out.println("<h1>Tabla del 2</h1>");
        for(int f=2;f<=20;f=f+2) {
            out.println(f);
            out.println(" - ");
        }
        out.println("</body>");
        out.println("</html>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}

```

Es importante hacer notar que la anotación del servlet coincide con la propiedad href del hipervínculo de la página HTML:

@WebServlet("/TablaDel2")

public class TablaDel2 extends HttpServlet {

En el método doGet procesamos la petición mediante la generación de la página dinámica:

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

// TODO Auto-generated method stub

PrintWriter out = response.getWriter();

out.println("<html>");


```

out.println("<head></head>");

out.println("<body>");

out.println("<h1>Tabla del 2</h1>");

for(int f=2;f<=20;f=f+2) {

    out.println(f);

    out.println(" - ");

}

out.println("</body>");

out.println("</html>");

}

```

Ahora crearemos el segundo servlet llamado Tabla:

```

import java.io.IOException;
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;

/**
 * Servlet implementation class Tabla
 */
@WebServlet("/Tabla")
public class Tabla extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Tabla() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head></head>");
        out.println("<body>");
        String cad=request.getParameter("num");
        int valor=Integer.parseInt(cad);
        out.println("<h1>Tabla del "+cad+"</h1>");
        for(int f=valor;f<=valor*10;f=f+valor) {
            out.println(f);
            out.println(" - ");
        }
        out.println("</body>");
        out.println("</html>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}

```

Para recuperar el parámetro del hipervínculo llamamos al método `getParameter` del objeto `request`. Debemos pasar un `String` con el nombre del parámetro y el mismo nos retorna el valor del parámetro:

```
String cad=request.getParameter("num");
```

Redireccionamiento a otro sitio o página desde un servlet

Problema:

Confeccionar un formulario html que solicite el ingreso de un sitio web. Cuando se presione un botón redireccione a dicha web.

Crearemos un proyecto llamado `proyectoservlet5` y dentro del mismo un archivo HTML:



```
<!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=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="get" action="Redireccionamiento">
Ingrese la dirección de la página que quiere visitar(Ej. www.google.com):
<input type="text" name="direccion" size="60">
<br>
<input type="submit" value="Ir">
</form>
</body>
</html>
```

Como vemos cuando se presiona el botón submit se envían los datos al servlet llamado `Redireccionamiento`:

```
<form method="get" action="Redireccionamiento">
```

Creamos ahora un servlet llamado `Redireccionamiento`:

```

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Redireccionamiento
 */
@WebServlet("/Redireccionamiento")
public class Redireccionamiento extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Redireccionamiento() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        String dire=request.getParameter("direccion");
        response.sendRedirect("http://"+dire);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}

```

Para redireccionar el objeto request de la clase HttpServletRequest tiene un método llamado sendRedirect que le pasamos como parámetro un String con la dirección del sitio que debe devolver el servlet (en lugar del servlet propiamente dicho)

La dirección del sitio a redireccionar la extraemos con el método getParameter como hemos visto:

```

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

    // TODO Auto-generated method stub

    String dire=request.getParameter("direccion");

    response.sendRedirect("http://"+dire);

}

```