

CURSO DE SERVLETS Y JSP' S

HOLA MUNDO CON SERVLETS



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO DE SERVLETS Y JSP' S

www.globalmentoring.com.mx

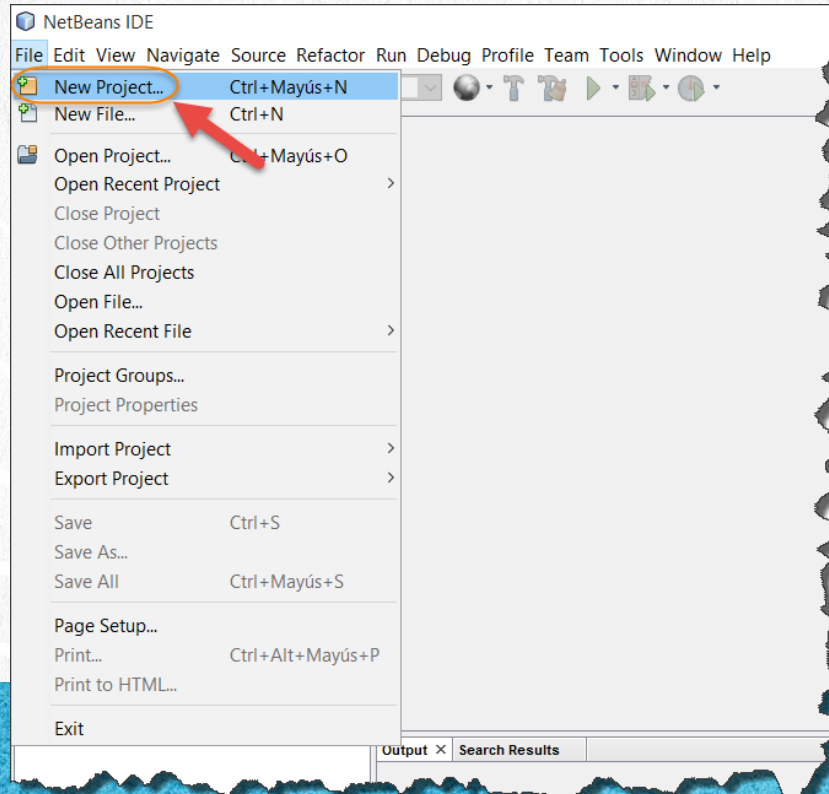
OBJETIVO DEL EJERCICIO

Crear un proyecto Web para crear un HolaMundo con Servlets.
Al finalizar deberemos observar lo siguiente:



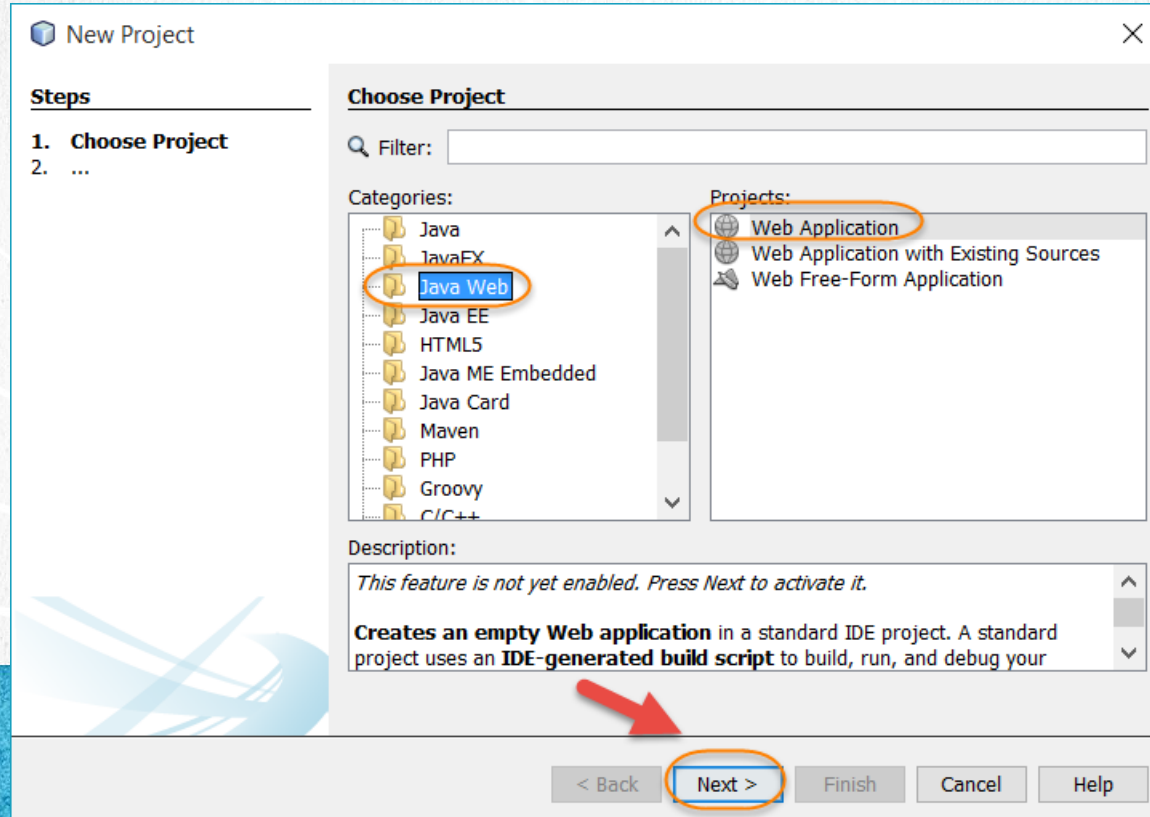
PASO 1. CREACIÓN DEL PROYECTO

Creamos un nuevo proyecto Web, como sigue:



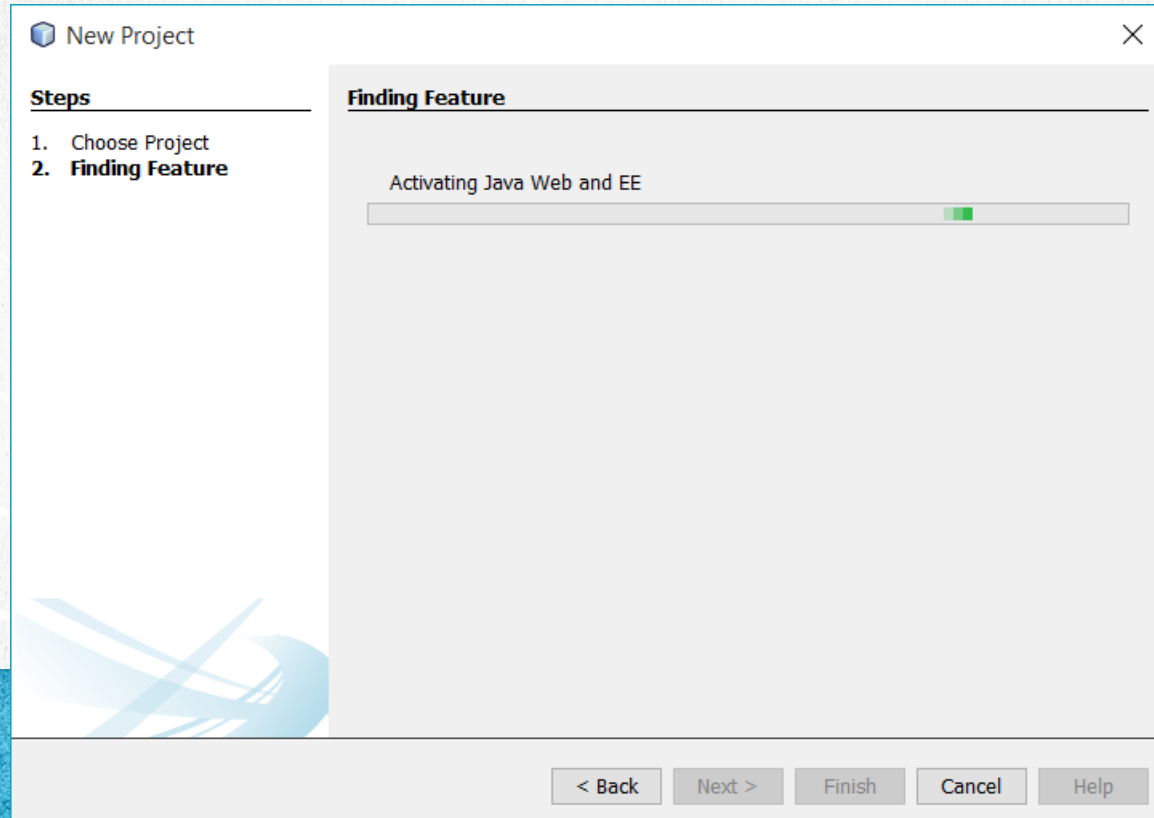
PASO 1. CREACIÓN DEL PROYECTO (CONT)

Creamos un nuevo proyecto Web, como sigue:



PASO 1. CREACIÓN DEL PROYECTO (CONT)

Creamos un nuevo proyecto Web, como sigue:



PASO 1. CREACIÓN DEL PROYECTO (CONT)

Creamos un nuevo proyecto Web, como sigue:

New Web Application

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

PASO 1. CREACIÓN DEL PROYECTO (CONT)

Creamos un nuevo proyecto Web, como sigue:

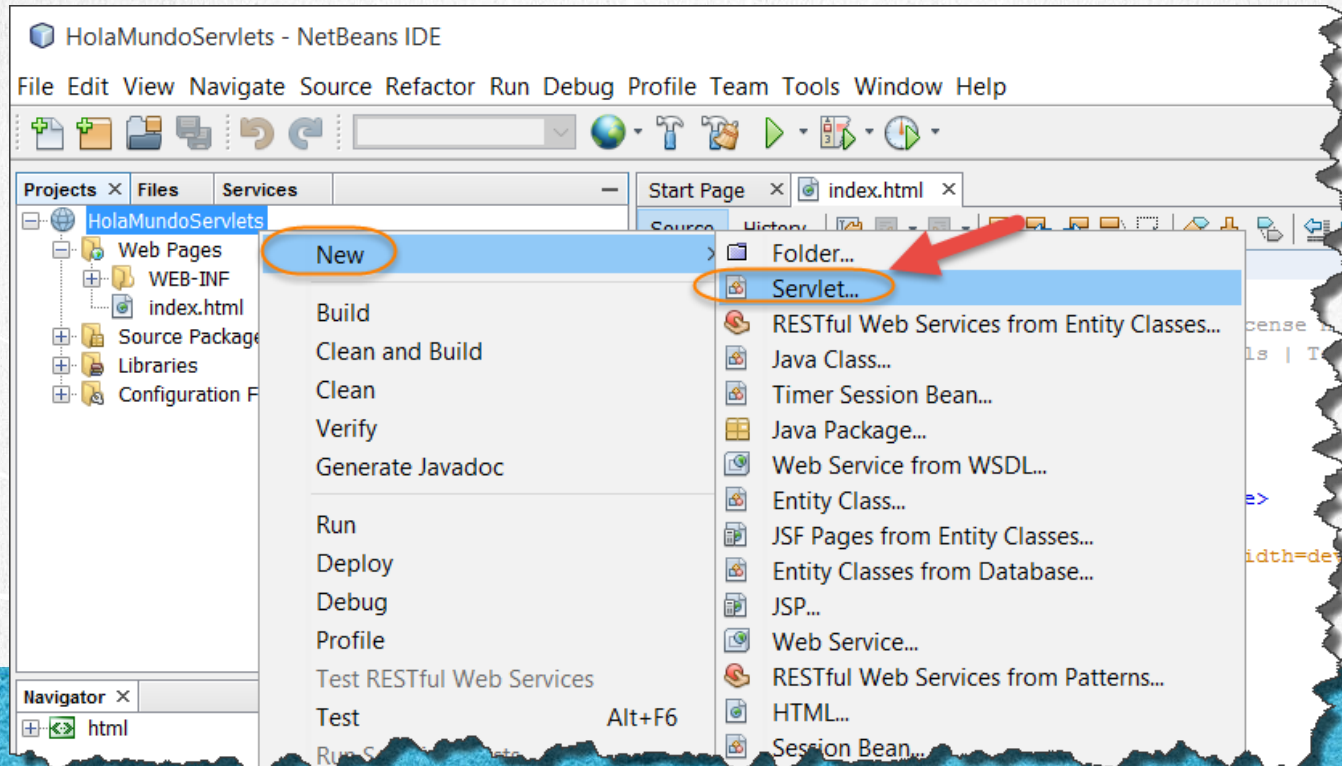
The screenshot shows the 'New Web Application' dialog box with the 'Server and Settings' step selected. The 'Steps' list on the left includes: 1. Choose Project, 2. Name and Location, 3. **Server and Settings**, and 4. Frameworks. The 'Server and Settings' section contains the following fields:

- Add to Enterprise Application:** A dropdown menu currently showing '<None>'.
- Server:** A dropdown menu showing 'GlassFish Server 4.1.1' with an 'Add...' button to its right.
- Java EE Version:** A dropdown menu showing 'Java EE 7 Web'.
- Context Path:** A text input field containing '/HolaMundoServlets'.

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted with a red circle, and a red arrow points to it from the right.

PASO 2. CREAMOS UNA CLASE SERVLET

Creamos una nueva clase Servlet:



PASO 2. CREAMOS UNA CLASE SERVLET (CONT)

Creamos una nueva clase Servlet:

New Servlet

Steps

1. Choose File Type
- 2. Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back **Next >** Finish Cancel Help

PASO 2. CREAMOS UNA CLASE SERVLET (CONT)

Creamos una nueva clase Servlet:

New Servlet

Steps

1. Choose File Type

2. Name and Location

3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☐ Add information to deployment descriptor (web.xml)

Class Name: web.HolaMundo

Servlet Name: HolaMundo

URL Pattern(s): /HolaMundo

Initialization Parameters:

Name	Value
------	-------

New

Edit...

Delete

< Back

Next >

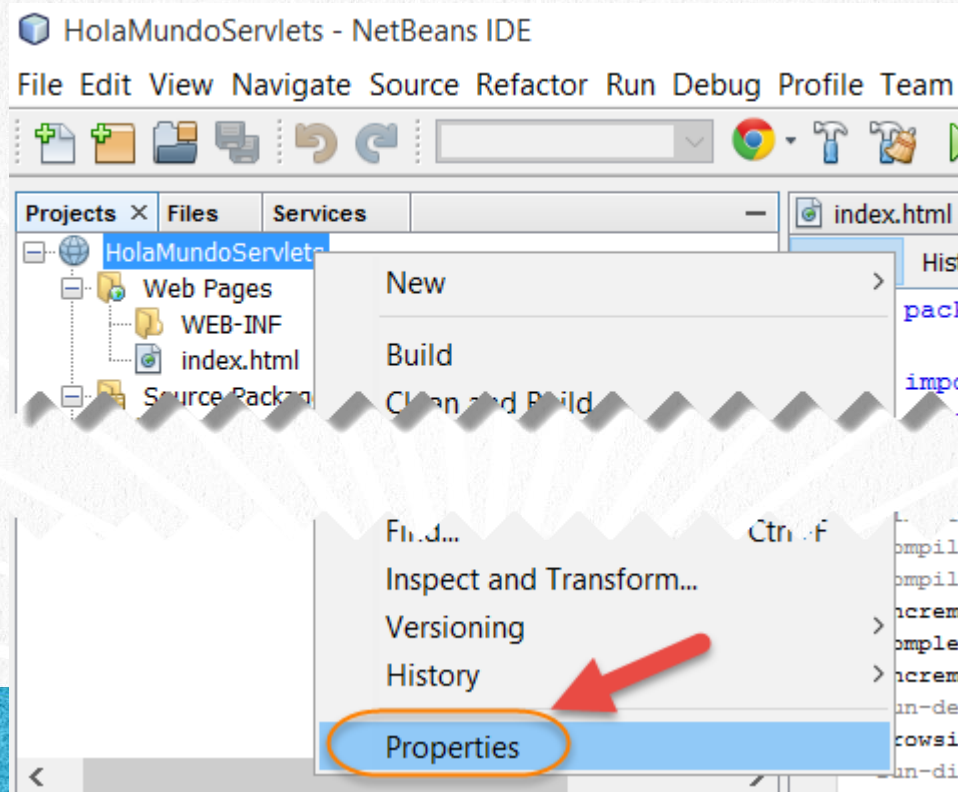
Finish

Cancel

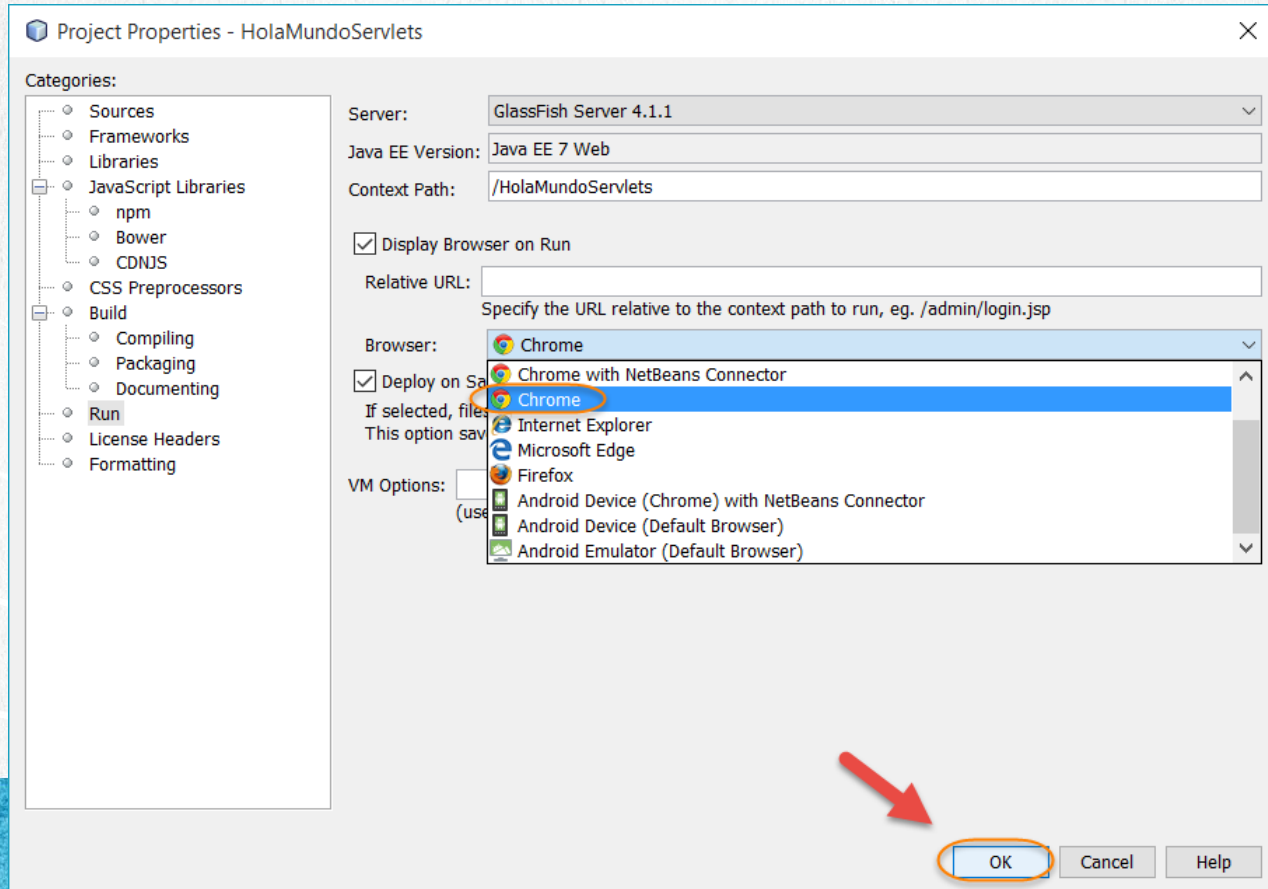
Help

PASO 3. SELECCIONAMOS NAVEGADOR

Seleccionamos a Google Chrome como el navegador por default:



PASO 3. SELECCIONAMOS NAVEGADOR (CONT)



PASO 4. MODIFICAMOS EL CODIGO

Archivo index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HolaMundoServlets</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Llamando al Servlet
    <a href="/HolaMundoServlets/HolaMundo">Hola Mundo</a>
    </h1>
  </body>
</html>
```

PASO 5. MODIFICAMOS EL CODIGO

Archivo HolaMundoServlet.java:

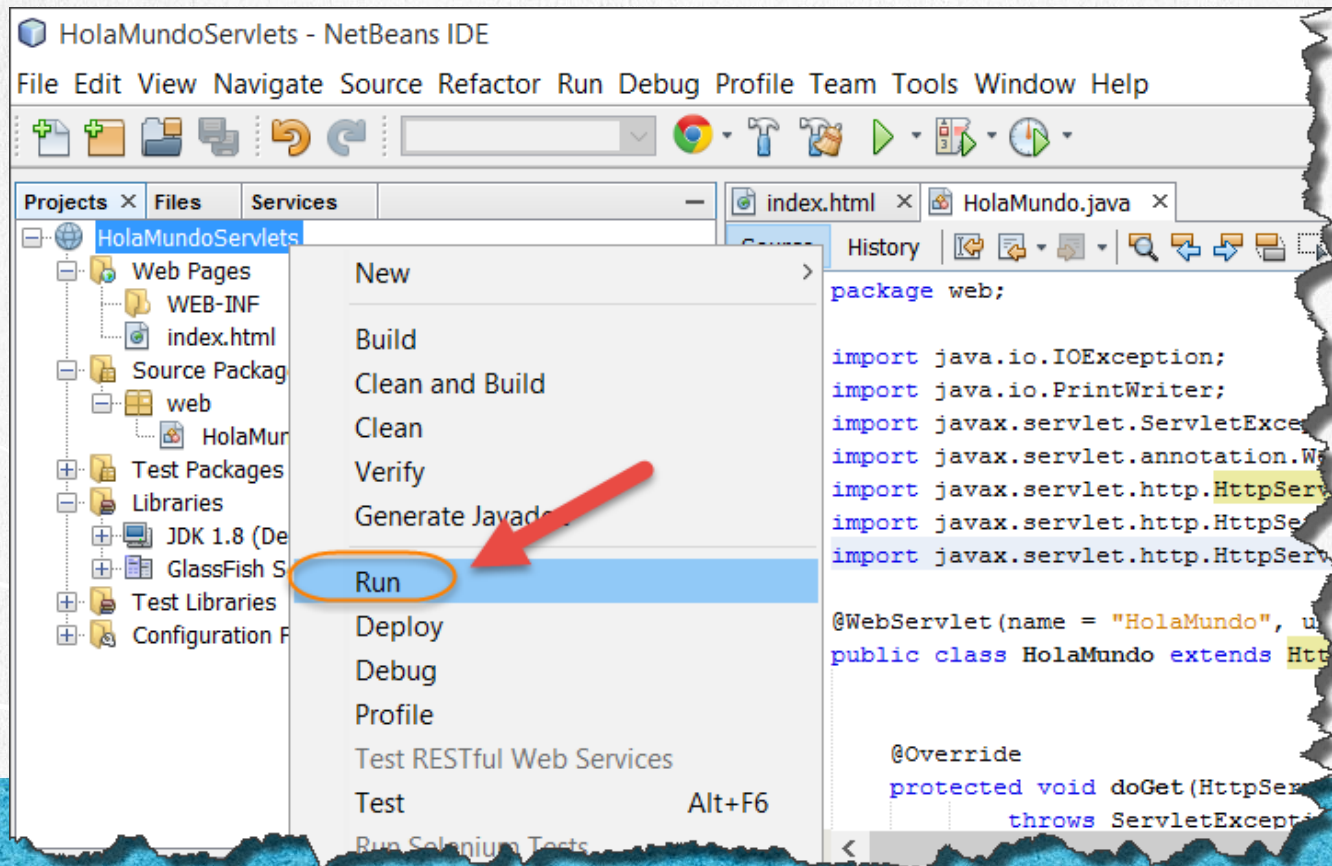
```
package web;

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;

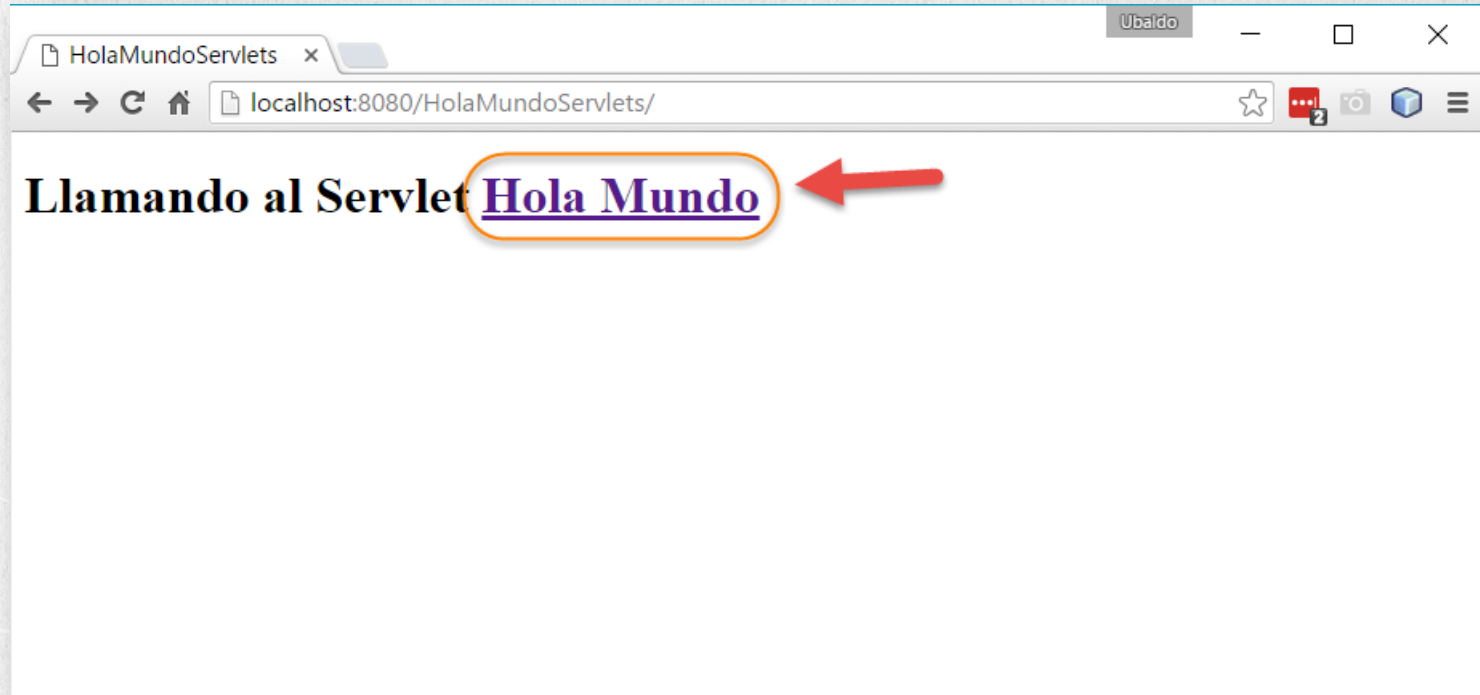
@WebServlet(name = "HolaMundo", urlPatterns = {"/HolaMundo"})
public class HolaMundo extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet HolaMundo</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet HolaMundo, la aplicacion se llama: " + request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

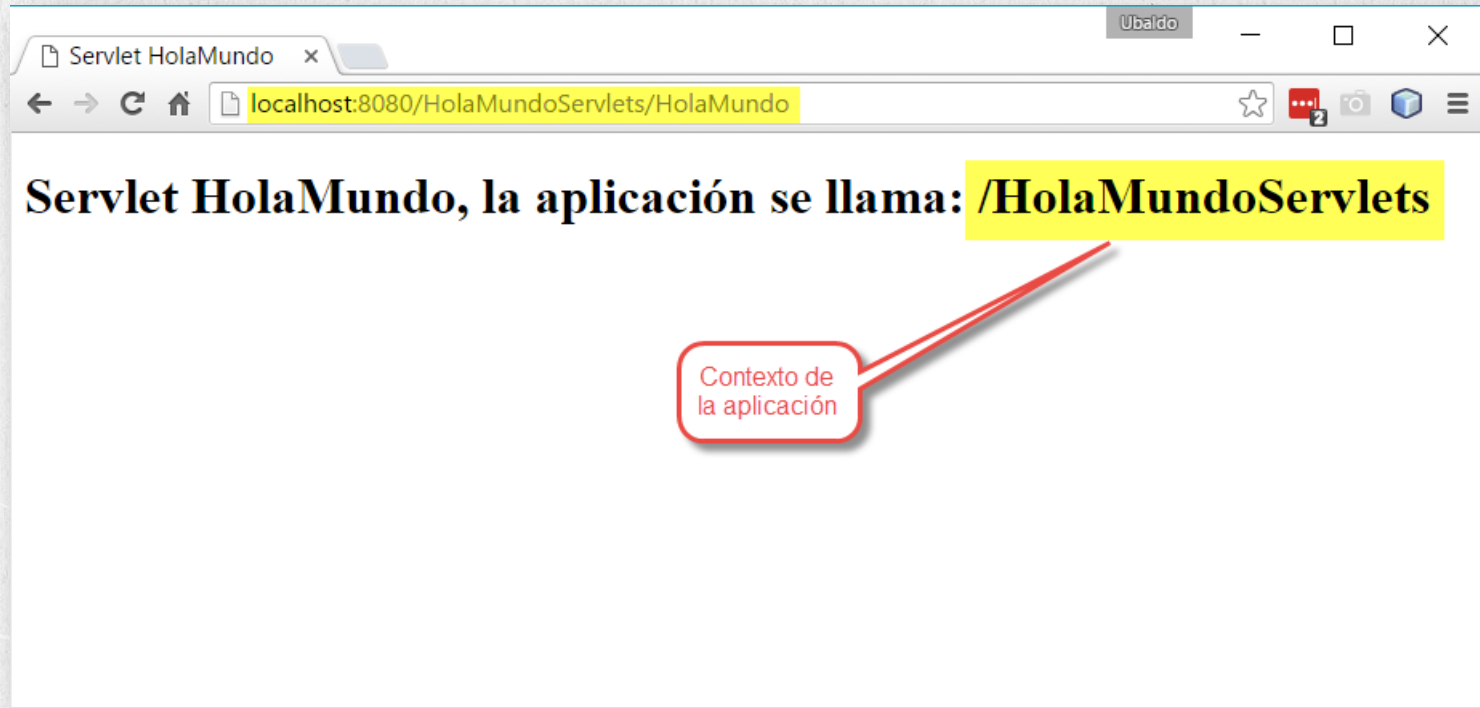

PASO 6. EJECUTAMOS LA APLICACIÓN



PASO 6. EJECUTAMOS LA APLICACIÓN



PASO 6. EJECUTAMOS LA APLICACIÓN (CONT)



CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos creado nuestro primer Servlet. Si bien lo creamos con ayuda del IDE, el código que se agregó es totalmente independiente del IDE que se utilice, de esta manera podemos simplemente crear una clase Java, y el código utilizado será el código que necesita el Servlet y la aplicación Web para funcionar.

Así que este código es independiente de plataforma e independiente del IDE que utilicemos.

CURSO ONLINE

SERVLETS Y JSP' S

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO DE SERVLETS Y JSP' S

www.globalmentoring.com.mx