



Ciclo 3

Semana 2

*Desarrollo de páginas web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.*

Lectura 1 - Creación de un proyecto Web en Java con Eclipse 2021

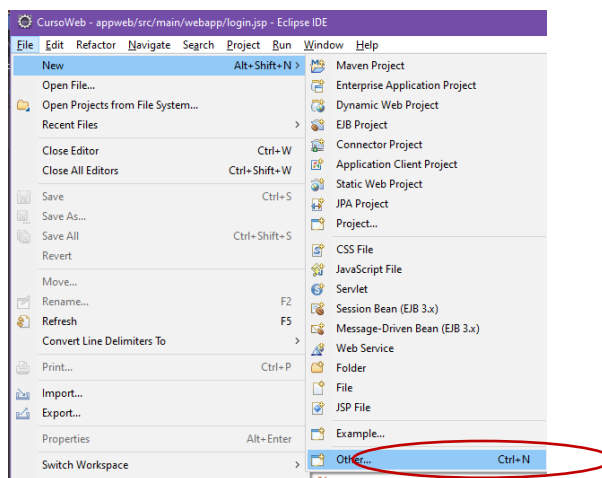


Semana 2

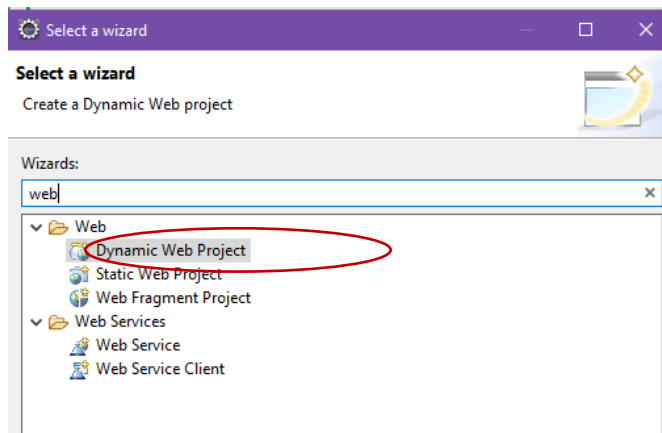
Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

| Creación de un proyecto Web en Java con Eclipse 2021-03

Una vez instalados los paquetes indicados, en particular Eclipse, y Tomcat 9.0, vamos a construir el frontend de la aplicación (o capa de presentación), para lo cual, vamos a crear un nuevo proyecto Web dinámico, para lo cual vamos a esta opción de menú en Eclipse: File -> New -> Other



Una vez ahí, vamos a buscar la opción Dynamic Web Project, buscando en la opción “Wizards”, la opción Web:

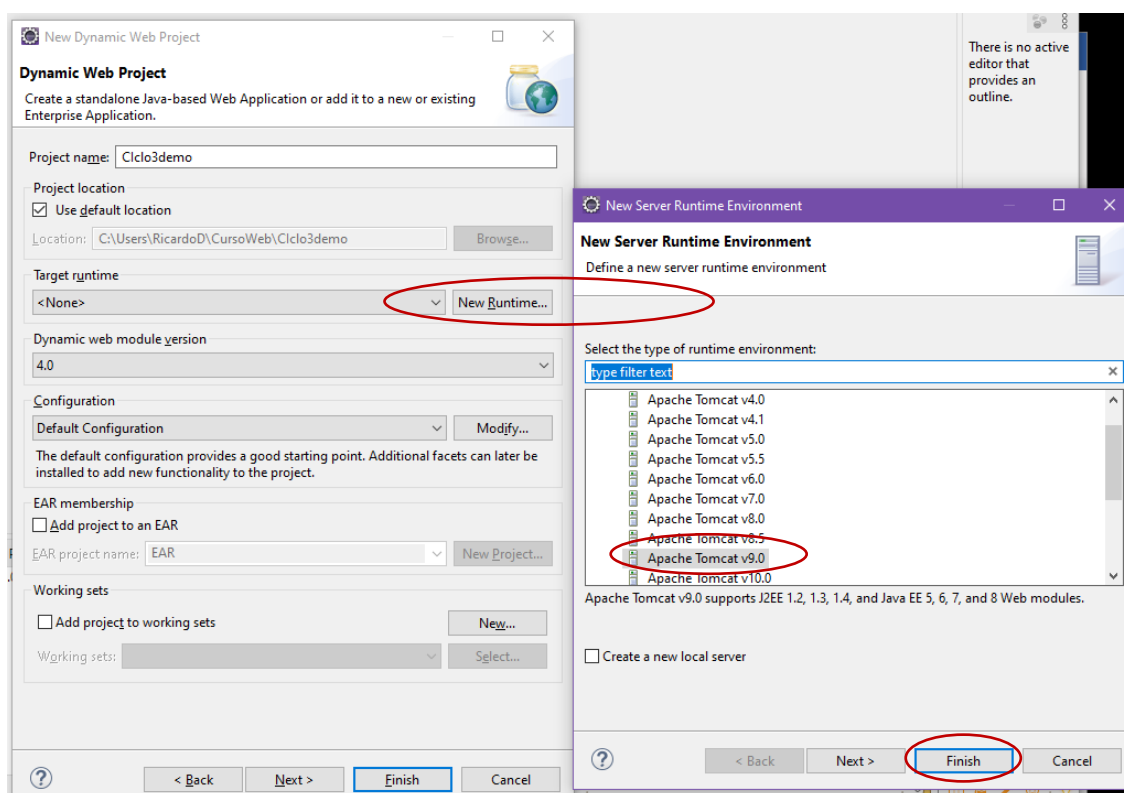




Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

Posteriormente, aparecerá una pantalla solicitando el nombre del proyecto, para lo cual colocamos el nombre de **Ciclo3demo**, luego, debemos configurar el Target Runtime, que es el servidor Tomcat, que ya instalamos en nuestro equipo, para que el proyecto pueda correrse en un servidor Web, de la siguiente forma:

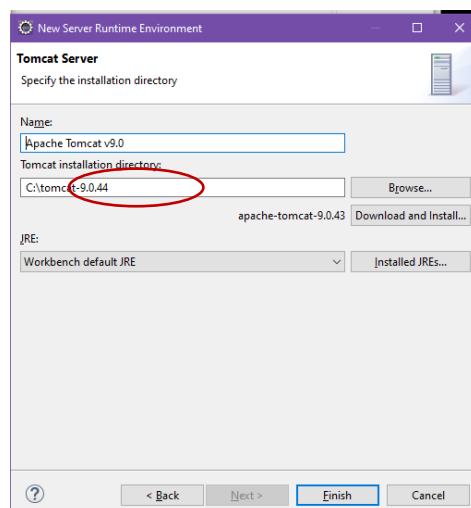


Nos saldrá la siguiente ventana, donde ubicaremos la carpeta donde está la instalación de Apache Tomcat 9.0:



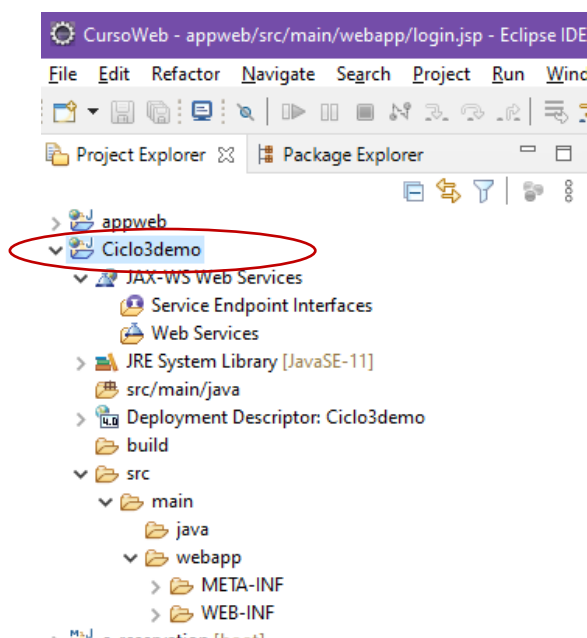
Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.



Una vez, hayamos indicado la ruta de la carpeta donde instalamos Tomcat 9.0, presionamos Finish.

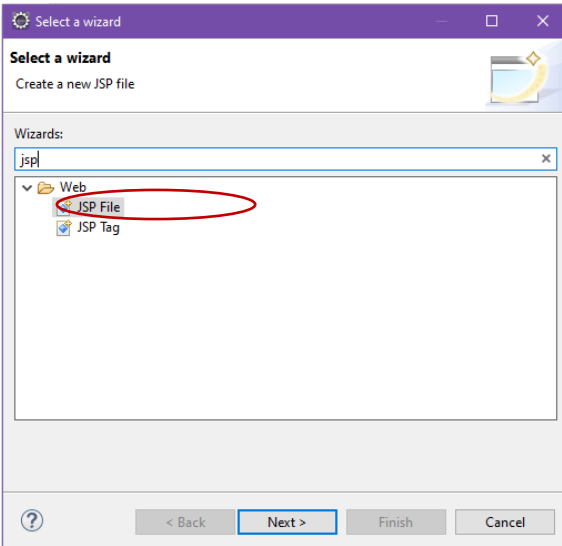
Finalmente, en la pestaña de Project Explorer, aparecerá la carpeta del proyecto, lo cual debe lucir igual o similar a esto, con las opciones de carpetas que Eclipse configura por defecto para este tipo de proyecto así:



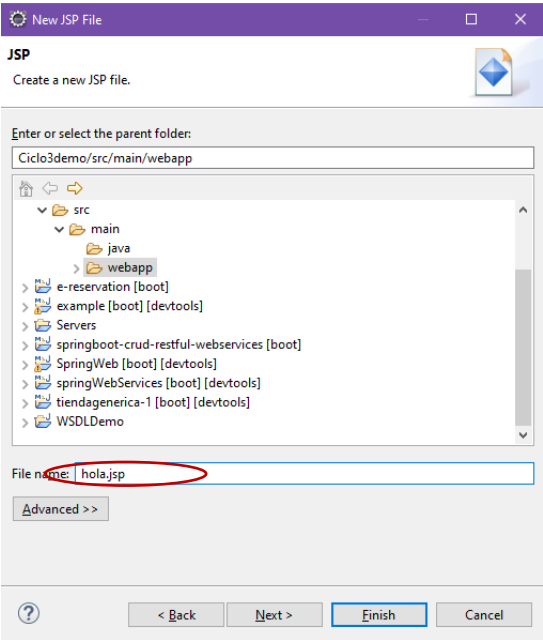
Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

Ahora vamos a crear nuestra primera página Web en JSP, para lo cual, y de la misma forma que lo hicimos creando el Project de Eclipse, Vamos a: File -> New -> other, y en los wizards, buscamos JSP, de la siguiente forma:



Posteriormente, damos **Next**, y colocamos el nombre de nuestra página JSP la cual la llamaremos hola.jsp, lo cual deberá verse de esta forma:

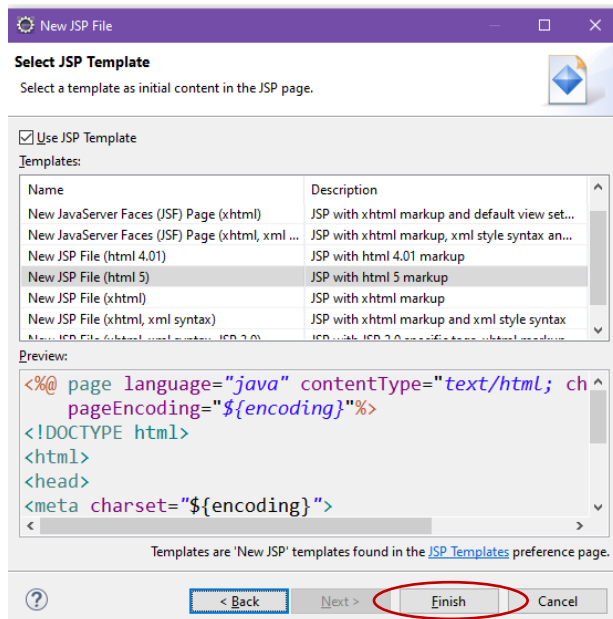




Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

Hacemos clic en Next, y vemos las diferentes opciones de platillas que tiene Eclipse para diseñar una página JSP, que se ve como esto:



Dejamos la opción por defecto, y, finalmente, hacemos clic en **Finish**. Deberá verse una página HTML como esta:

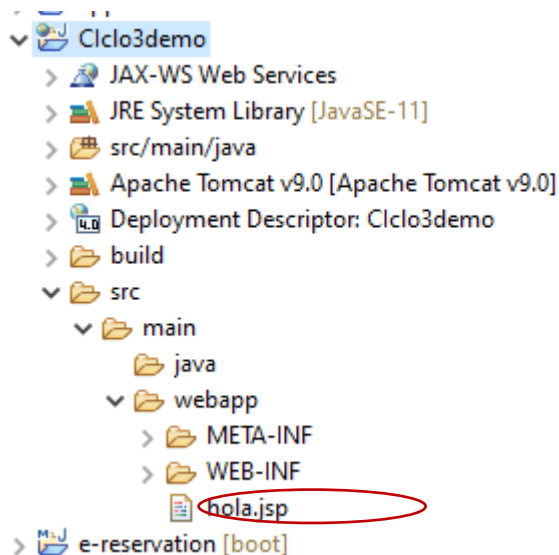
```
hola.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```




Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

En el Project Explorer, la página estará ubicada debajo de la carpeta WEB-INF, en este lugar:



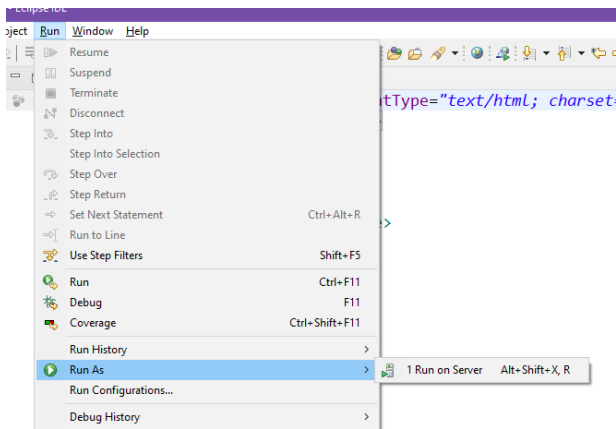
Vamos a agregar algo de contenido a esta página hola.jsp, para lo cual en <body> vamos a colocar un texto en tamaño <h1>, así:

```
1 <%@ page language="java" conte
2   pageEncoding="ISO-8859-1"%
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</titl
8 </head>
9 <body>
10 <h1>Mi primer JSP</h1>
11 </body>
12 </html>
```

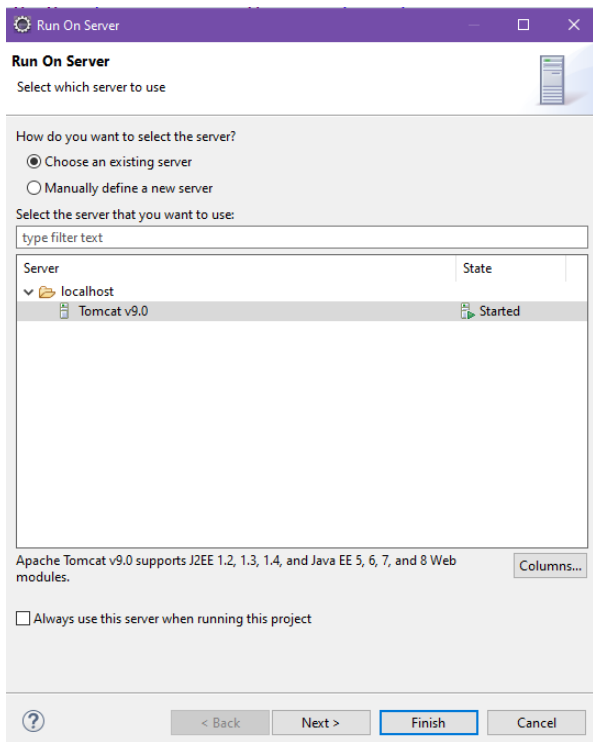
Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1 y modelamiento de bases de datos en MySQL.

Para probar cómo quedó la página, vamos a ejecutarla (Run), para lo cual deberemos definirla a Eclipse en qué servidor estará esta, para lo cual haremos clic en Run -> Run As -> Run On Server.



Luego saldrá una ventana, donde seleccionaremos el servidor que definimos al momento de crear el Project.





Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

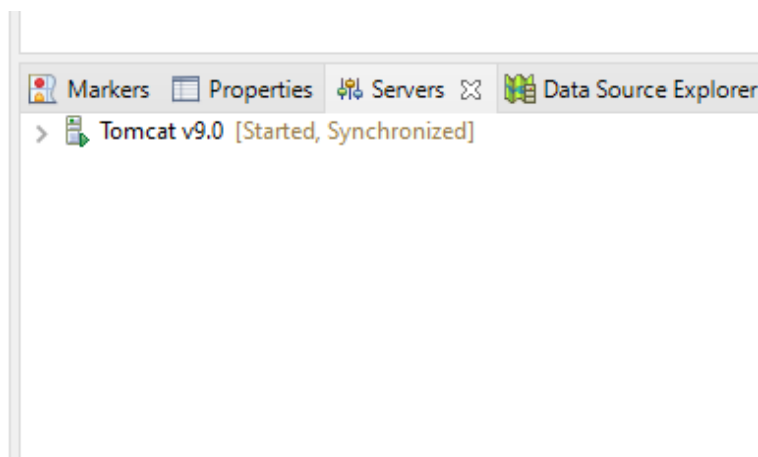
Finalmente, hacemos clic en **Finish**. Puede que salga una ventana que indica que el servidor debe ser reiniciado, si es así, hacemos click en la opción “Restart Server”.

En este momento, ocurren tres cosas: 1) aparece una nueva ventana con nuestra página JSP, 2) en la ventana de abajo, Servers, vemos el servidor Tomcat ejecutándose, y 3) En la ventana de Console, vemos el resultado de la activación del servidor como tal.

1) Página Web



2) Ventana “Servers”:

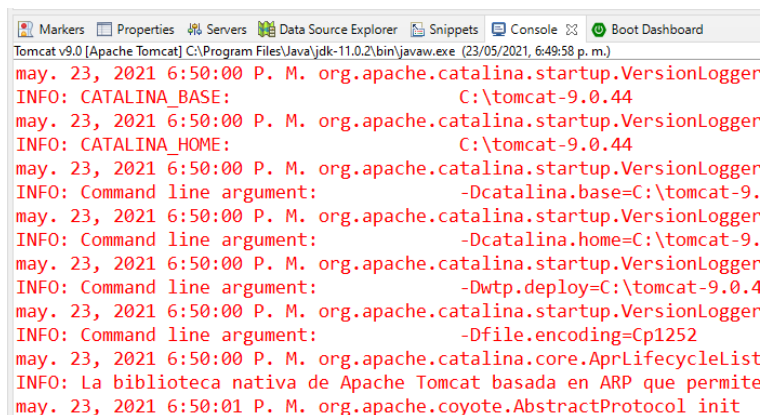




Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

3) Ventana "Console"



```
Tomcat v9.0 [Apache Tomcat] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (23/05/2021, 6:49:58 p. m.)
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: CATALINA_BASE: C:\tomcat-9.0.44
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: CATALINA_HOME: C:\tomcat-9.0.44
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: Command line argument: -Dcatalina.base=C:\tomcat-9.
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: Command line argument: -Dcatalina.home=C:\tomcat-9.
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: Command line argument: -Dwtp.deploy=C:\tomcat-9.0.4
may. 23, 2021 6:50:00 P. M. org.apache.catalina.startup.VersionLogger
INFO: Command line argument: -Dfile.encoding=Cp1252
may. 23, 2021 6:50:00 P. M. org.apache.catalina.core.AprLifecycleList
INFO: La biblioteca nativa de Apache Tomcat basada en ARP que permite
may. 23, 2021 6:50:01 P. M. org.apache.coyote.AbstractProtocol init
```

Se debe prestar atención a cómo se debe conformar la URL de la aplicación, ya que comienza con <http://localhost:8080>. Localhost indica que estamos ejecutando la página Web desde un sitio local en nuestro computador, y 8080 es el puerto TCP por donde Tomcat escucha las peticiones desde el cliente. Luego, se ve el nombre de la aplicación que dimos al momento de crear el Project en Eclipse, y, por último, el nombre de la página JSP.



De la misma forma, podemos copiar la URL completa (<http://localhost:8080/Ciclo3demo/hola.jsp>) y pegarla en cualquier Navegador para verificar funcionamiento.

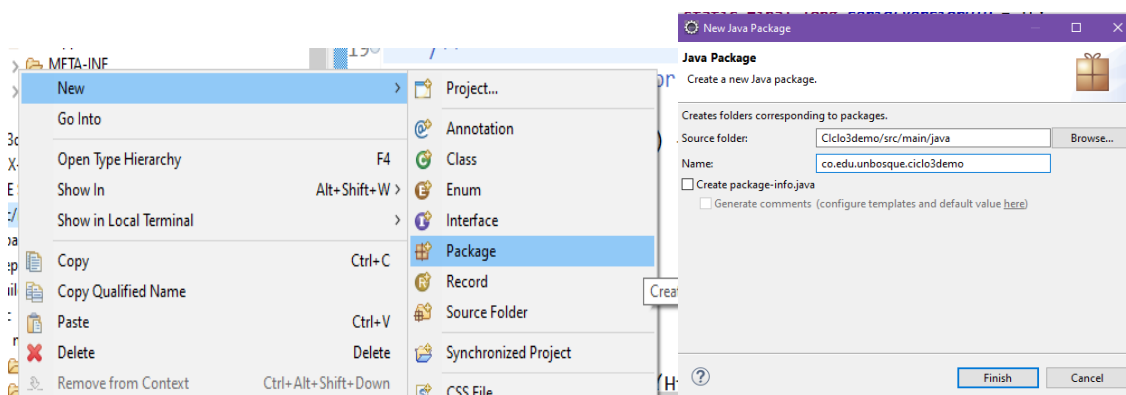
Nada mal, ¿cierto?, ahora debemos crear el Servlet, o el procesador de peticiones que lleguen a nuestra aplicación desde la página JSP, para lo cual lo haremos de la siguiente forma:



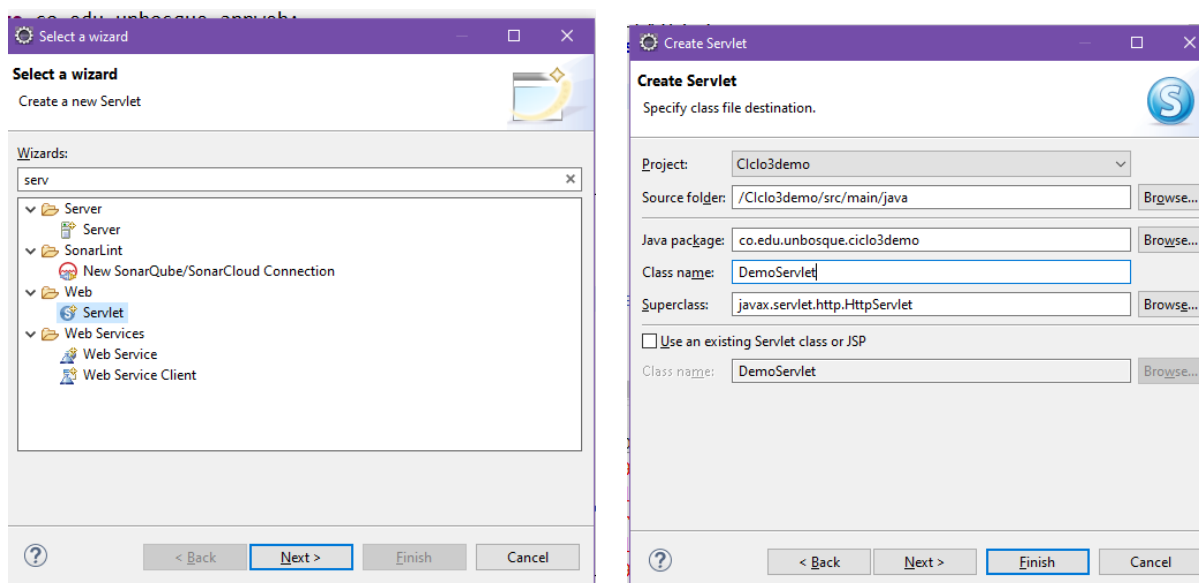
Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

En el Project Explorer, hacemos clic derecho en la carpeta **src/main/java** (que es la carpeta donde colocaremos nuestras clases y packages de java) y haremos un nuevo package llamado **co.edu.unbosque.ciclo3demo**.



Una vez creada, hacemos click, en este nuevo package, y creamos un nuevo Servlet haciendo clic derecho en File -> New -> Other, buscamos en el wizard, el nombre Servlet, y colocamos por nombre **DemoServlet**, como se indica abajo:



Y quedará un archivo en Java, de la siguiente forma:



Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

```
1 package co.edu.unbosque.ciclo3demo;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class DemoServlet
7  */
8 @WebServlet("/DemoServlet")
9 public class DemoServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14      */
15     public DemoServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22      */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         response.getWriter().append("Served at: ").append(request.getContextPath());
26     }
27
28     /**
29      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
30      */
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // TODO Auto-generated method stub
33         doGet(request, response);
34     }
35 }
```

La vista real del código puede variar, dependiendo de las opciones de documentación y comentarios Java que tenga configurado Eclipse. Vemos en el anterior código que se tiene la anotación (o directiva Java que indica que la clase, o el método que le pre tiene un patrón de comportamiento definido), `@WebServlet`, de la siguiente forma: `@WebServlet("/DemoServlet")`, lo cual quiere decir que la clase `DemoServlet` es una clase que será un `Servlet` para las peticiones que se hagan desde cualquier página Web o JSP, en nuestro caso, será desde la página `hola.jsp`.

En nuestro JSP (`hola.jsp`), vamos a colocar un formulario HTML, con tres (3) elementos para que puedan ser procesados por `DemoServlet`, estos son dos etiquetas (`<label>`), dos campos de entrada (`<input>`), y un botón "Enviar". Para lo cual colocaremos el siguiente código HTML en `<body>`:



Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

```
9 <body>
10   <h1>Mi primer JSP</h1>
11
12   <form method="get" action="./DemoServlet">
13     <table>
14       <tr>
15         <td><label>Nombre:</label></td>
16         <td><input type="text" name="nombre"></td>
17         <td><label>Cedula:</label></td>
18         <td><input type="text" name="cedula"></td>
19       </tr>
20       <tr>
21         <td><input type="submit" value="Enviar"></td>
22       </tr>
23     </table>
24   </form>
25 </body>
```

Nótese que en el método (method) de acceso a la información del `<form>`, se ha colocado GET, y, al momento de presionar el botón “Enviar”, se pasarán al Servlet definido en la sección (action) – el cual, por supuesto es DemoServlet (tener en cuenta el ./, para que se pueda procesar de la siguiente forma: <http://localhost:8080/Ciclo3demo/DemoServlet?nombre=jorge&cedula=123>), esto es, los datos del nombre y cédula serán visibles en el navegador.

Volviendo a DemoServlet.java, vemos que automáticamente genera, además de un constructor dos (2) métodos, los cuales tienen por nombre doGet, y doPost. Estos dos métodos tienen como propósito el procesamiento de la acción que reciba del JSP, según el método que se utilice para esto. En nuestro caso, sería doGet el que procese, para lo cual, en este método, escribiremos dos instrucciones que permiten obtener los parámetros de nombre y cédula, haciendo un Request.getParameter, para luego procesarlos con un mensaje de vuelta por medio de PrintWriter, de la siguiente forma:



Semana 2

Desarrollo de páginas Web dinámicas en Java parte 1
y modelamiento de bases de datos en MySQL.

```
30 protected void doGet(HttpServletRequest request, HttpServletResponse response) th
31 // TODO Auto-generated method stub
32 response.getWriter().append("Served at: ").append(request.getContextPath());
33 String nombre = request.getParameter("nombre");
34 String cedula = request.getParameter("cedula");
35 PrintWriter writer = response.getWriter();
36 if( nombre != null && cedula != null) {
37     writer.println("Bienvenido "+nombre+" a mi JSP");
38 }
39 else
40     writer.println("Error: Nombre o cédula faltante!");
41 writer.close();
42 }
```

Como vemos en las líneas 33, y 34, se definen dos variables `nombre` y `cedula`, las cuales reciben lo que `hola.JSP` envían por GET, posteriormente, se define un objeto `PrintWriter` para poder enviar una respuesta HTML, por medio de `println`, la cual, dependerá de la condición definida en la línea 36. Al ejecutarse la página nuevamente en el navegador, deberá verse de la siguiente forma:

hola.jsp ServletLogin.java DemoServlet.java login.jsp Insert title here

http://localhost:8080/Ciclo3demo/hola.jsp

Mi primer JSP

Nombre: Cedula:

Colocamos por nombre “juan”, por cédula “123”, y cuando hagamos clic en el botón “Enviar”, deberá verse esto:

hola.jsp ServletLogin.java DemoServlet.java login.jsp http://localh

http://localhost:8080/Ciclo3demo/DemoServlet?nombre=juan&cedula=123

Served at: /Ciclo3demoBienvenido juan a mi JSP