

7.4. Despliegue de *Servlets*

En esta práctica desarrollaremos una aplicación basada en *Servlets* y la desplegaremos en el servidor *Tomcat*.

1. Creación de una aplicación HolaMundo.

1.1. Inicia sesión en el equipo **DesarrolloW7XX** y arranca *Eclipse*.

1.2. En *Eclipse*, crea un nuevo *Dynamic Web Project* de nombre **HolaMundo**. Marca la opción de crear un descriptor de despliegue de la aplicación, Figura 7.30.

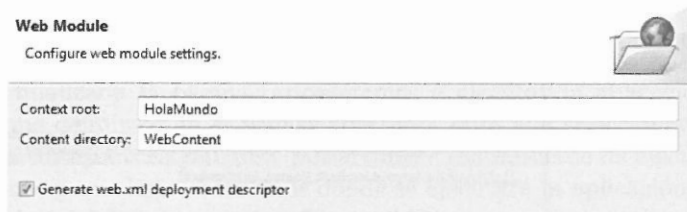


Figura 7.30: Crear descriptor de despliegue de la aplicación

1.3. Si observas la estructura de carpetas del proyecto creado, Figura 7.31.

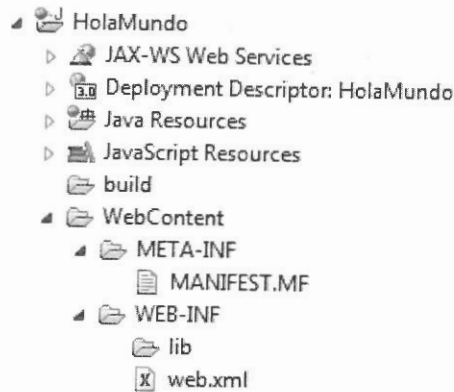


Figura 7.31: Estructura carpetas proyecto

Podrás ver los siguientes elementos:

- **Deployment Descriptor.** Acceso al descriptor de despliegue guardado en el fichero **WEB-INF\web.xml**.
- Carpeta **WebContent**. Donde colocaremos todos los recursos Web, como ficheros *HTML*, *JSPs*, imágenes, etc. Los ficheros no colocados en este directorio, o en algún subdirectorio del mismo, no estarán disponibles cuando la aplicación se ejecute en el servidor. Todos los ficheros fuera de este directorio se consideran archivos de desarrollo, no de producción. Por ejemplo ficheros *.java*, *.sql*, etc.
- Carpeta **WebContent\META-INF**. Contiene el fichero **MANIFEST.MF**, utilizado para mapear clases de ficheros JAR existentes en otros proyectos pertenecientes al mismo *Enterprise Application project*.
- Carpeta **WebContent\WEB-INF**. Este directorio contiene el descriptor de despliegue de la aplicación, **web.xml**.
- Carpeta **Java Resources**. Donde colocaremos nuestros *Servlets* y nuestros ficheros *.java*.

1.4. Ahora añade una página **HolaMundo.html** a nuestro proyecto y editála para incluir el mensaje Hola Mundo.

1.5. En vez de ejecutar directamente el proyecto en el servidor *Tomcat* que tenemos registrado en *Eclipse*, vamos a exportarlo como un archivo **war**, *Web archives*, Figuras 7.32 y 7.33. Puedes guardar el fichero exportado en cualquier ubicación.

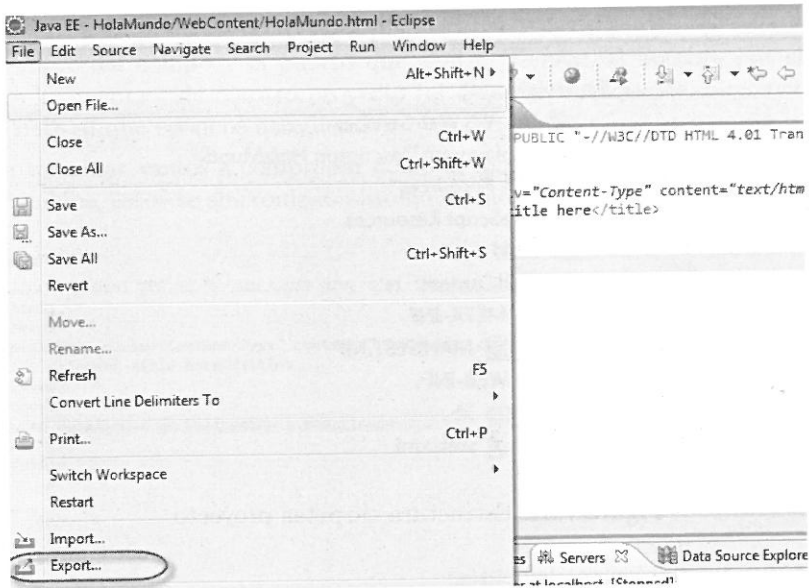


Figura 7.32: Exportar proyecto

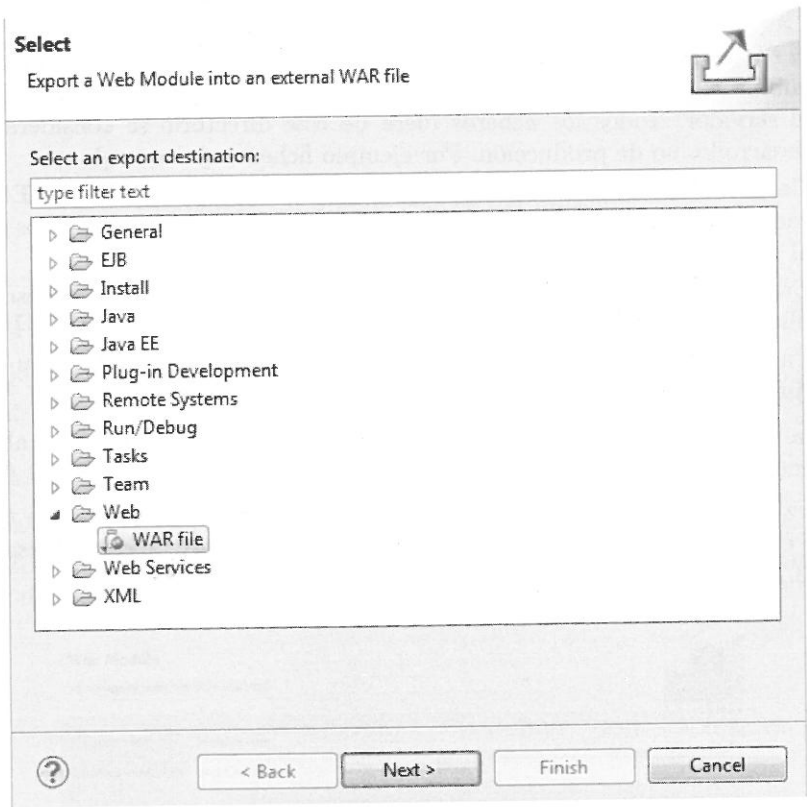


Figura 7.33: Exportar proyecto

- 1.6. Inicia ahora el servidor *Tomcat*, ejecutando el *script startup*, situado en **CATALINA_HOME\bin**.
- 1.7. Para desplegar el proyecto en *Tomcat*, copia el fichero **HolaMundo.war** en la carpeta **CATALINA_HOME\webapps**. Automáticamente se desplegará el proyecto.
- 1.8. A continuación accede a la dirección `http://localhost:8080/HolaMundo/HolaMundo.html` desde el navegador.
- 1.9. En *Eclipse*, renombra el fichero **HolaMundo.html**, llamándolo **index.html** y vuelve a exportar y desplegar el proyecto. Comprueba cómo ahora puedes acceder también con las *urls*:
 - `http://localhost:8080/HolaMundo/index.html`
 - `http://localhost:8080/HolaMundo`
- 1.10. Esto es debido a que el **index.html** es uno de los nombres especificados en el descriptor de despliegue como página de inicio.
- 1.11. Para finalizar observa cómo en el directorio **CATALINA_HOME\webapps** se ha creado automáticamente un directorio **HolaMundo** con todos los componentes del proyecto, Figura 7.34

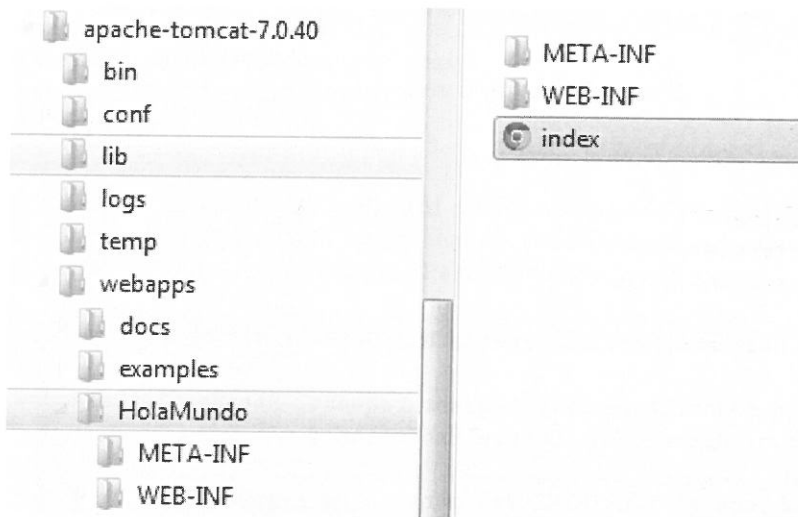


Figura 7.34: Proyecto desplegado

Creación de un *servlet* *HolaMundo*.

- 2.1. En *Eclipse*, en el mismo proyecto, colócate sobre el directorio **Java Resources\src** y añade un nuevo *servlet* llamado *HolaMundoServlet*, Figura 7.35. No crearemos ningún paquete, simplemente se creará en el paquete por defecto.

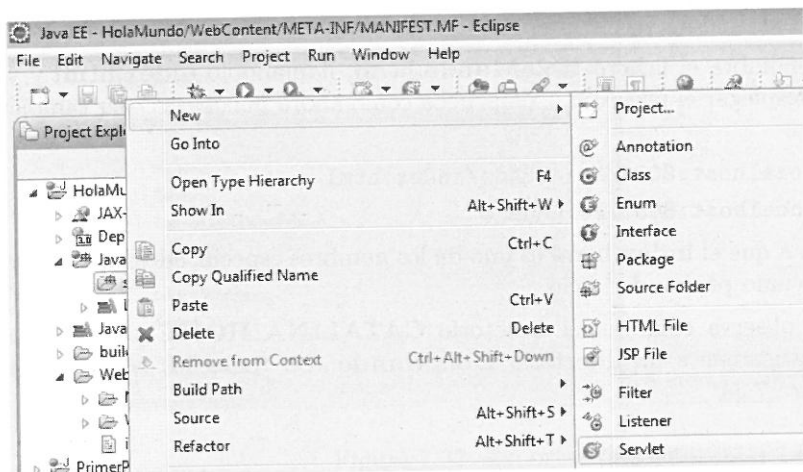


Figura 7.35: Nuevo *servlet*

Utiliza el siguiente código para el *servlet*:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HolaMundoServlet extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {

        // Establece el tipo MIME del mensaje de respuesta
        response.setContentType("text/html");
        // Crea un flujo de salida para escribir la respuesta a la petición del cliente
        PrintWriter out = response.getWriter();

        // Escribe el mensaje de respuesta en una página html
        try {
            out.println("<html>");
            out.println("<head><title>Hola Mundo</title></head>");
            out.println("<body>");
            out.println("<h1>Hola Mundo</h1>"); // dice hola
            // Muestra información de la petición del cliente
            out.println("<p>Request URI: " + request.getRequestURI() + "</p>");
            out.println("<p>Protocolo: " + request.getProtocol() + "</p>");
            out.println("<p>Dirección remota: " + request.getRemoteAddr() + "</p>");
            // Genera un número aleatorio para cada petición
```

```

        out.println("<p>Número aleatorio: <strong>" + Math.random() + "</strong></p>");
        out.println("</body></html>");
    } finally {
        out.close(); // Cierra el flujo de salida
    }
}
}

```

- 2.2. Desde *Eclipse* vamos a editar el descriptor de despliegue, **web.xml**, de la aplicación introduciendo el siguiente contenido.

```

<!-- Indicamos que la clase HolaMundoServlet se corresponde
con el nombre de servlet HolaMundo -->
<servlet>
    <servlet-name>HolaMundo</servlet-name>
    <servlet-class>HolaMundoServlet</servlet-class>
</servlet>

<!-- Indicamos que el nombre de servlet HolaMundo corresponde a la url /DiHola -->
<servlet-mapping>
    <servlet-name>HolaMundo</servlet-name>
    <url-pattern>/DiHola</url-pattern>
</servlet-mapping>

```

Para editar el descriptor de despliegue **web.xml** seleccionaremos la pestaña **Source**, Figura 7.36. Ten en cuenta que para cada *servlet* tendrás que indicar un elemento *servlet* y un elemento *servlet-mapping* que compartan el *servlet-name*.

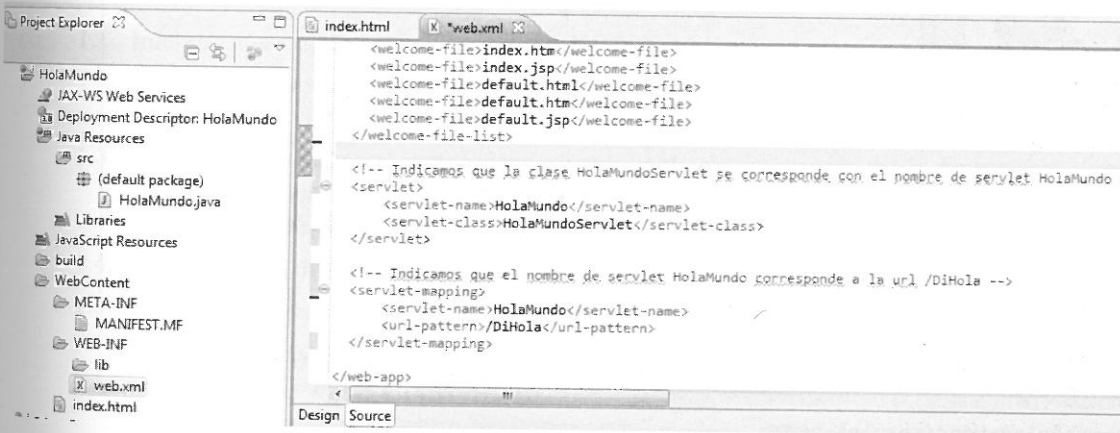


Figura 7.36: Editar descriptor de despliegue **web.xml**

- 2.3. Para todas las prácticas, es aconsejable copiar el código del fichero **pdf** a un editor de texto plano, como el bloc de notas, y de aquí a *Eclipse*. Así evitaremos errores debidos a caracteres presentes en el **pdf** que pueden provocar errores.
- 2.4. Para que el código copiado y pegado aparezca bien formateado, utiliza la opción **Format** del menú **Source** de *Eclipse*.

- 2.5. A continuación exportamos el proyecto a un archivo **.war** y lo desplegamos en el servidor.
- 2.6. Para acceder al *Servlet* utilizaremos la dirección `http://localhost:8080/HolaMundo/DiHola` y veremos el resultado mostrado en la Figura 7.37.

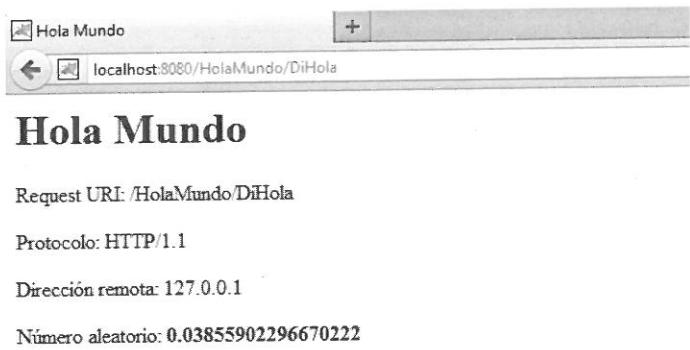


Figura 7.37: *Servlet* HolaMundo

- 2.7. En la siguiente figura podemos ver el resumen de funcionamiento desde la petición por parte del cliente a la ejecución del *servlet*, Figura 7.38.

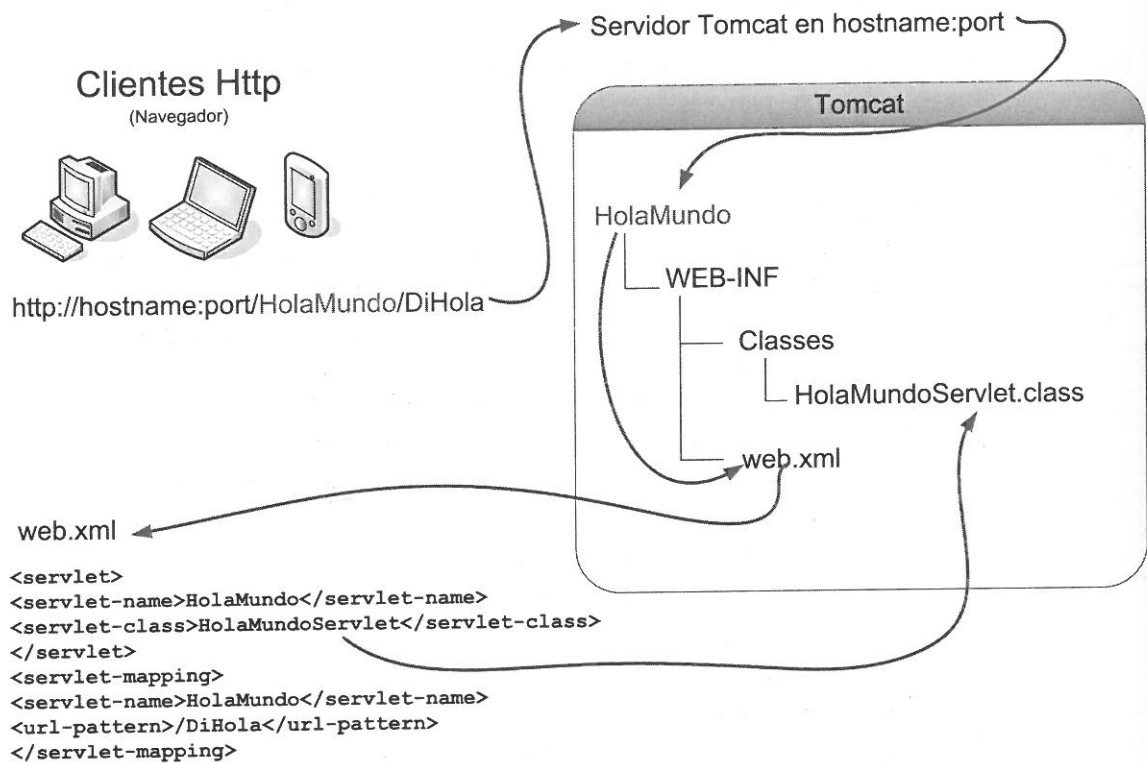


Figura 7.38: *Servlet* HolaMundo

- 2.8. Para finalizar comprueba cómo en la carpeta: **CATALINA_HOME\webapps\HolaMundo\WEB-INF\classes** aparece el fichero **HolaMundoServlet.class** correspondiente al fichero **HolaMundoServlet.java** que creamos en *Eclipse*, Figura 7.39.

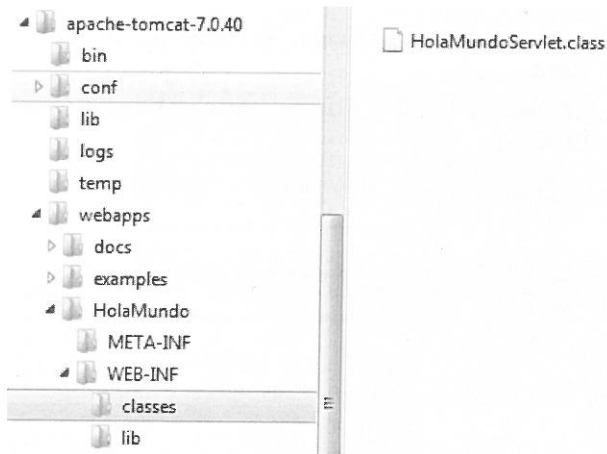


Figura 7.39: Servlet HolaMundo