

Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla



Prácticas de Laboratorio

Programación Distribuida Aplicada

Primavera 2025

Dra. Carmen Cerón Garnica

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS. DE LA COMPUTACIÓN

LABORATORIO

Programación Distribuida Aplicada

Práctica No.	UNIDAD 1.	Introducción a los Sistemas Distribuidos.	Duración
0	Nombre de la Práctica	Configuración de Servidores	120 minutos
Objetivo de la Práctica		Configurar el entorno de desarrollo y un servidor para el despliegue de aplicaciones web.	

1. INTRODUCCIÓN

1. Funcionamiento de un servidor web

Para el funcionamiento correcto de un servidor web necesitamos un cliente web que realice una petición http o https a través de un navegador como Chrome, Firefox o Safari y un servidor donde esté almacenada la información.

El proceso sería el siguiente:

- Tras la primera consulta por parte del usuario hacia una web, se establece una conexión entre el servidor DNS y el ordenador que realiza la consulta o petición.
- Este servidor DNS responde con la dirección IP correcta del servidor web donde está alojado el contenido solicitado.

Existen diferencias entre un servidor web y un hosting web.

¿Qué es un servidor web?

Un servidor web es un tipo de software que realiza ciertas tareas en el nombre de los usuarios. En términos simples es un ordenador que se encarga de servir o enviar información a otros elementos que están conectados a él. En él se alojan contenidos que se distribuyen de diversas formas, siendo la más común para todos la página web. Aquí está la clave, dentro de un mismo servidor se pueden alojar diferentes hostings, pero no al revés.

¿Qué es un hosting?

Un servicio de hosting o alojamiento web es un espacio donde se encuentra alojado un sitio web, para que pueda ser visible y accesible al mundo. Este servicio y espacio lo proporcionan los servidores. Para que tú tengas tu hosting dentro albergando tu web.

Existen diferentes tipos de hosting:

- a) Hosting gratuito: es el servicio de alojamiento web gratuito, pero que posee unos recursos muy limitados. Suele financiarse a través de la publicidad y tienen pocos servicios.
- b) Hosting compartido: aquí se alojan varias webs o plataformas y están dentro del mismo servidor, lo que favorece a la hora de compartir costes y obtener más recursos que el anterior. Es usado para web sencilla que no poseen mucho tráfico y aún están en periodo de crecimiento. La ventaja es que se reparte costes y se obtienen más servicios, las desventajas que las penalizaciones a cualquier otra web que esté en el hosting también afectara a la tuya.
- c) Hosting dedicado: está dedicado exclusivamente a una cuenta de cliente con todos los servicios disponibles.
- d) Servidor virtual o VPS: se hace una compartición en varios servidores y se usa como si no fuera compartido, aunque si lo es en realidad.
- e) Cloud hosting: En él se combinan los recursos de una multitud de servidores, pero que actúa como uno solo. Este tipo de alojamiento web posee más ventajas, pues es más flexible pues permite ajustar las dinámicas, los tiempos y las funcionalidades.

¿Cuál es la diferencia entre servidor web y hosting?

Los servidores permiten alojar contenidos que luego son distribuidos de múltiples maneras, la más común hoy día son las páginas web, si bien que los contenidos distribuidos mediante Apps cada vez cobran más protagonismo.

Las diferencias que hay entre servidor web y hosting son muchas porque cada uno tiene una función diferente:

- a) Mientras que un servidor web es una máquina física o virtual que se encuentra conectada a la red de internet y ofrece una serie de recursos, el hosting es solo un espacio para almacenar que utiliza los recursos del servidor. Esto significa que un servidor cuenta con más recursos que solo el hosting.
- b) Un hosting puede compartir con otros la misma dirección IP dentro del servidor, ya que dentro de un mismo servidor puede haber varios hostings, pero no al revés.
- c) Muchos usuarios y empresas aún confunden el hosting con el servidor web, algo que a priori puede ser muy normal, pero que puede traer muchos problemas cuando se trata de adquirir unos servicios u otros.

Tipos de servidores y hosting web

Los servidores web y los Hosting web son dos conceptos diferentes que no deben confundirte, por un lado están los Hosting VPS, Cloud o Compartidos, por citar los más conocidos, pero existen otros servicios asociados a los servidores web

Resumen:

Tipos de servidor web:	Tipos de hosting web:
<ul style="list-style-type: none">• Servidores Proxy.• Servidores de correo.• Servidores DHCP.• Servidores de DNS.• Servidores de aplicaciones.• Servidores de comunicación en tiempo real.• Servidores de archivos.• Servidores de impresión.• Servidores de base de datos.• Otro	<ul style="list-style-type: none">• Hosting gratuito (asumes publicidad y recursos muy limitados).• Hosting compartido (el más común en los alojamientos web).• Hosting dedicado.• Hosting Reseller (No llega a ser un VPS pero disfruta de particionado propio).• Se podría considerar el VPS como un servidor virtual privado pero es importante saber que al final se puede tratar de varios VPS alojados en un servidor físico particionado, motivo por el que se suele considerar un tipo de Hosting y no de Servidor.

Se debe configurar los entornos de Netbeans y Eclipse

2. DESARROLLO DE LA PRÁCTICA

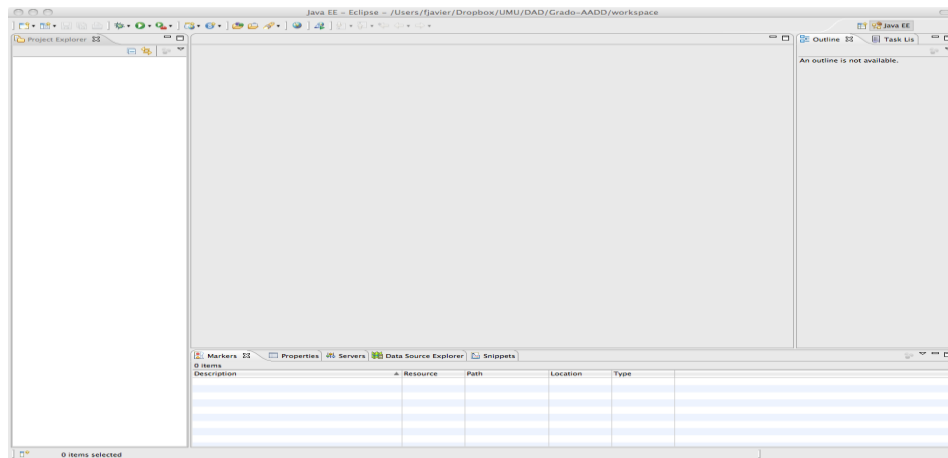
Servidor Web

1. Entorno de Eclipse

En las prácticas de laboratorio el alumno dispondrá de la herramienta Eclipse para el desarrollo de sus proyectos. La distribución de Eclipse incorpora el plugin JDT para la edición de código en Java, así como otros plugins para asistir en el desarrollo de aplicaciones enterprise.

En esta práctica el alumno se familiarizará con el entorno de desarrollo y aprenderá a configurar un servidor para el despliegue de aplicaciones web. Creará un sitio web como ejemplo sobre el servidor configurado.

En la siguiente imagen se puede observar el entorno de desarrollo Eclipse bajo la perspectiva JavaEE.

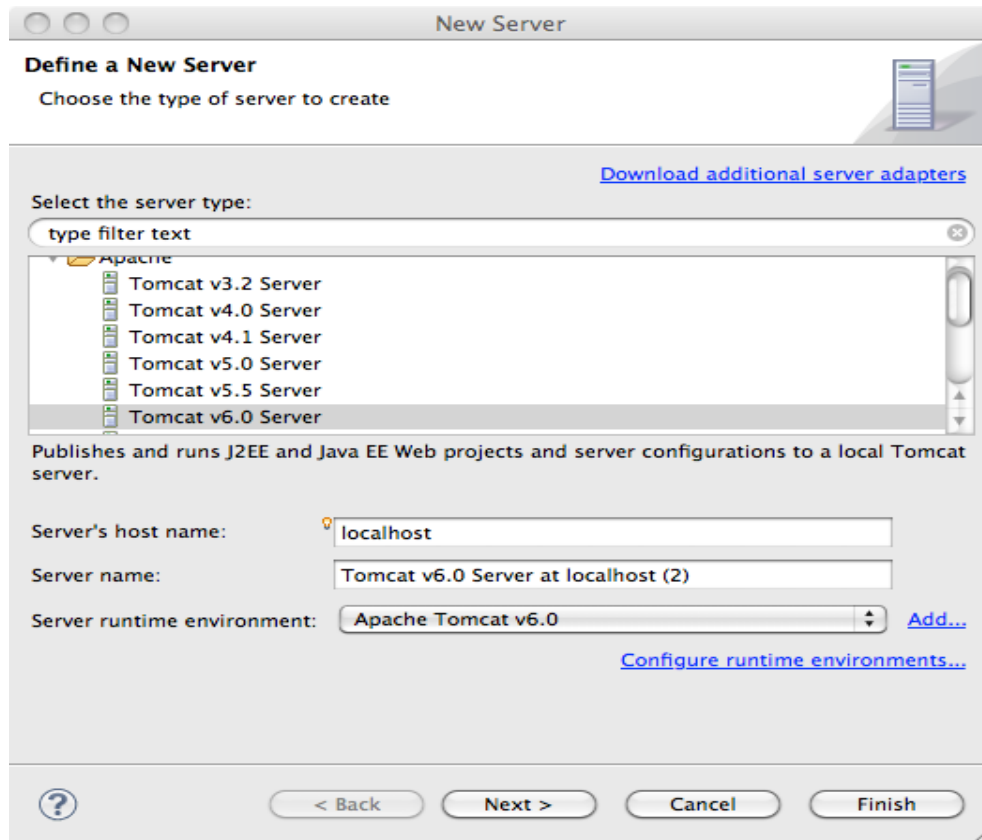


En el área izquierda se puede ver el explorador de proyectos desde el cual se podrán gestionar los proyectos y los recursos contenidos en los mismos. En el área inferior se pueden ver varias pestañas para gestionar la configuración de servidores (servers), la lista de errores (markers), la salida del servidor (console) o la página de propiedades (properties).

Para la configuración de un servidor, accederemos a la pestaña servers



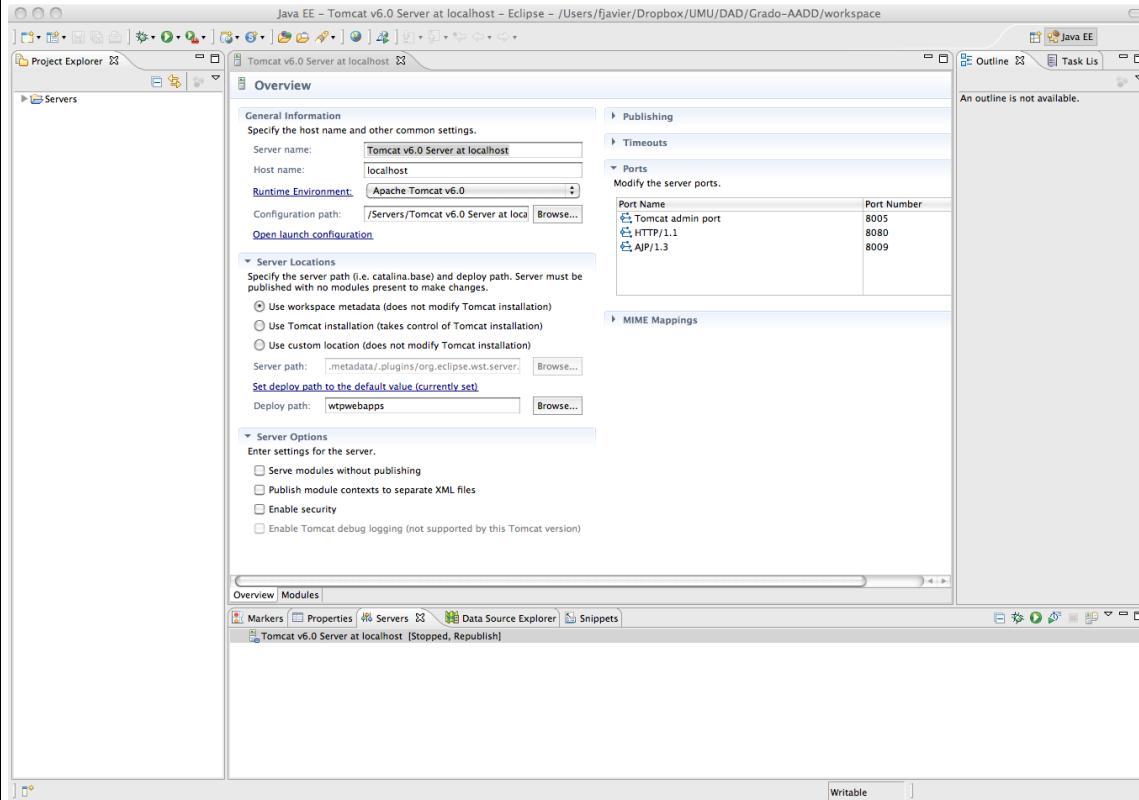
Con el botón derecho sobre la pestaña servers, seleccionamos **new→server**. Se abrirá una nueva ventana que nos asistirá en la configuración de un servidor. Se procederá a configurar un servidor Apache Tomcat 6



Tras seleccionar **Next** se abrirá una nueva pantalla para la configuración del directorio instalación de Tomcat y la distribución de JRE (será necesario usar una distribución JDK para futuras prácticas)

1. Configuración de un servidor de aplicaciones (Tomcat)

Una vez configurado el servidor, hacemos doble click para abrir la interfaz de configuración:



Cabe resaltar las siguientes secciones:

- *Server Locations*

Permite indicar si el despliegue de las aplicaciones sobre el servidor se hará en la misma instalación del servidor (directorio del servidor) o bien sobre el workspace de Eclipse.

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)
☐ Use Tomcat installation (takes control of Tomcat installation)
☐ Use custom location (does not modify Tomcat installation)

Server path:

[Set deploy path to the default value \(currently set\)](#)

Deploy path:

General Information

Se puede editar la ubicación de la distribución del servidor, y se puede acceder a la configuración del entorno de ejecución y del proceso de ejecución.

General Information

Specify the host name and other common settings.

Server name:

Host name:

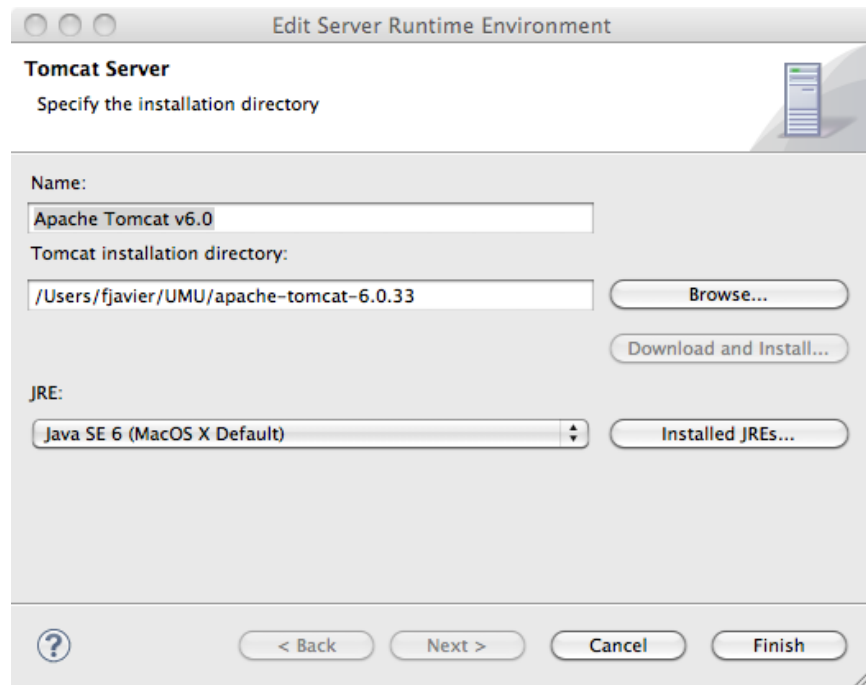
[Runtime Environment:](#)

Configuration path:

[Open launch configuration](#)

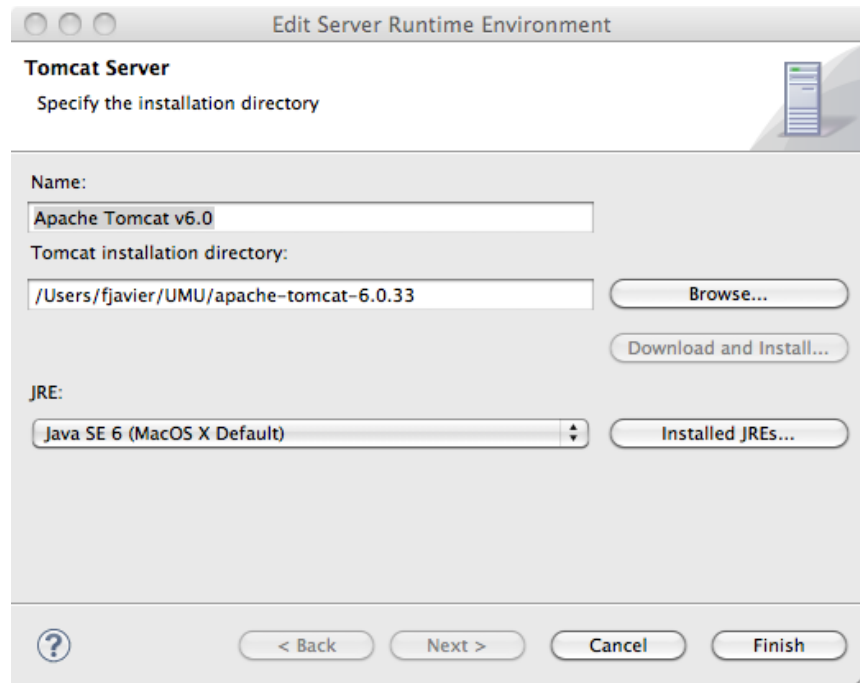
Runtime Environment

Permite modificar el directorio de instalación y seleccionar la instalación de la maquina virtual a usar para el servidor.



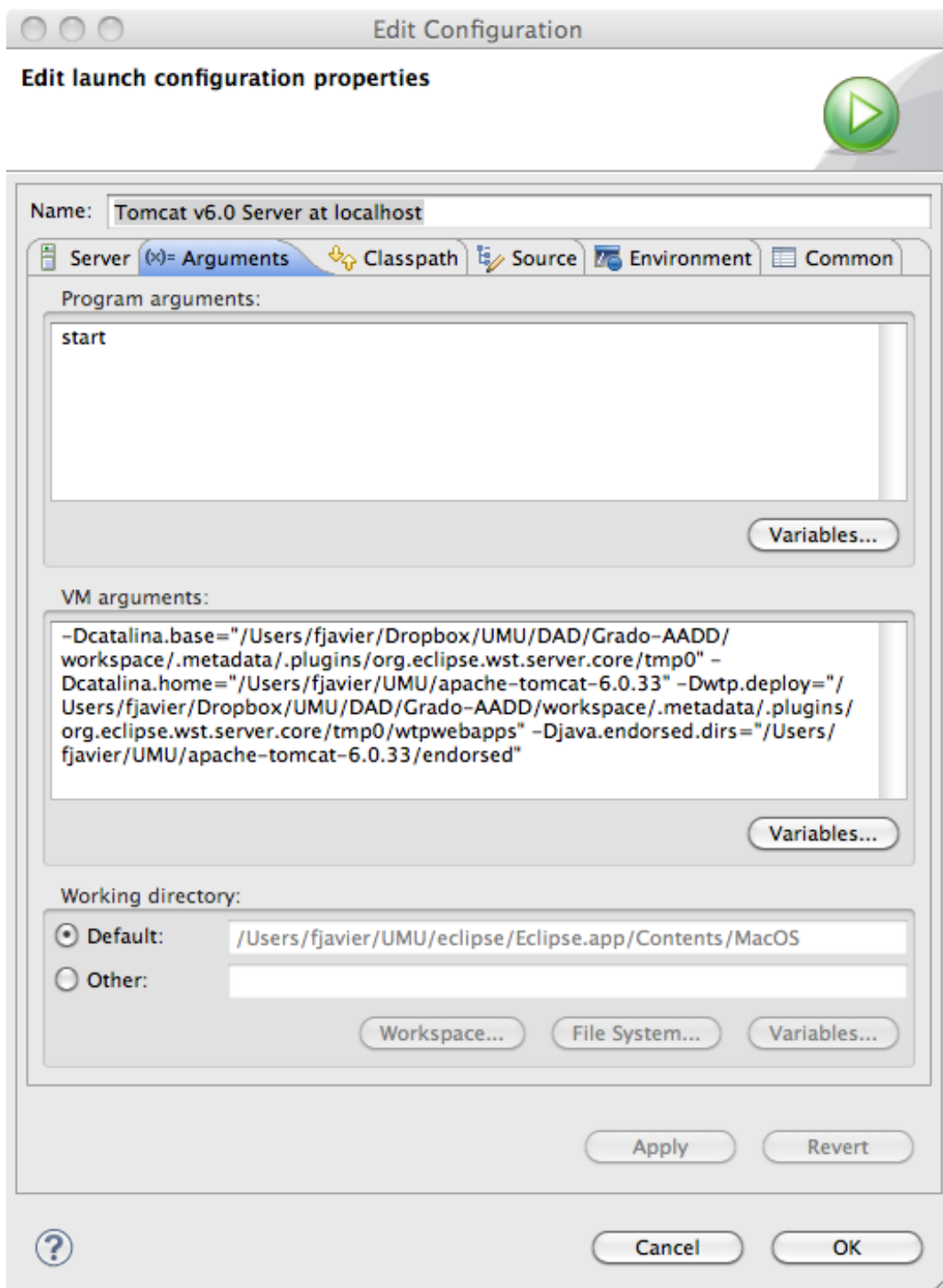
Runtime Environment

Permite modificar el directorio de instalación y seleccionar la instalación de la máquina virtual a usar para el servidor.



Open Launch Configuration

Permite configurar los argumentos de la maquina virtual sobre la que se ejecuta el servidor, el classpath de la ejecución del servidor, y las variables de entorno.



2. Creación de un sitio web en Tomcat

Se va a crear un sitio web con contenido HTML y JavaScript.

Un formulario HTML es un área de una página web donde el usuario puede introducir información. Típicamente esta información es enviada para su procesamiento en un servidor. Un formulario se define con las etiquetas HTML:

```
<form method="POST" name="formulario" action="/servlet/MiServlet">
```

El primer atributo de la etiqueta es el método. Tenemos dos opciones, POST que enviará los parámetros dentro del contenido del mensaje HTTP, o GET, que los establecerá en la URL. El siguiente atributo es el nombre del formulario que utilizaremos desde JavaScript para poder acceder a sus campos. Por último, el atributo action especifica el recurso que procesará la petición.

Los formularios pueden contener distintos tipos de campos. Todos ellos tienen como propiedad el nombre, que será el nombre del parámetro que recibirá el recurso de procesamiento y un valor asociado al nombre. A continuación describiremos brevemente los tipos y propiedades de los campos de un formulario:

- Cuadro de texto. Podemos especificar un valor inicial y disponemos de una opción para editar contraseñas (oculta el valor con asteriscos).
- Área de texto (area). Al igual que el anterior, podemos indicar el valor inicial, el ancho en caracteres y número de líneas.
- Casilla de verificación (checkbox). Podrá estar “activada” o “desactivada”. Si se queda activada, se enviará el valor (el que especifiquemos) al servlet.
- Botón de opción (radio). Se definen en grupo. El nombre del botón está asociado al grupo y el valor representa la información que se envía cuando es marcada esa opción. Sólo podrá estar marcado un botón de un mismo grupo.
- Menú desplegable (select): tiene un conjunto de pares opción-valor, siendo opción la etiqueta que se muestra en el formulario y valor su información asociada. Podemos permitir selección única o múltiple. Si elegimos selección única, tendrá la apariencia de una lista desplegable, si hemos editado el menú con una sola línea de altura. Si queremos una representación en forma de lista daremos un alto mayor que uno al menú.
- Botones (button). Tiene como propiedad el tipo, que determina la acción que realizará el botón: submit enviará los datos del formulario, reset restablecerá los valores (dejará vacío el formulario) y un botón normal (button). Si tenemos varios botones submit, sólo se enviará el valor del botón pulsado.

Utilizaremos Eclipse para crear los formularios (html) y el sitio web.

Creemos un nuevo proyecto con la opción: File / New/ Dynamic Web Project

Aparecerá un asistente para la creación del proyecto que solicitará el nombre del proyecto, la carpeta de ubicación del proyecto, el servidor destino donde será desplegado y ejecutado, y la versión de modulo web empleada. Otra información que podremos introducir hace referencia a las facetas del servidor, o el proyecto EAR donde será contenido (estas dos últimas cuestiones no será útiles en este práctica).

Haciendo click en next, el asistente irá mostrando distinta información sobre la configuración/gestión de los recursos incluidos en el proyecto (ir explorando la información del asistente y dejar los valores por defecto: carpeta con los fuentes de las clases, carpeta donde se generarán los compilados, carpeta con ubicación de los recursos públicos del sitio web y donde se estructurará el contenido de una aplicación web).

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:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

A good starting point for working with Apache Tomcat v6.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

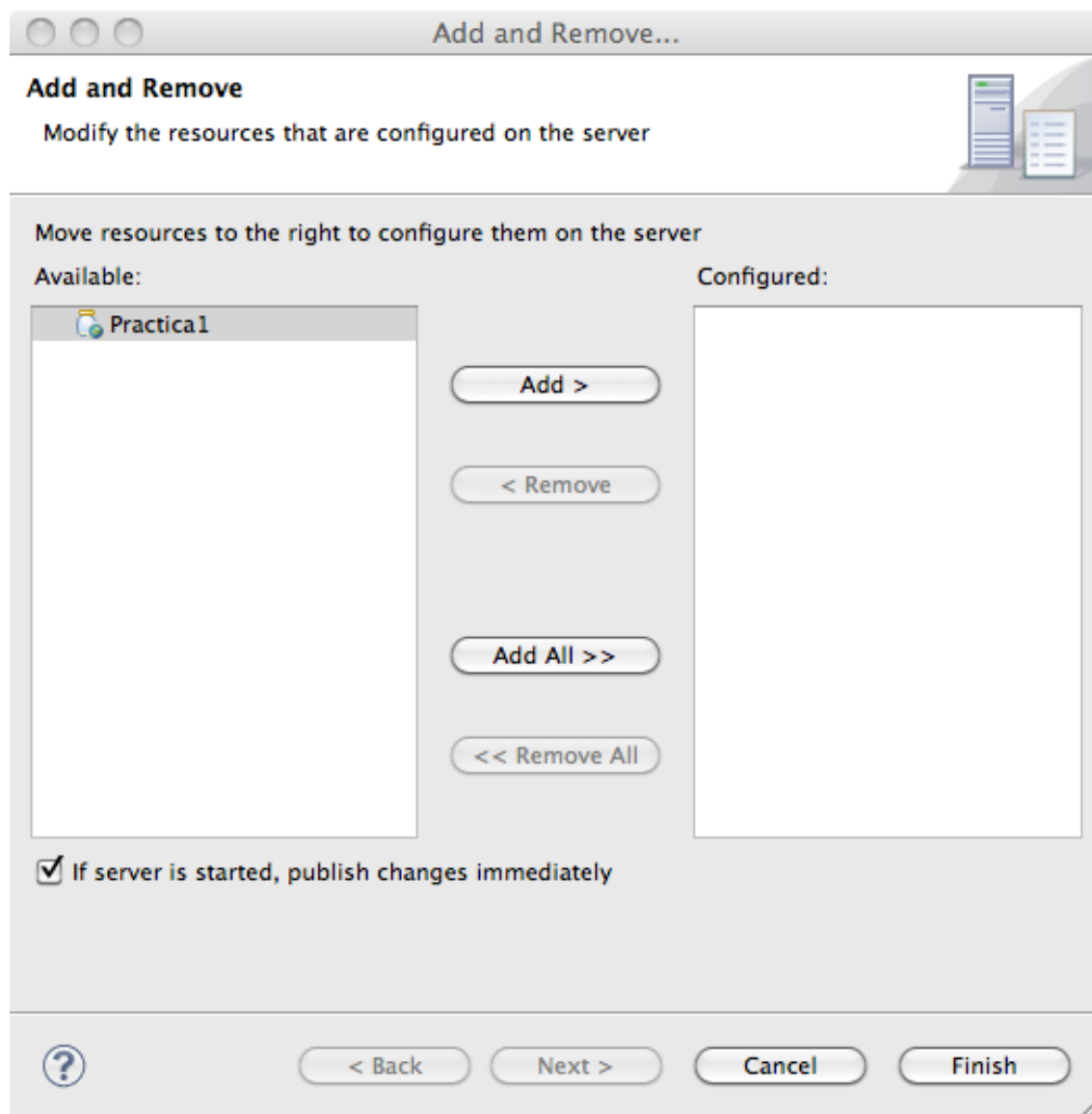
EAR project name:

Working sets

☐ Add project to working sets

Working sets:

Una vez creado el proyecto web, es necesario añadirlo al servidor Tomcat (boton derecho sobre el servidor y seleccionar Add and remove...) tal y como se muestra en la siguiente imagen:



- Consultamos ahora lo que se ha creado en el explorador de proyectos:
- Consultamos ahora lo que se ha creado en el explorador de proyectos:
- Añadimos el index.html a la parte pública de la aplicación web (WebContent):
- Una vez creado el index.html, lo seleccionamos con el botón derecho y elegimos:
Open with / Web Page Editor
- Se abrirá el editor de páginas web, tal y como se muestra en la imagen. Dicho editor

permite editar el contenido del recurso html mediante un editor de texto (asistido) y, además, un editor gráfico (que se apoyará la paleta de HTML que encontramos en la parte superior derecha del área de edición del recurso. Este editor gráfico permitirá, mediante drag&drop, diseñar las interfaces html.

3. Ejercicio: Probar a desplegar los compontes de la paleta html sobre el fichero index.html.

Ejercicio: Diseña un formulario para solicitar los datos de registro a un usuario: nombre y apellidos, NIF, correo, identificador y clave por duplicado. Añade al formulario un pequeño cuestionario con una pregunta con varias opciones y sólo una posible respuesta (usar botones de opción), otra con opciones múltiples (con casillas de verificación) y un campo que dé el consentimiento del usuario a enviarle información por e-mail.

JavaScript

JavaScript es un lenguaje de script que permite realizar procesamiento en el lado del cliente. El código JavaScript está incrustado en la página HTML. Utilizando este lenguaje podemos acceder a la estructura del documento web, lo que es especialmente útil para validar los campos de un formulario, por ejemplo. Su sintaxis es bastante parecida a Java. El código JavaScript es procesado por el navegador web. A continuación explicaremos brevemente las características más interesantes de este lenguaje.

Introducción de código JavaScript dentro de una página web.

```
<SCRIPT LANGUAGE="JavaScript"> ... </SCRIPT>
```

Escribir en el documento web.

```
document.writeln("<H1>Hola Mundo</H1>");
```

Pedir información al usuario.

```
window.prompt("Introduce un valor", "0");
```

Mostrar una información de alerta.

```
window.alert("Cuidado \n No has indicado tu DNI");
```

Declaración de variables.

JavaScript es un lenguaje tipado dinámicamente con los tipos básicos de cualquier lenguaje de programación (enteros, reales, booleanos, etc.). Las variables se declaran del siguiente modo:

```
var nombre, edad;
```

Estructuras de control. Las estructuras de control (condicionales, bucles) son iguales que en Java.

Operadores aritméticos, lógicos y relacionales. Iguales que en Java. Arrays. Definición de un array:

```
var n1 = new Array(10); var n2 = [1, 2, 3];
```

```
var n3 = new Array();
```

Tamaño: n1.length;

Definición de funciones.

```
function cuadrado(x) { return x * x;
```

```
}
```

Acceso a un campo del formulario.

`edad = document.miFormulario.edad.value; document.miFormulario.nombre.value = "Juan";`

Otro modo: `document.forms["miFormulario"].elements["edad"].value = "Juan".`

Captura de eventos.

Podemos ejecutar funciones JavaScript como respuesta a eventos producidos en la página web, especialmente en los formularios. Desde nuestro punto de vista, los eventos más importantes son el evento ONCLICK, de los botones, y el evento ONCHANGE del resto de campos del formulario.

```
<input type="submit" name="boton1" value="ok" onClick="return validarDatos()">
```

Habitualmente se suele programar un solo evento ONCLICK del botón

Objetos.

Con JavaScript podemos trabajar con objetos. Dentro del lenguaje existen algunos objetos predefinidos, como el objeto Math que contiene rutinas matemáticas y los métodos de las cadenas.

Math:

Funciones `sqrt`, `abs`, `min`, `max`, `round`, `random`, etc. `m1 = Math.abs(m2);`

String:

`length` (es un atributo), `charAt`, `indexOf`, `lastIndexOf`, `substring`, `toLowerCase`, `anchor(etiqueta)`, `bold`, `link(url)`, etc.

`caracter = nombre.charAt(3); if (nombre.length >= 0) ...`

Los caracteres se expresan con dobles comillas, igual que las cadenas `if (nombre.charAt(3) == "a") ...`

Incluir el script desde un fichero.

```
<script language="javascript" src="FormCheck.js"></script>
```

Otros.

Convertir una cadena en entero:

`numero = parseInt(cadena);`

Podemos comprobar que la conversión ha sido correcta utilizando la función

`isNaN`: `if (isNaN(numero)) ...`

Los switch pueden tomar cadenas como valores de las etiquetas `case`.

3. EVIDENCIA DE APRENDIZAJE

Entregar el **reporte de la práctica**

Fecha de Entrega: Laboratorio de clase

ACTIVIDAD COLABORATIVA

- Realiza en equipo los siguientes ejercicios
 1. **Ejercicio:** Utiliza JavaScript para validar los datos del formulario del ejercicio anterior antes de su envío. Ningún campo puede estar vacío, el NIF debe tener como máximo 8 dígitos seguidos de una letra, la dirección de correo debe ser válida (el servidor debe pertenecer a algún dominio), las dos claves de usuario deben coincidir y la pregunta con opciones debe tener al menos una elección
 2. **Ejercicio:** Realizar un formulario de la captura de inventario de productos usando JavaScript