

Prácticas de Laboratorio**BLOQUE Nº 2. Desarrollo de servicios Web****I. Desarrollo de servicios web básicos en Java.**

1. Codificar un pequeño cliente HTTP que abra una conexión con el servidor <http://www.lcc.uma.es>, inicie la descarga del fichero cuyo nombre se le pasa como parámetro, lo almacene localmente y muestre el resultado de llamar a los siguientes métodos:

- `getLength()`
- `getContentType()`
- `getDate()`
- `getLastModified()`

2. Desarrollo de un servidor web simple en Java de acuerdo a las siguientes especificaciones:

- Debe ser un servidor concurrente TCP y debe esperar conexiones en el puerto 8000
- Solamente soporta la gestión de peticiones GET
 - Ej: GET objeto HTTP/1.1
- No soporta cookies ni sesiones
- Solamente soporta las respuestas:
 - HTTP/1.0 200 OK
 - `<html><body><h1>404 Not Found</h1></body></html>`
 - HTTP/1.0 404 Not Found
 - HTTP/1.0 404 Permission Denied
- Los tipos MIME que soporta son:
 - `text/plain` (correspondiente a ficheros `.txt`)
 - `text/html` (correspondiente a ficheros `.htm` y `.html`)
- Debe enviar las siguientes cabeceras:
 - `Content-Type`
 - Ej: `Content-Type: text/html`
 - `Content-Length`
 - Ej: `Content-Length: 3453`

II. Desarrollo de servicios web con servlets

1. Probar los *servlets* vistos en clase en las diferentes transparencias.

2. Copiar y editar el ejemplo con *Login* visto en clase, cuyo código se incluye en el campus virtual y ejecutarlo, tanto la implementación con *servlets* como la implementación con páginas JSP. Este ejercicio pretende que te sirva de base para realizar el ejercicio obligatorio.

- Diseña tu propia solución aplicando el patrón de diseño MVC y utilizando servlets, beans y páginas JSP para implementarlo
- Agrégale la autenticación con password
- Busca la cookie que genera esta aplicación con la sesión que has creado

3. Se pretende desarrollar un **sistema tutor de la asignatura**, cuyo objetivo es evaluar el conocimiento de los alumnos de cara al examen. El sistema tutor consiste en la presentación de un test consistente en varias preguntas de múltiple respuesta. Una vez finalizado el test, se le informará al alumno de su puntuación global. Utilizar la tecnología de *servlet* (usar el concepto de sesión se proporciona código de sesión por si quieres utilizarlo) para desarrollar esta aplicación de acuerdo a la siguiente especificación de requisitos:

- La pantalla inicial mostrará el nombre de la aplicación y un formulario donde se podrá introducir el nombre de usuario y la contraseña (no pueden ser nulos), así como un enlace a la función de registro de nuevo usuario. Para introducir la contraseña se deberá usar un campo en el formulario HTML como el siguiente:

```
<input type="password" name="password" />
```

- Los usuarios registrados se almacenarán en un fichero de texto en el que cada línea contendrá el nombre de usuario y su contraseña tal y como se indica en el ejemplo:

usuariol	password1
usuario2	password2
usuario3	password3

- La pantalla de registro mostrará también un formulario para introducir el nombre de usuario y la contraseña que desee el usuario. Cuando el usuario rellene estos datos y envíe el formulario, se almacenará en el fichero de usuarios registrados. Tener en cuenta, que ha de comprobarse que no se introduce un nombre de usuario o contraseña nulos y que el nombre de usuario introducido no existe con anterioridad en el sistema. Si se ha registrado correctamente, se le re-direccionará a la pantalla de inicio para que pueda iniciar sesión.
- Después de autenticarse el usuario verá la página principal donde se mostrará el nombre y una breve descripción sistema tutor (ej: "Este sistema tutor evalúa los conocimientos de la materia XXXX) y aparecerá un enlace al sistema tutor y otro para desconectarse.
- Una vez que el usuario ha iniciado la sesión correctamente se le mostrará la pantalla inicial del *Sistema Tutor*. En la parte superior de la página se pondrá un enlace que le permita al usuario cerrar la sesión y el nombre del usuario. A continuación, se mostrará una lista de los test existentes (tres al menos) para que el usuario pueda seleccionar uno. Cada test irá acompañado de una breve descripción de su dificultad.
- El usuario podrá realizar por lo menos un test. El sistema le irá mostrando cada pregunta en una página diferente, o cada grupo de preguntas en páginas diferentes. Deben mostrarse al menos cinco preguntas o grupos de preguntas.
- Todas las respuestas son del tipo RADIO con las posibles contestaciones Si, No, Algunas veces.

- Se inicializará en memoria una matriz (o tabla Hash) con las preguntas a realizar y las respuestas correctas para cada una de ellas.
- El procesamiento principal consistirá en generar dinámicamente las páginas con las preguntas, leer y comprobar la respuesta del alumno y actualizar alguna variable de sesión que cuenta el número de fallos para ese alumno. Tras comprobar si el alumno ha acertado o fallado la respuesta, se mostrará una página de respuesta felicitando o lamentando el fallo junto con la respuesta correcta.
- Al finalizar el test se le mostrará al alumno una página de resultado donde se le indicará el número de fallos y el porcentaje de aciertos. Si este porcentaje fuese menor o igual al 33 % se le aconsejará no presentarse al examen, si está entre el 33 y 66% se le aconsejará estudiar más, y si el porcentaje supera el 66 % se le felicitará deseándole buena suerte. En esta página se le dará la opción de desconectarse o de ir a la página inicial y elegir otro test.
- En cada página el alumno tendrá la opción de cancelar que lo llevará a la página final.
- Incluir dentro de cada página información sobre la sesión como el nombre del usuario conectado.

Las páginas web no pueden contener definición de estilos de presentación, todas deben realizarse dentro de las hojas de estilo.

- Definir un estilo para cada texto (tipo de letra, color, tamaño, etc.), para los mensajes de error (ídem anterior), fondo (definir un color o una imagen) y otros elementos que tu consideres.
- Definir las características generales de los enlaces (ej: tipo letra, color, etc.):
 - `a { } /*Para características generales de los enlaces*/`
 - `a:link { } /*para el enlace, tal cual*/`
 - `a:visited { } /*Para cuando ha sido visitado*/`
 - `a:hover { } /*Cuando el ratón está sobre él*/`
 - `a:active { } /*Cuando está siendo pulsado*/`
- Agregar un pie de página con los datos del autor del servicio web, definiendo un estilo “autor” común a todas las páginas
 - Ejemplo:

Juan Pérez (Universidad de Málaga)
Copyright©2013 Todos los derechos reservados.

<http://www.juanperez.es/>

- La validación de campo obligatorio login y password realízala con javascript.
- En la página de registro, debes contemplar el doble campo password y comprobar que ambas passwords son iguales. coloca también al menos cuatro campos obligatorios más que requieran validación con javascript. Por ejemplo, dirección de email/ pregunta/ código de seguridad para recuperar una cuenta, fecha de nacimiento, tarjeta de pago por el servicio, etc.
- Las validaciones de campos de formularios deben realizarse en código Javascript en cada una de las páginas de la práctica..
- Colocar una animación con imágenes
- En la página principal pon la fecha de última modificación.

Todo el código Java script debe estar en un fichero y las funciones necesarias se llaman.

4. Ejercicio 3 ampliado. Se podrá mejorar y extender el funcionamiento básico del ejercicio 3 considerando una o varias de las sugerencias que se hacen a continuación.

- Considerar que las respuestas puedan ser de cualquier tipo, y no de un tipo fijo.
- Almacenar el resultado de todos los usuarios (como parte del fichero de usuarios registrados o creando otro fichero) y dar la opción de ver una lista de los cinco alumnos que obtuvieron mejor puntuación hasta el momento en un determinado test.
- Agregar la opción de consultar los test realizados así como su nota, si el alumno proporciona un nombre de usuario más *password* ya registrados.
- Cualquier otra mejora que se considere apropiada

5. Hacer con javascript una aplicación web “Conversor de unidades de medida” de acuerdo a las siguientes especificaciones:

- Unidades de longitud: Convierte de metros a pies y pulgadas (y viceversa)
- Unidades de peso: Convierte de gramos a libras y onzas (y viceversa)

6. Probar el ejemplo con AJAX dado en clase

7. Modificar el servicio web propuesto como ejercicio obligatorio del bloque no. 2 de las prácticas de la asignatura con AJAX para el campo login/passw y/o para alguna de las otras páginas del servicio Web.

Obligatorios: II.3

Optativo: El resto

Fecha entrega: 21/01/21 (para los obligatorios)