

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:

- `getContentLength()`
- `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 y otras tecnologías.

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 (si decides hacerlo así para el ejercicio obligatorio).
- Agrégale la autenticación con password
- Busca la cookie que genera esta aplicación con la sesión que has creado

3. Se pretende desarrollar un **sistema de publicación de noticias con formato de blog**. Se entiende por blog, un servicio Web que permite almacenar un conjunto de entradas (que serán noticias en nuestro caso), las cuales se muestran ordenadas desde la más reciente a la más antigua. Asociada a cada entrada, los usuarios pueden añadir comentarios, que también se almacenarán y mostrarán ordenados por fecha. Se implementará el blog de noticias usando *servlets*, de acuerdo a la siguiente especificación de requisitos:

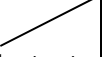
- La pantalla inicial *Inicio* mostrará el nombre de la aplicación y un formulario donde se podrá introducir el nombre de usuario y la contraseña (no pueden ser nulos), así como un enlace a la función de registro de nuevo usuario. Para introducir la contraseña se deberá usar un campo en el formulario 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. Una vez que el usuario ha pasado por la opción de registro, y tanto se haya registrado como no, se le redirigirá a la pantalla principal *Inicio*.
- La pantalla principal del blog de noticias *Blogs* incluirá un enlace a *Inicio*, otro a *Registro*, y el nombre del usuario actual. Si el usuario no se ha autenticado, entonces pondrá *anónimo*. Esta página mostrará la lista de entradas (noticias) y para cada una de ellas se visualizará la fecha, el título, el autor y los 100 primeros caracteres del texto de dicha entrada y un enlace para añadir un nuevo comentario (además de visualizar los ya existentes).
- La pantalla *Blogs* también incluirá un enlace a *Crear Entrada*. Si el usuario está autenticado se mostrará una pantalla mediante la cual introducirán los campos título y texto de una nueva noticia (*Título*, *Texto*). El autor de esa entrada se tomará directamente del usuario autenticado y la fecha se introducirá automáticamente.
- Una vez que tenemos la pantalla *Blogs* con las entradas, (su título, autor, texto y enlace), el usuario seguramente intentará abrir una entrada para visualizar el texto completo y añadir un comentario. En este momento habrá que comprobar que el usuario es un usuario válido, sino se le daría un mensaje de acceso no permitido y se

le redirige a la pantalla inicial *Inicio* (con enlace a *Registro*) por si quiere autenticarse o registrarse. En caso de que el usuario esté autenticado, entonces simplemente se abriría la entrada correspondiente mostrando el texto completo y los comentarios existentes y permitiendo añadir un nuevo comentario (con un campo de texto al final de esa página).

- El almacenamiento persistente de los blogs puede implementarse usando JDBC o de forma más sencilla mediante un conjunto de ficheros de texto. En el caso de usar un fichero de texto índice del blog, este tendría más o menos el siguiente formato:

Titulo1 Laura Casal Fecha texto1blalblalblalbal ficheroComentariosTitulo1.txt & Titulo2 Silvia Blanco Fecha texto2blalblalblalbal ficheroComentariosTitulo2.txt ...		%ficheroComentariosTitulo1.txt Juan Padilla Fecha comentarioComentarioComentario1 & Luis Torres Fecha comentarioComentarioComentario2 ...
--	---	---

donde el fichero de comentarios almacena los comentarios a dicha entrada con el autor y la fecha correspondiente (pueden modificar el formato del fichero para agregar nuevos delimitadores de campo).

- Probar la práctica por lo menos con 10 entradas cada una con 3 comentarios. Cada una de las noticias que se almacenen en el blog debe ocupar más de 30 caracteres.
- No usar hojas de estilo CSS ni etiquetas de formato, que no sean las básicas de HTML (ej: ``, ``, `<h1>`, ...), ya que esta tecnología se aplicará en la siguiente práctica.

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.

- Aplicar el MVC e implementar las vistas usando JSP en lugar de *servlets*.
- Incluir una sección para la búsqueda de entradas en el blog con el campo *Título* y *Autor*. Una vez que se introduce un criterio de búsqueda, se le devolverá la lista de entradas que cumplen dicho criterio incluyendo además de la fecha, el título, el autor y los 100 primeros caracteres del texto.
- Utilizar ficheros .html en vez de ficheros de texto para almacenar las entradas (noticias). En ese caso, cuando el usuario introduce una nueva noticia, en vez de escribir un texto, introduciría la noticia codificada en HTML y el tipo MIME sería text/html.
- En la página donde se muestra una entrada y la lista de comentarios, incluir para cada comentario un enlace a responder. En ese caso, los comentarios de respuesta se almacenarían asociados al comentario principal y se mostrarían justo debajo, indentados a la derecha.
- Cualquier otra mejora que se considere apropiada

Obligatorios: II.3

Optativo: I.2 y II.4 (en este caso, si se entrega en fecha, entregar solamente el ampliado e indicar en la memoria con qué características se ha ampliado)

Fecha entrega: 15/01/19 (para los obligatorios)