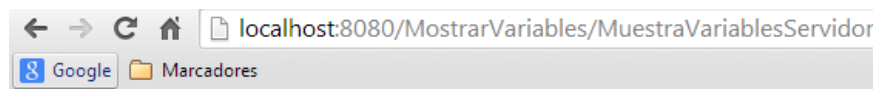


Ejercicios de desarrollo web con J2EE

1. Instalar y arrancar Tomcat. Arrancar el servicio. Acceder a la “home page” en localhost a través del puerto 8080.
2. Desde el “Home Page” de Tomcat, entrar en “First web application” → “Example app” y descargar la aplicación “sample.war”. Desplegarla (copiarla) en el directorio “webapps” de Tomcat y observar qué ocurre. Acceder a la aplicación a través de la URI “<http://localhost:8080/sample>”
3. Crear una aplicación J2EE (Dynamic web project) que muestre un “Hola mundo”. Observar dónde está desplegado. Crear un war y desplegarlo en el directorio webapps estándar de Tomcat.
4. Crear un servlet que muestre información de algunas de las variables manejadas por el servidor.



Variables de Servidor

Variable	Valor
AUTH_TYPE	No especificado
CONTENT_LENGTH	-1
CONTENT_TYPE	No especificado
DOCUMENT_ROOT	C:\xampp\tomcat\wtpwebapps\MostrarVariables\
PATH_INFO	No especificado
PATH_TRANSLATED	No especificado
QUERY_STRING	No especificado
REMOTE_ADDR	0:0:0:0:0:0:1
REMOTE_HOST	0:0:0:0:0:0:1
REMOTE_USER	No especificado
REQUEST_METHOD	GET
SCRIPT_NAME	/MuestraVariablesServidor
SERVER_NAME	localhost
SERVER_PORT	8080
SERVER_PROTOCOL	HTTP/1.1
SERVER_SOFTWARE	Apache Tomcat/7.0.42

5. Crear un servlet que muestre en su versión get, un formulario con este aspecto

Nombre

AFICIONES

Bailar ☐

Cantar ☐

Al enviar el formulario, el mismo servlet recibirá vía post los datos, pudiendo ocurrir una de estas tres cosas:

- Que no se envíe ningún nombre, a lo que se responderá “Debes introducir un nombre”
 - Que se envíe nombre, pero ninguna afición, a lo que se responderá “A <nombre> no le gusta nada”
 - Que se envíe nombre y alguna/s afición/es, a lo que se contestará “A <nombre> le gusta <afición1> <afición2> ...”
6. Realizar los mismos scriptlets que se realizaron en PHP: `generaRadio(...)`, `generaCheckBox(...)` y `generaSelect(...)`. NOTA: Utilizar `Map<String,String>` para suplir los arrays asociativos de PHP.
 7. Hacer un formulario con dos campos de texto nombre, clave y país (a elegir entre España, Italia, Francia y Portugal). Al enviar se comprobará si todo ha ido bien en cuyo caso se mostrarán los datos enviados. Si hubo algún error, según la lógica de negocio (nombre no vacío, clave entre 6 y 12 caracteres), se informará del error y se repintará el formulario con los datos anteriores.
 8. Hacer un formulario que seleccione un archivo local, y lo suba a una carpeta en el servidor (NOTA: La carpeta debe existir previamente). El formulario permitirá renombrar el archivo si se desea (si se deja el nombre en blanco, mantendrá el nombre anterior).

Selecciona el fichero

EclipsePHP.sh

Nuevo nombre

9. Hacer un programa que muestre un formulario nombre/clave. Cuando el usuario se valide con éxito (de entre dos usuarios al menos). El programa informará de hace cuántos segundos que no se validaba. A partir de un minuto, el programa no monitorizará este tiempo, y simplemente informará de que ha pasado más de un minuto. NOTA: Utilídense cookies. Puede ser útil el método `System.currentTimeMillis()`
10. Implementar, utilizando sesiones, un contador de visitas por usuario (para un par de usuarios distintos)

11. Para la tabla **libros(isbn, titulo, categoria)**, hacer un formulario de inserción y una página que muestre el catálogo. El primero ha de redirigir al segundo.
12. Realizar el ejercicio anterior, utilizando JSP.
13. Desdoblar a una clase externa “Libro” el objeto de negocio “Libro” y todas las operaciones de modelo relacionadas con libros. Implementar dichas operaciones con métodos estáticos.
14. Separar la clase Libro en dos clases: “Libro” y “LibroDAO”, dejando en ésta última las operaciones propias de modelo que teníamos en la otra.