

Ejercicios de introducción a los servlets

Índice

1 Servlet que muestra la fecha y hora actuales.....	2
2 Servlet que muestra parámetros de inicio.....	2
3 (*) Configurar logging en servlets.....	3

1. Servlet que muestra la fecha y hora actuales

Completar el servlet **ejercicios.FechaServlet** de la aplicación **appserv1** para que, tanto por GET como por POST, muestre una página HTML con la fecha y hora actuales en una cabecera <H3>, y en el <TITLE> de la página. Para ello podéis utilizar la clase `java.util.Date`, y sacar por la salida del servlet la hora en formato cadena:

```
public void doGet(...) throws ...
{
    String fecha = "" + new java.util.Date();
    response.setContentType(...);
    out = response.getWriter();
    ... // sacar la fecha tanto en el TITLE como en una cabecera H3
}
```

Una vez hecho, configurad el descriptor de la aplicación para que, además de poder llamar al servlet con su nombre por el alias `servlet`, lo podamos llamar con la URL:

```
http://localhost:8080/appserv1/fechaHora
```

2. Servlet que muestra parámetros de inicio

Implementar en el servlet **ejercicios.ParamIniServlet** de la aplicación **appserv1** un servlet que muestre en una tabla el nombre y el valor de todos los parámetros de inicio que se tengan configurados para ese servlet en el fichero descriptor (`web.xml`). La tabla tendrá dos columnas: una con el nombre del parámetro y otra con el valor.

Una vez hecho, probadlo añadiéndole en el fichero `web.xml` 3 parámetros de inicio con nombres `param1`, `param2` y `param3` y valores `val1`, `val2` y `val3`. Para ello deberéis dar un nombre al servlet (el nombre es arbitrario).

NOTA: para recorrer todos los parámetros de inicio, deberéis utilizar el método `init(...)` que tiene un parámetro `ServletConfig`, y utilizar dicho parámetro para obtener los nombres de los parámetros, y recorrerlos uno a uno para obtener su valor:

```
ServletConfig sc;

public void init(ServletConfig s) throws ServletException
{
    super.init(s);
    sc = s;
}

public void doGet(...) throws...
{
    java.util.Enumeration nombres = sc.getInitParameterNames();
    while (nombres.hasMoreElements())
```

```
    {  
        String nombre = (String)(nombres.nextElement());  
        String valor = sc.getInitParameter(nombre);  
        ... // Mostrar nombre y valor en una tabla  
    }  
}
```

3. (*) Configurar logging en servlets

En la aplicación **appserv1** tenemos dos servlets, **ServletLog4J1** y **ServletLog4J2** en el paquete **ejercicios**. Queremos configurar las librerías de logging para poder ver los mensajes que miten. Se pide:

- Comprobar que las librerías de logging de *commons-logging* y *log4j* están correctamente copiadas en `WebContent/WEB-INF/lib` (o en las librerías del proyecto)
- Añadir los ficheros de configuración pertinentes `commons-logging.properties` y `log4j.properties` en una carpeta de fuentes llamada `resources` para que saquen los mensajes de ambos servlets (de tipo INFO o superior) a un fichero `C:/errores.log`, con el formato:

dd/MM/aaaa hh:mm:ss - prioridad - texto del mensaje - salto de línea

- El servlet `ServletLog4J2` no saca mensajes de log. Añadid las líneas de código necesarias para que saque un mensaje de tipo INFO cuando empiece a procesar la petición (al inicio del `doGet`) y otro cuando la termine, anotando el tiempo transcurrido entre ambos mensajes (puede serte de utilidad el método `System.currentTimeMillis()` de Java).

Probad a llamar a los servlets **ServletLog4J1** o **ServletLog4J2** alguna vez, y que generen logs en el fichero `errores.log` que viene por defecto en el fichero de configuración.

