

# Ejercicios de procesamiento de peticiones

## Índice

|   |   |
|---|---|
| 1 Recogida de parámetros del usuario..... | 2 |
| 2 Trabajando con redirecciones.....       | 2 |
| 3 (*) Pruebas en servlets.....            | 2 |

## 1. Recogida de parámetros del usuario

La aplicación **appforms** contiene un formulario **form\_datos.html** con una serie de campos (tipo texto, listas, checkboxes...). Se pide que dicho formulario, en su acción, llame al servlet **ejercicios.DatosServlet** que deberéis implementar. El servlet recogerá todos los parámetros del formulario y los mostrará en una tabla de dos columnas (una con el nombre del parámetro y otra con el valor).

## 2. Trabajando con redirecciones

Sobre la aplicación anterior, tenemos otro formulario **form\_datos2.html** idéntico al del ejercicio anterior, que deberá llamar al servlet **ejercicios.DatosServlet2**. Hacer que el servlet **ejercicios.DatosServlet2** redirija a una página **index.html** (que deberéis implementar en el directorio `WebContent` de la aplicación) con un mensaje de bienvenida, si los datos introducidos en el formulario son correctos, y a la misma página **form\_datos2.html** si hay algún dato incorrecto. Entenderemos por dato incorrecto que alguno de los campos de texto se quede vacío. Utilizad el método `sendRedirect` de la respuesta para las redirecciones.

## 3. (\*) Pruebas en servlets

Vamos a realizar una clase de prueba que verifique ciertas cosas sobre la petición que se envía al servlet **ejercicios.DatosServlet2** del ejercicio anterior. Para ello:

- Cread una carpeta de fuentes `test`, y en ella un paquete `ejercicios`
- Cread en este paquete una clase `DatosServlet2Test` que será nuestra clase de prueba. Implementad en ella un método `testPetición` que verifique si la petición es correcta (según el ejercicio anterior, la petición era correcta si todos los campos de texto eran no-vacíos).
- Añadid todas las librerías, configuración y ficheros de propiedades necesarios de Cactus para poder compilar y probar este caso de prueba. Podéis consultar la sesión 4 de Servidores Web y su plantilla para obtener los archivos y ayuda necesaria.

**Ayuda:** además de definir el método `testPetición`, también deberéis definir uno `beginPetición`, para construir una petición como si llamarais a un formulario, y que así el método de test pueda tener una petición que explorar. Este método tendrá una estructura como sigue:

```
public void beginPetición(WebRequest theRequest)
{
    theRequest.setURL("localhost:8080", "/appforms",
```

```
"/servlet/ejercicios.DatosServlet2",  
    null, "aaa=&bbb=1");  
}
```

La clase `WebRequest` que toma como parámetro es del paquete `org.apache.cactus`. El método `setURL` es el que se encarga de construir la URL a la que enviar la petición, indicando, por separado, servidor y puerto, contexto, recurso a llamar, información del path, y parámetros de la petición. Este último campo es el que deberéis modificar para hacer diferentes pruebas, y ver si funciona bien el método de prueba. En el caso de este ejemplo, el parámetro `aaa` se envía vacío, por lo que debería dar error.

