

Trabajo (bloque I)

Fundamentos de Aplicaciones y Servicios Telemáticos
2º Curso Grado en Ingeniería de Tecnologías de Telecomunicación
Departamento de Ingeniería Telemática (DIT)

Universidad de Sevilla

1 Introducción

1.1 Ficheros a realizar

La parte del trabajo correspondiente al bloque I, se corresponde con el desarrollo de los ficheros correspondientes a dos páginas y un fichero de datos:

1.1.1 Usuario

- usuForm.html
- js/funUsu.js
- css/estilo.css

1.1.2 Dirección IP

- ipForm.html
- js/funIp.js
- css/estilo.css

1.1.3 Ficheros de datos (se suministra uno de ejemplo, pero se deben probar otros)

- data/direcciones.json

1.2 Ficheros a utilizar

1.2.1 Ficheros jsp

- jsp/muestraUsu.jsp
- jsp/muestraIp.jsp

1.3 Especificaciones generales

No debe haber llamadas a funciones ECMAScript dentro de los ficheros HTML, ni siquiera las funciones de validación de formularios (las funciones de validación deben asignarse al terminar de carga la página, tal como se explica más adelante con window.onload).

No debe haber especificaciones de formato dentro de los ficheros HTML.

2 Usuario

2.1 Página de creación/modificación de usuario usuForm.html

Es una página HTML5, que usa el juego de caracteres “UTF-8 y su título es: Usuario

El cuerpo de la página debe tener un contenedor de bloque con identificador “contenedor”, y dentro de él un encabezado de nivel 1, un encabezado de nivel 2, dos párrafos, una lista, un formulario y un pie (elemento footer).

2.1.1 Encabezado de nivel 1

Contenido: Creación/modificación de usuario

2.1.2 Encabezado de nivel 2

Contenido: Introduzca usuario y contraseña

Clase: entrada

2.1.3 Párrafo

Contenido: El usuario debe contener un máximo de 11 caracteres

Dentro del párrafo, el fragmento “11 caracteres” debe ser un elemento de línea, y de las clases “warning entrada”.

2.1.4 Párrafo

Contenido: La contraseña debe cumplir las siguientes condiciones:

2.1.5 Lista

Debe ser una lista ordenada

Elementos de la lista: el contenido de cada elemento es:

- debe tener una longitud mínima de 6 caracteres
- debe tener una longitud máxima de 10 caracteres
- debe contener al menos dos caracteres numéricos
- debe contener al menos dos caracteres en mayúsculas
- debe contener al menos dos caracteres en minúsculas
- no debe contener el nombre de usuario

2.1.6 Formulario

De nombre formUsu con identificador formUsu que tenga tres campos:

- Usuario: de nombre usu, con identificador usu, de tipo texto y máximo 11 caracteres. Es obligatorio rellenarlo.
- Contraseña: de nombre contra1, con identificador contra1, que no se vean los caracteres escritos y máximo 10 caracteres. Es obligatorio rellenarlo.
- Repetición de la contraseña: de nombre contra2, con identificador contra2, que no se vean los caracteres escritos y máximo 10 caracteres.

Cada campo estará compuesto por un párrafo que contendrá un elemento label, y éste contendrá un elemento input y texto.

También, dentro de otro párrafo, debe incluir un botón para enviar (“Enviar”) y otro para borrar (“Limpiar”). Antes de enviar se debe usar la función validaFormUsu que impida que se envíen los datos si la contraseña no cumple los criterios comprobados en la función compruebaContra y que muestre mediante alert un mensaje indicando el problema.

Se debe usar el método HTTP que envía la información en el cuerpo del mensaje, y se debe enviar a muestraUsu.jsp (dentro de un subdirectorío de nombre jsp).

2.1.7 Pie

Contiene un encabezado de nivel 2 cuyo contenido es: 2017-18 Fast

2.2 Fichero con funciones en ECMAScript funUsu.js

Debe estar en un subdirectorío de nombre js, cargarse en la cabecera y contener las siguientes funciones:

2.2.1 Funciones de validación

2.2.1.1 compruebaContra (pass, lonMin, minNum, minMayus, minMinus, nomUsu)

Toma como argumento una cadena de caracteres, cuatro números y otra cadena de caracteres.

Analiza la cadena de caracteres y:

devuelve -1 si tiene menos de lonMin caracteres, si no

devuelve -2 si no contiene al menos minNum caracteres numéricos, si no

devuelve -3 si no contiene al menos minMayus caracteres en mayúsculas, si no

devuelve -4 si no contiene al menos minMinus caracteres en minúsculas, si no

devuelve -5 si contiene nomUsu, si no

devuelve 0

2.2.1.2 validaFormUsu()

No toma argumentos. Llama a la función compruebaContra.

Devuelve true si las contraseñas cumplen los requisitos y son iguales y. En caso contrario devuelve false.

Si el valor devuelto por compruebaContra para el campo “Contraseña” es -1, -2, -3, -4, -5, deberá imprimir en pantalla (sólo uno de ellos):

- Error: debe tener una longitud mínima de 6 caracteres
- Error: debe contener al menos dos caracteres numéricos
- Error: debe contener al menos dos caracteres en mayúsculas
- Error: debe contener al menos dos caracteres en minúsculas
- Error: no debe contener el nombre de usuario

Si el valor devuelto por compruebaContra para el campo “Contraseña” es 0, pero las dos contraseñas no son iguales, deberá imprimir en pantalla:

- Error: contraseña debe ser igual en los dos campos

2.2.1.3 inicial()

No toma argumentos. Debe ser llamada cuando termine de cargarse la página (window.onload=inicial;). Debe hacer que se ejecute la función validaFormUsu cuando se vaya a enviar el formulario. Utilice la propiedad onsubmit del elemento formulario para asignarle una función sin nombre que devuelva el valor devuelto por validaFormUsu.

2.3 Fichero de formato estilos.css

Debe estar en un subdirectorio de nombre css y contener las reglas necesarias para que la apariencia sea la deseada y cumpla lo especificado en los comentarios. Se suministra un fichero inicial que hay que completar.

3 Dirección IP

3.1 Página de introducción de dirección IP ipForm.html

Es una página HTML5, que usa el juego de caracteres “UTF-8 y su título es: IP

El cuerpo de la página debe tener un contenedor de bloque con identificador “contenedor”, y dentro de él un encabezado de nivel 1, un párrafo, dos secciones (elemento section) y un pie (elemento footer).

3.1.1 Encabezado de nivel 1

Contenido: Gestión de IPs

3.1.2 Párrafo

Contenido: Introducción de datos para dar de alta o comprobar existencia direcciones IP.

Dentro del párrafo, el fragmento “o comprobar existencia” debe ser texto insertado, y de la clase “revisando”.

3.1.3 Sección

Dentro de la primera sección habrá un encabezado de nivel 2, un formulario y un párrafo

Clase: “col-2”

3.1.3.1 Encabezado de nivel 2

Clase: “entrada”

Contenido: Alta/comprobación IP

3.1.3.2 Formulario

De nombre formIp con identificador formIp que tenga tres campos:

- Dirección IP: de nombre ip, con identificador ip, de tipo texto, tamaño 16, longitud máxima 15 y es obligatorio rellenarlo. Antes de escribir debe mostrar la indicación (placeholder) "A.B.C.D"
- Máscara: de nombre mask, con identificador mask, de tipo número (mínimo 8 y máximo 30), valor por defecto 24 y es obligatorio rellenarlo
- Dirección IP pasarela: de nombre gw, con identificador gw, de tipo texto, tamaño 16, longitud máxima 15 y valor por defecto 192.168.1.2

Cada campo estará compuesto por un párrafo que contendrá un elemento label y un elemento input (del mismo nivel, hermanos), que estarán ligados entre sí con el atributo for.

También, dentro de otro párrafo, debe incluir un botón para enviar (“Enviar”) y otro para borrar (“Limpiar”). Antes de enviar se debe usar la función validaFormullp que impida que se envíen los datos si los datos introducidos no son correctos.

Se debe usar el método HTTP que envía la información en la cabecera del mensaje, y se debe enviar a muestraIp.jsp (dentro de un subdirectorío de nombre jsp).

3.1.3.3 Párrafo

Debe contener un botón con identificador “mostrar”.

Contenido del botón: Mostrar Redes

3.1.4 Sección

Dentro de la segunda sección habrá un encabezado de nivel 2, dos párrafos y una tabla.

Clase: “col-2”

3.1.4.1 Encabezado de nivel 2

Clase: “warning”

Contenido: Tabla redes

3.1.4.2 Párrafo

Contenido: Esta tabla se rellena al pulsar el botón "Mostrar".

3.1.4.3 Párrafo

Clase: “revisando”

Contenido: Se oculta pulsando el mismo botón, que cambia de texto. Al pulsar sobre las filas de la tabla se elimina.

3.1.4.4 Tabla

La tabla tiene un elemento cabecera de tabla (thead), y un elemento cuerpo de tabla (tbody). En la cabecera hay una fila de cabecera con tres celdas: IP Mask Gw. El cuerpo de la tabla está vacío.

3.1.5 Pie

Contiene un encabezado de nivel 2 cuyo contenido es: 2017-18 Fast

3.2 Fichero con funciones en ECMAScript funIp.js

Debe estar en un subdirectorio de nombre js, cargarse en la cabecera y contener las siguientes funciones:

3.2.1 Funciones de validación

Implemente las siguientes funciones para impedir que se envíen los datos si no son correctos. Si son correctos, se deben modificar los campos para que las direcciones que se envíen estén en formato decimal con longitud mínima (sin espacios y sin ceros a la izquierda).

3.2.1.1 procesaIp (cadena)

Toma como argumento una cadena de caracteres que representa una dirección IP en formato decimal, que puede ser errónea (pero se supone que no tiene más de 15 caracteres).

Devuelve un valor negativo si el argumento era erróneo o una cadena de caracteres con la dirección IP correcta y con la longitud mínima (se han eliminado los espacios iniciales y finales y los ceros a la izquierda en cada número).

La función realiza las siguientes acciones (en este orden):

- Elimina los espacios iniciales y finales.
- Separa la cadena en varios elementos usando el punto como separador, y si no hay 4 elementos devuelve -1.
- Para cada elemento:
 - o si contiene un carácter no numérico devuelve -2
 - o si todos los caracteres son numéricos, pero no está entre 0 y 255 devuelve -3

3.2.1.2 ipToBin (cadena)

Toma como argumento una cadena de caracteres, que representa una dirección IP correcta en formato decimal (no hay que comprobarla).

Devuelve una cadena de 32 caracteres, que es la representación en binario de la dirección IP.

3.2.1.3 validaFormulIp()

No toma argumentos. Llama a las funciones procesaIp e ipToBin.

Devuelve false si alguna dirección IP es errónea (procesaIp devuelve un valor negativo) o no están en la misma subred (no coinciden en binario las dos direcciones hasta la longitud de la máscara). En caso contrario devuelve true y además hace que el formulario envíe en los campos correspondientes el valor devuelto por procesaIp.

Si el valor devuelto por procesaIP con el campo correspondiente a “Dirección IP” es -1, -2, -3, deberá imprimir en pantalla (sólo uno de ellos):

- Error en Dirección IP: no contiene sólo tres puntos
- Error en Dirección IP: no contiene cuatro campos sólo numéricos
- Error en Dirección IP: contiene números fuera de rango

Si el valor devuelto por procesaIP con el campo correspondiente a “Dirección IP pasarela” es -1, -2, -3, deberá imprimir en pantalla un mensaje similar al anterior, pero cambiando “Dirección IP” por “Dirección IP pasarela”.

Si ninguna de las direcciones es errónea, debe comprobar si están en la misma subred, y si no es así imprimir en pantalla:

- Error: no pertenecen a misma subred

Ejemplos:

| Dirección IP | Valor devuelto por procesaIP | Valor devuelto por ipToBin | Valor devuelto por validaFormuIp |
|----------------|------------------------------|------------------------------------|----------------------------------|
| “ 008.8.08.8 “ | “8.8.8.8” | 0000100000000100000000100000001000 | true |
| “8.8.8.8.” | -1 | | false |
| “8.8.8. 8” | -2 | | false |
| “8.8.8.256” | -3 | | false |

Ejemplos:

| Dirección IP Long. máscara Dir. IP pasarela | Valor devuelto por procesaIP | Valor devuelto por ipToBin | Valor devuelto por validaFormuIp |
|---|------------------------------|-------------------------------------|----------------------------------|
| “ 008.8.08.8 “ 24 | “8.8.8.8” | 0000100000000100000000100000001000 | true |
| “8.8.8.001 ” | “8.8.8.1” | 00001000000001000000001000000000001 | |
| “ 008.8.08.8 “ 24 | “8.8.8.8” | 0000100000000100000000100000001000 | false |
| “8.8.9.001 ” | “8.8.9.1” | 00001000000001000000001001000000001 | |

Nota: si se introduce la dirección “ 008.8.08.8 “ lo que se debe enviar es “8.8.8.8”.

3.2.2 Funciones de eventos

Implemente las siguientes funciones para que cuando se pinche en el botón con identificador “mostrar” se realice una petición AJAX y el contenido del fichero direcciones.json (en directorio data) se muestre en el cuerpo de la tabla. Si se vuelve a pinchar en el botón se eliminarán todas las filas del cuerpo de la tabla. Cada vez que se pinche el botón cambiará su texto adecuadamente (debe ser “Ocultar Redes” cuando el cuerpo de la tabla contenga alguna fila).

Cuando se pinche en una fila del cuerpo de la tabla, se debe eliminar esa fila, y si se eliminan todas las filas también se debe cambiar el texto del botón a “Mostrar Redes”.

Para más información ver fichero direcciones.json de ejemplo.

3.2.2.1 creaFila()

Recibe como argumento un objeto que tiene 3 campos (representan dirección ip, máscara y dirección ip pasarela) y devuelve una cadena que contiene el HTML correspondiente a una fila de una tabla.

3.2.2.2 mostrarRedes()

Recibe como argumento una tabla con objetos que tienen 3 campos y representan dirección ip, máscara y dirección ip pasarela, y debe representarlos en el cuerpo de la tabla. Para ello usa la función creaFila y va rellenando el cuerpo de la tabla con el valor devuelto. Además, hace que cuando se pinche en una fila, ésta se elimine, y si se eliminan todas las filas cambia el texto del botón a “Mostrar Redes”.

3.2.2.3 pedirRedesURL()

Debe cambiar el texto del botón entre “Mostrar Redes” y “Ocultar Redes” dependiendo si el cuerpo de la tabla contiene alguna fila o no.

Si el cuerpo de la tabla no contiene ninguna fila, debe cambiar el texto del botón a “Ocultar Redes” y realizar una petición AJAX con el método GET de la URL “data/direcciones.json”. El valor recibido como respuesta tiene el formato JSON y se debe almacenar en una variable que se pasa como argumento a la función mostrarRedes.

Si el cuerpo de la tabla contiene alguna fila, debe dejar vacío el cuerpo de la tabla y cambiar el texto del botón a “Mostrar Redes”.

Nota: Antes de realizar la petición, la función debe comprobar si el valor del campo “Dirección IP” es correcto (la función procesaIp devuelve un string) y en ese caso añadir a la URL dos parámetros de nombre ip y mask. Los valores de los parámetros serán el valor devuelto por procesaIp y el valor del campo “Máscara” respectivamente.

3.2.2.4 inicial()

No toma argumentos. Debe ser llamada cuando termine de cargarse la página (window.onload=inicial;). Debe hacer que se ejecute la función pedirRedesURL cuando se pulse el botón con identificador “mostrar”. También debe hacer que se ejecute la función validaFormIp cuando se vaya a enviar el formulario. Utilice la propiedad onsubmit del elemento formulario para asignarle una función sin nombre que devuelva el valor devuelto por validaFormIp.

3.3 Fichero de formato estilos.css

Debe estar en un subdirectorio de nombre css y contener las reglas necesarias para que la apariencia sea la deseada y cumpla lo especificado en los comentarios. Se suministra un fichero inicial que hay que completar.

4 Indicaciones adicionales

A continuación, algunas funciones y propiedades que pueden ayudar en la resolución (no es obligatorio usarlas, puede usar otras).

Algunas funciones de cadena:

```
.trim()  
.substr()  
.parseInt()
```

Algunas formas de comprobar si una cadena contiene o no caracteres no numéricos (recuerde que si se introducen las IPs 1.1.1.1 ó 1.1. 1.1 ó 1.1.a1.1 deben dar error):

```
if (/\\D/.test(cadena)) // contiene algún carácter no numérico  
  
if (!/^\\d+$/.test(cadena)) // la exp. reg. es cierta si la cadena sólo  
contiene uno o más dígitos, la condición es lo contrario
```

Puede consultar el funcionamiento de las expresiones regulares en:

https://www.w3schools.com/jsref/jsref_obj_regexp.asp

Algunas funciones de número que ayudan a pasar a binario (no confundir con funciones de cadena del mismo nombre):

```
.toString(2)
```

Para asociar una función con argumentos a un elemento con un evento:

```
.addEventListener("click", function(){.....});
```

Para asociar una función de validación a un elemento formulario:

```
.onsubmit=function(){return funcionDeValidacion()};
```

Para ejecutar una función cuando se termine de cargar la página:

```
window.onload = nombreDeFuncion;
```