

PROYECTO04_EVA2: DOM

Se pide una aplicación que, a partir de la información proporcionada por un fichero de texto tipo *csv*, *lst* y *txt*, sea capaz de generar otro fichero con las palabras cuyo tamaño sea el proporcionado por el usuario.

Funcionamiento:

1. En primer lugar, seleccionamos el fichero, puede ser cualquiera de los presentados en el actual proyecto: *palabras1.lst* y *palabras2.txt*.

Pulse en el botón para seleccionar su fichero de palabras:

2. A continuación, nuestra aplicación generará el formulario necesario para que el usuario pueda elegir las letras por palabra y las columnas de la tabla informativa generada. El botón Examinar sigue activado por si queremos cargar otros ficheros.

¿Letras x palabra?: ¿Columnas x fila?:

3. Una vez introducida esa información, se activará el botón Genera palabras.

¿Letras x palabra?: ¿Columnas x fila?:

4. Al pulsarlo, generará una tabla con las opciones escogidas, activándose el botón Genera CSV:

Pulse en el botón para seleccionar su fichero de palabras: <input type="button" value="Examinar..."/> <input type="text" value="palabras1.lst"/>					
¿Letras x palabra?: <input type="text" value="4"/> ¿Columnas x fila?: <input type="text" value="6"/> <input type="button" value="Genera palabras"/> <input type="button" value="Genera CSV"/>					
agil	aloe	amel	apex	area	azoe
abab	abad	abaz	abes	abey	abia
abra	abur	acal	acle	acme	acne
acre	acta	adan	adir	ador	adra

5. Al pulsar dicho botón se descargará un fichero *csv* con la información mostrada en la tabla. El nombre del fichero será **palabras<nº de Letras>.CSV**

Pulse en el botón para seleccionar su fichero de palabras:

Mostrar todas las descargas

agil	aloe	amel	apex	area	azoe
------	------	------	------	------	------

Indicaciones:

- ⇒ Excepto el primer apartado, todo lo demás será generado usando DOM.
- ⇒ El programa deberá dar una salida adecuada a cualquier interacción con el usuario, estando preparado para cualquier gestión de la interfaz.
- ⇒ El programa debe darnos a elegir para abrir, por defecto, archivos *txt*, *lst* y *csv*.
- ⇒ Funciones y variables deben estar, sobre todo, en español.

Salida:

- ✓ El formulario requerido.
- ✓ La tabla con las palabras.
- ✓ El fichero *csv* generado.

Salida documentada:

- ✓ Es importante la correcta documentación de métodos y líneas de código que se consideren importantes y/o complejas. Usaremos la herramienta JSDOC.
- ✓ Debido a que este proyecto está orientado a la búsqueda de información y posterior implementación del código, se requerirá una correcta explicación de API, objetos y métodos usados.
- ✓ Además, se requerirá un fichero ***explicación.docx*** donde se explique el funcionamiento de la aplicación atendiendo a los métodos empleados, explicándolos de un modo resumido, pero claro. Ejemplo:

Empezamos en la página html index, donde creamos el formulario principal gracias al cual seleccionaremos el fichero con los datos. Le asignamos un listener que pondrá en marcha el resto de la web llamando a la función... la cual muestra... calcula... a través de la función... etc

- ✓ No me expliquéis código ni comentarios, simplemente quiero estar seguro de que entendéis la aplicación que habéis creado.
El proyecto estará organizado en carpetas.

Entrega (opciones):

- ☞ En plataforma moodleSed.
- ☞ Enviar enlace github con el proyecto al usuario **belisoc**. El nombre del proyecto será **PROYECTO04EVA2_DOM**