

Ejercicio 1

Enunciado.

Se entrega un fichero HTML en el que se muestra un formulario de introducción de los datos de un envío. Hay un campo para el número de referencia del pedido y otro para el peso en Kg.

- El número de referencia siempre está formado por una sola letra del conjunto {A, B, C, D, E} y a continuación exactamente 5 dígitos decimales
- El peso en Kg siempre está formado por una o dos cifras decimales, seguidas de un punto o una coma (a elegir por el programador) y una última cifra decimal. Es decir, los pesos pertenecen al intervalo [0.1 , 99.9]

Se pide la creación de un archivo “ej1.js” que contenga exclusivamente código JavaScript y donde se resuelvan los siguientes objetivos:

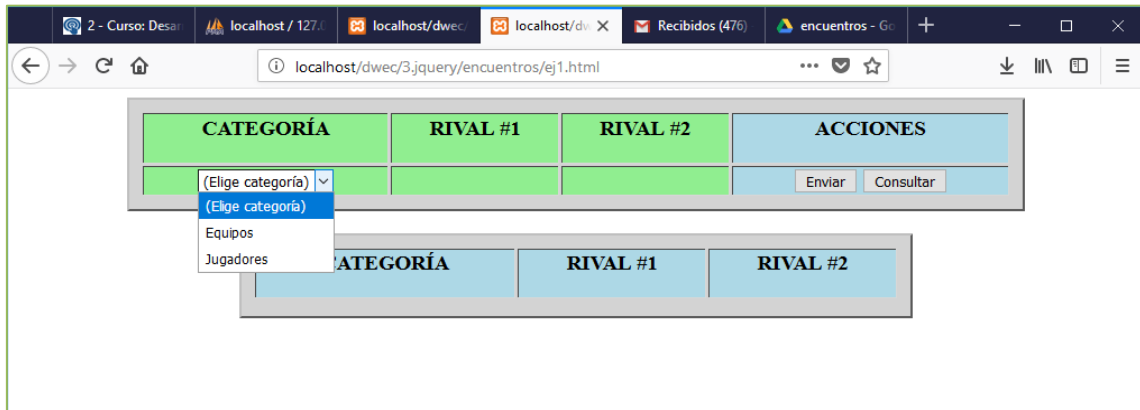
Objetivos.

- a) **[OBLIGATORIO 1]** Añadir funcionalidad al botón “Enviar”. El formulario no se envía si algún campo no es correcto y se envía si todos los campos son correctos mostrando en ese caso el fichero “formOK.html” entregado por el profesor.
- b) **[OBLIGATORIO 2]** Añadir mensajes de error. El formulario muestra de forma llamativa (colores y/o mensajes de advertencia, pero no “alerts”) todos los campos que presentan irregularidades. Estas advertencias visuales desaparecen si los errores son corregidos.
- c) **[OPCIONAL 1]** Cada cierto tiempo, el formulario es verificado automáticamente y se mostrarán los correspondientes mensajes de error, aunque el usuario no presione el botón “Enviar”.
- d) **[OPCIONAL 2]** En el campo de texto para el número del envío, aceptar únicamente pulsaciones de teclas numéricas.

Ejercicio 2

Enunciado.

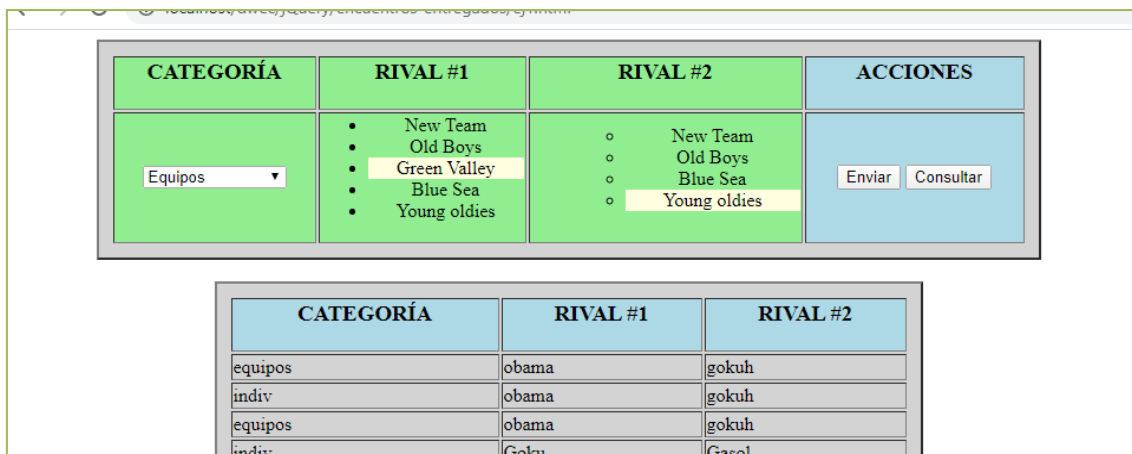
Necesitamos programar la funcionalidad de una web para planificar encuentros de una competición. La web tiene este aspecto:



Los encuentros pueden ser de dos categorías: por equipos o por jugadores individuales. Al elegir una de las dos categorías, automáticamente se rellenará la celda de RIVAL#1 con los datos de un JSON que nos llega tras la llamada al fichero `load.php` con los parámetros correspondientes.

Cuando el usuario haga clic en un elemento de RIVAL#1 quedará resaltado con una clase CSS definida para esto y además provocará que se rellene la celda de RIVAL#2, esta vez sin necesidad de iniciar comunicaciones AJAX adicionales, sino mediante un algoritmo. En la celda de RIVAL #2 deben aparecer disponibles todos los elementos de la celda RIVAL #1 excepto el seleccionado. El usuario podrá entonces hacer clic sobre un elemento de RIVAL #2 para seleccionarlo y provocar que el **botón Enviar** se habilite.

Pulsar el **botón Enviar** hará que se inicie otra comunicación AJAX con el fichero `save.php` para insertar en la base de datos ese encuentro entre los rivales seleccionados por el usuario. Este fichero PHP inserta un encuentro, pero también devuelve en formato XML todos los encuentros que actualmente se encuentran en la base de datos. Este objeto XML debe ser procesado y mostrado correctamente en la tabla inferior cada vez que se inserta un nuevo encuentro.



Opcionalmente, el botón Consultar permitirá mostrar en la tabla inferior todos o parte de los encuentros de la base de datos sin necesidad de insertar previamente un nuevo encuentro. Para ello, clicar el botón provocará una comunicación AJAX con un tercer fichero PHP llamado `list.php` que devuelve un objeto XML con similar estructura. Este fichero PHP puede devolver todos los encuentros si no se le pasa parámetro o devolver solamente los encuentros de una categoría concreta si se le indica a través del parámetro “cat”, que puede tomar los valores “eq” y “indiv”.

Objetivos:

[OBLIGATORIO] Programar la funcionalidad que permita lo siguiente:

- Rellenar mediante *Fetch* la celda de “RIVAL #1” como se indica en el enunciado
- Rellenar mediante algoritmos (sin *Fetch*) la celda de “RIVAL #2” como se indica en el enunciado
- Clicar el botón “Enviar” para que haga lo que se pide en el enunciado
- Rellenar la tabla de salida cada vez que se haga una inserción en la base de datos

[OPCIONAL] Programar la funcionalidad que permita lo siguiente:

- Clicar el botón “Consultar” para que se rellene la tabla de salida sin necesidad de que se haga previamente una inserción en la base de datos
- Que los rivales sean clicables por delegación de eventos
- Implementar los filtros “Mostrar equipos” y “Mostrar jugadores” haciendo uso de *dataset*

Conocimientos examinados.

En este examen se **requiere** que el alumno o alumna sepa:

- Asociar código al evento clic de ratón sobre un elemento del árbol DOM
- Verificar diferentes campos de un formulario a partir de un patrón especificado
- Impedir el envío de un formulario si alguno de sus campos es erróneo
- Manipular, mostrar y ocultar contenido del árbol DOM
- Recuperar un recurso de la red mediante la API Fetch
- Procesar respuestas JSON y XML
- Insertar filas en una tabla

En este examen se **valora** que el alumno o alumna sepa:

- Controlar la entrada por teclado dentro de un campo de texto de un formulario
- Manejar otros eventos adicionales sobre elementos de la página
- Establecer tareas automáticas que se repiten cada cierto tiempo establecido
- Delegar eventos
- Aprovechar las ventajas de *dataset*

Calificación del examen.

La correcta resolución de los objetivos obligatorios otorga una calificación de 5 sobre 10.

Solamente si los objetivos obligatorios han sido resueltos adecuadamente se procederá a sumar puntos adicionales por cada uno de los objetivos opcionales solucionados con éxito.