

Apellidos y Nombre:

Hora de inicio: 8:30

Duración: 5 horas y media.

NOTA: Para la realización del examen solo se puede consultar el codecar. No se puede consultar libros, ni apuntes, ni páginas web, ni ficheros de prácticas, ejercicios realizados, ni cualquier otro fichero que contenga código. A la hora de evaluar los ejercicios se tendrá en cuenta la optimización del código y la utilización correcta de los recursos que se pueden utilizar

- 1.-) A partir del fichero **ejercicio-01.html**, vamos a realizar un proceso de validación de dicho formulario, dichos formularios no se van a enviar si se produce algún error en el formulario.(Se puntúa sobre 3 puntos).
 - a.-) El **documento HTML NO se puede modificar y NO se pueden utilizar DOM ni jQuery.**
 - b.-) Cuando se pulse el botón de "**Enviar**" vamos a llevar la validación. Se deberán de tener en cuenta las validaciones descritas más adelante. Vamos a sacar un único cuadro de mensaje en el que aparezcan todos los errores, cada uno en una línea
 - c.-) Las cajas de texto de unidades del producto disponibles y número de unidades mínimas solamente va a admitir dígitos.
 - d.-) La caja de texto del precio del producto solamente va a admitir dígitos y un único punto.
 - e.-) Cuando nos situemos sobre las cajas de texto su contenido se va a borrar, el color de los caracteres será amarillo y el color de fondo rojo, cuando abandonemos la caja de texto la visualización será la inicial.

Apellidos y Nombre:

f.-) Para las validaciones deberemos tener en cuenta los siguientes puntos:

i. Validaciones exhaustivas:

1. El **código del producto** va a contener 7 o 11 dígitos, de los cuales el último dígito.
2. La **descripción del producto** va a empezar por cuatro letras y va a terminar por una letra, en medio puede tener letras, espacios, el guión y dígitos, con un número de caracteres comprendidos entre 10 y 23.
3. La **fecha de alta en la empresa** deberá tener una fecha correcta, en la cual el día va a tener dos dígitos, al igual que el mes y el año va a tener cuatro dígitos, siendo su valor mínimo 1930 y su valor máximo el año actual
4. El **precio del producto** va a ser un número, que la parte entera siempre va a ser mayor que 10 y la parte decimal puede tener hasta dos dígitos.

ii. Validaciones mediante expresiones regulares directas:

1. El **nombre de la empresa que distribuye el producto** va a empezar por tres letras y va a terminar por una letra o un dígito, en medio puede tener letras, espacios y el punto, con un número de caracteres comprendidos entre 10 y 27.
2. El **código de la empresa** va a tener tres dígitos, un punto, luego cuatro letras que puede tener uno de los

Apellidos y Nombre:

siguientes valores: ABCE, CADE, FEGU, IJOK, LIMA; a continuación, va a tener de 5 a 8 dígitos, un punto y va a terminar con un conjunto de 5 caracteres, que pueden ser letras o dígitos.

3. La **dirección de la empresa** va a empezar por dos letras, va a terminar por letra, dígito o punto y en medio va a contener letras, números, el punto, la coma, el guion y los caracteres “o”, “a”, “\”, con una longitud comprendida entre 10 y 28 caracteres.
4. La **localidad de la empresa** va a contener de cinco a veinte letras.

iii. Validaciones mediante expresiones regulares del objeto RegExp.

1. El **número de unidades mínimas** va a ser un número entero de entre 2 y 4 dígitos, siendo el valor mínimo de 30
2. Las **unidades del producto disponibles** van a ser un número entero de entre 2 y 7 dígitos.
3. La **familia a la que pertenece el producto** va a empezar con cinco letras y va a terminar con tres letras, en medio puede tener letras, dígitos, espacios, los caracteres el guion, el punto y la barra “|” va a tener de 10 a 21 caracteres.
4. La **empresa de transporte utilizada** va a tener uno de los siguientes valores

Apellidos y Nombre:

- a. SEUR.
- b. NACEX.
- c. DHL.
- d. MRW.

- iv. Se debe seleccionar un tipo de IVA.
- v. Se deben seleccionar al menos un sector de producción.
- vi. Se deben seleccionar al menos tres países en los que se vende el producto.

2.-) A partir del fichero **ejercicio-02.html** realiza las siguientes acciones utilizando **DOM**. El **código HTML NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador**. (Se puntúa sobre 2 puntos).

- a.-) Cuando seleccionamos o deseccionamos una localidad vamos a hacer que todos los monumentos de las localidades seleccionadas aparezcan en la lista no ordenada que hay debajo, solo los correspondientes a las localidades que están seleccionadas en ese instante. Los nombres de los monumentos deberán aparecer ordenados en modo ascendente. Aquí tienes las localidades y los monumentos, al incluirlos deberás añadir al final del nombre del monumento la localidad. Esta lista la tienes en el fichero monumentos.txt en el directorio textos.

Apellidos y Nombre:

Localidad	Monumentos
Burgos	“Arco de Santa María”, “Monasterio de San Juan”, “Puente de Santa María”, “Arco de San Esteban”, “Solar del Cid”, “Arco de Fernán González”, “Antiguo Seminario Mayor”, “Monasterio de Santa María la Real de las Huelgas”, “Catedral”, “El Cid Campeador”
Córdoba	“Mezquita-Catedral”, “Alcázares de los Reyes Cristianos”, “Medina Azahara”, “Puente Romano”, “Caballerizas Reales”, “Torre de la Calahorra”, “Templo Romano”, “Torre de la Malmuerta”, “Alminar de San Juan”, “Mausoleos Romanos”, “Capilla de San Bartolomé”
A Coruña	“Torre de Hércules”, “Obelisco Millenium”, “iglesia de las Capuchinas”, “Castillo de San Antón”, “Convento de Santa Bárbara”, “Convento de Santo Domingo”, “Iglesia de San Jorge”, “iglesia de San Nicolás”, “Colegiata de Santa María”, “Iglesia de Santiago”
León	“Catedral”, “Basílica de San Isidoro”, “Casa de Botines”, “Convento de las Concepciones”, “Cripta de Puerta Obispo”, “Iglesia de los Padres Capuchinos”, “Iglesia de Nuestra Señora del Camino”, “Iglesia de San Marcelo”, “Iglesia de Santa Ana”
Mérida	“teatro Romano”, “Templo de Diana”, “Acueducto de los Milagros”, “Puente romano sobre el Guadiana”, “Anfiteatro Romano”, “arco de Trajano”, “Alcazaba árabe”, “Basílica de Santa Eulalia”, “Foro romano”, “Circo Romano”, “Catedral de Santa María”, “Puente romano sobre el Albarregas”, “Templo

Apellidos y Nombre:

Localidad	Monumentos
	de Marte"
Salamanca	"Catedral Nueva", "Catedral Vieja", "Fachada de la Universidad", "Casa de las Conchas", "La Clerencia", "convento de San Esteban", "Plaza Mayor", "Casa Lis"
Segovia	"Alcázar", "Acueducto", "Catedral", "Real Casa de Moneda", "Casa de los Picos", "Iglesia de San Martín", "Iglesia de la Santísima Trinidad", "Iglesia de San Esteban", "Iglesia de San Millán", "Iglesia de la Vera Cruz", "Iglesia del Corpus Cristi", "Monasterio del Parral"
Sevilla	"Giralda", "Torre del Oro", "Archivo de Indias", "Casa Pilatos", "Catedral", "Palacio de San Telmo", "Hospital de la Caridad", "Parque de María Luisa", "Reales Alcázares", "Real Maestranza de Caballería", "Plaza España", "Baílica de la Macarena", "Jardines de Murillo"
Zamora	"Catedral", "Puente de Piedra", "Puerta del Obispo", "Puerta de Doña Urraca", "Muralla", "Monasterio de la Carballeda", "Puerta de la Traición", "Molinos de Agua", "Castillo", "Palacio de los Monos"

b.-) Cuando se pulse el botón **Borrar** deberemos comprobar que hay un país seleccionado en la lista desplegable y entonces deberemos borrar de la tabla todas las regiones del país seleccionado y de la lista borraremos el país seleccionado. Cuando se hayan eliminado todos los países se van a añadir de nuevo

Apellidos y Nombre:

(todos los países) y cambiaremos el nombre del botón por **Añadir** y cuando se pulse el botón con este nuevo literal vamos a añadir las regiones del país seleccionado a la tabla, solamente se puede añadir una vez las regiones de los países. Una vez añadidas todas las regiones el literal del botón volverá a ser **Borrar**; y se desarrollaran las acciones descritas anteriormente. Hay un fichero llamado regiones.txt con las regiones de cada país.

3.-) A partir del fichero **ejercicio-03.html** realiza las siguientes acciones utilizando **jQuery**. El **código HTML NO se puede modificar y los eventos se deben programar mediante jQuery**. (Se puntúa sobre 2 puntos).

a.-) Cuando pulsemos en el botón **Aplicar** vamos a hacer qué en la tabla, con identificador coches, en las líneas impares los datos de los extremos aparezcan con el color de los caracteres en verde y el color de fondo en naranja y la celda interior el color de los caracteres va a ser rojo y el color de fondo va ser amarillo. En las líneas pares va a suceder al revés, las celdas de los extremos van a tener como color de los caracteres rojo y color de fondo amarillo, mientras que la celda interior va a tener como color de los caracteres verde y color de fondo naranja.

Apellidos y Nombre:

- b.-) Crea una clase de css llamada **saltarina** con las siguientes características: tipo de letra "**Calibri**" con un tamaño de **24 puntos**, color de fondo **Azul** y color de los caracteres en **Blanco**. Haz que el texto que tenemos referente a salamanca, párrafo de la clase **salamanca**, cuando el cursor del ratón entre dentro del texto utilice las características de la clase **salmantina** y cuando salga vuelve a su estado normal. **No se puede utilizar la propiedad css, ni asignarle el atributo style.**
- c.-) Cuando se pulse el botón de **Cambiar** vamos a eliminar de la tabla todas las provincias de las comunidades autónomas que estén seleccionadas en la lista desplegable. Las provincias que se borren de la tabla se van a incluir en una lista ordenada que pondremos debajo, si ésta no existe se deberá de crear.
- 4.-) A partir del fichero **ejercicio-04.html** realiza las siguientes acciones **DOM y el método fetch**. El código **HTML y PHP NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador.**:(Se puntúa sobre 1 puntos)
- a.-) Cuando se pulse el botón "**Calcular Velocidad**" vamos a comprobar que hay al menos una línea de datos en la tabla.

Apellidos y Nombre:

- b.-) Vamos a generar con todos los datos de la tabla, excepto la columna de Velocidad Final, (cada fila va a ser un registro) una cadena en xml en la cual el dataset es **resultado**, el nombre de los registros es **vehículos**, los nombre de los campos son: **coche**, **velocidad**, **aceleración** y **tiempo**.
- c.-) Llamaremos al programa **velocidad.php** pasándole la cadena xml.
- d.-) Nos va a devolver una cadena XML en la cual a cada registro se le ha añadido el campo **velfinal**, que se va a mostrar en la columna Velocidad final de la tabla.
- 5.-) A partir del fichero **ejercicio-05.html** realiza las siguientes acciones **Utilizando jQueryy utilizando el método \$.ajax**. El **código HTML y PHP NO se puede modificar y los eventos se deben programar mediante jQuery**(Se puntúa sobre 1 puntos)
- a.-) Cuando se pulse el botón “**Calcular Distancia**” vamos a comprobar que hay al menos una línea de datos en la tabla.
- b.-) Con los datos de cada línea de la tabla vamos a generar un objeto con las propiedades: **moto**, **velocidad**, **aceleración** y **tiempo**. Y los objetos generados se van a almacenar en un array.

Apellidos y Nombre:

- c.-) Llamaremos al programa **distancia.php** pasándole los datos en JSON.
- d.-) Cuando nos devuelva un objeto JSON, que va a ser un array de objetos. Cada objeto va a tener las propiedades **moto, velocidad, aceleración, tiempo y distancia**. Vamos a mostrar la distancia de cada objeto en la columna de **Espacio recorrido** de la tabla.
- 6.-) A partir del fichero **ejercicio-06.html** realiza las siguientes acciones Utilizando el método **ajax puro y el DOM**. El **código HTML y PHP NO se puede modificar y los eventos se deben programar mediante el DOM para cualquier navegador.**:
(Se puntúa sobre 1 puntos)
- a.-) Cuando se seleccione un equipo y una temporada se va a generar una cadena XML con el equipo y la temporada en donde el nombre del dataset es **datos**, el nombre del registro es **resultado**, el nombre de los campos son: **equipo y temporada**.
- b.-) Se va a llamar al programa **resultados.php**, pasándolo como parámetro la cadena XML generada en el punto anterior.
- c.-) Cuando nos devuelva la solicitud, la información la recibiremos en una cadena XML y deberemos mostrar los resultados en las cajas de texto correspondientes.