	<p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Recuperación del Bloque II – 06/03/2023</p>	<p align="center">DEPARTAMENTO DE INFORMÁTICA</p>
---	--	--

Ejercicio 1 (3,5 puntos)

Desarrolle un script cuya función es realizar una cuenta atrás. El usuario ha de introducir los valores de las horas, minutos y segundos correctamente para poder empezar dicha cuenta. Por tanto, podemos dividir este ejercicio en dos partes:

- Validación de los campos a introducir. Dichos campos han de ser validados de modo que no puedan tener valores en blanco y, en caso de que el número de segundos o minutos sea mayor de 60, se actualicen los campos correspondientes. Las funciones del experimento correspondiente con esta parte son:
 - actualizaHoras(): evita que el campo correspondiente a las horas esté vacío (cadena vacía). Si estuviera vacío lo pone a cero.
 - actualizaMinutos(): evita que el campo correspondiente a los minutos esté vacío, si estuviera vacío lo pone a cero. Comprueba que el rango introducido sea correcto y, en caso contrario (> 59), actualiza el campo de las horas.
 - actualizaSegundos(): evita que el campo correspondiente a los segundos esté vacío, si estuviera vacío lo pone a cero. Comprueba que el rango introducido sea correcto y, en caso contrario (> 59), actualiza los campos correspondientes (minutos y horas).
- Cronómetro. En este caso se hace uso de las funciones `setTimeout` para actualizar la cuenta y `clearTimeout` para cancelar la cuenta iniciada con la función anterior. Las funciones que corresponden al cronómetro son:
 - detenerCrono(): para el cronómetro llamando a la función de JavaScript `clearTimeout`.
 - iniciarCrono(): comienza la cuenta atrás y llama a `actualizaCrono`.
 - actualizaCrono(): función que actualiza el cronómetro. Para ello, usa la función de JavaScript `setTimeout` para que ejecute `actualizaCrono` una décima más tarde. De esta forma, el cronómetro se irá actualizando diez veces por segundo. Cuando el cronómetro llega a cero se debe levantar un mensaje de alerta indicando tal circunstancia, parando el cronómetro en ese momento.


Para resolver este ejercicio se facilitan dos funciones que ayudan a transformar los valores de horas, minutos, segundos y décimas en una cadena de texto con el formato adecuado para la salida de este ejercicio.

Ejercicio 2 (3,5 puntos)

Partiendo de la plantilla facilitada con el ejercicio, implementar el código javascript que permita inicializar la tabla con el siguiente estilo al abrir el documento, donde las filas impares estarán destacadas. Para ello añada la clase *“destacada”* a las filas necesarias.

Recuperación Bloque II - Ejercicio 2		
Año	Fabricante	Modelo
1965	Ford	Mustang
1970	Toyota	Corolla
1979	AMC	Jeep CJ-5
1983	Ford	EXP
1985	Dodge	Daytona
1990	Chrysler	Jeep Wrangler Sahara
1995	Ford	Ranger
1997	Chrysler	Jeep Wrangler Sahara
2000	Chrysler	Jeep Wrangler Sahara
2005	Chrysler	Jeep Wrangler Unlimited
2007	Dodge	Caliber R/T

Además, habrá que dotar de la funcionalidad adecuada para que al colocar el cursor del ratón encima de la tabla, ésta cambia de estilo pasando las filas no destacadas a destacadas y

	<p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Recuperación del Bloque II – 06/03/2023</p>	<p align="center">DEPARTAMENTO DE INFORMÁTICA</p>
---	--	--

viceversa, tal como se puede ver en la imagen siguiente. La tabla vuelve a su estado inicial al salir el cursor de esta.

Recuperación Bloque II - Ejercicio 2

Año	Fabricante	Modelo
1965	Ford	Mustang
1970	Toyota	Corolla
1979	AMC	Jeep CJ-5
1983	Ford	EXP
1985	Dodge	Daytona
1990	Chrysler	Jeep Wrangler Sahara
1995	Ford	Ranger
1997	Chrysler	Jeep Wrangler Sahara
2000	Chrysler	Jeep Wrangler Sahara
2005	Chrysler	Jeep Wrangler Unlimited
2007	Dodge	Caliber R/T

Ejercicio 3 (3 puntos)

Partiendo de la plantilla facilitada con el ejercicio, implementar el código javascript que permita validar el formulario siguiendo las siguientes especificaciones:

- En la capa de salida se mostrará una lista con los campos vacíos y otra con los campos con errores. En el caso de que todos los campos sean correctos se procesará el formulario.
- Todos los inputs deben ser de tipo texto y ninguno de ellos podrá quedarse en blanco.
- Tanto la marca como el modelo deben tener una única palabra que comienza por letra mayúscula y la pueden seguir minúsculas o mayúsculas. Habría que tener en cuenta la Ñ y los caracteres tildados.
- La fecha de matriculación tendrá el formato DD-MM-AAAA, por ejemplo: 06-03-2023.
- La matrícula tendrá tres posibles formatos en función del tipo de matrícula seleccionado:
 - Actual: la matrícula estará compuesta por cuatro dígitos seguidos de tres letras mayúsculas. Ejemplo: 1234BCD.
 - Antigua: la matrícula tendrá el distintivo provincial que se usaba antes del año 2000. Estará compuesta por uno o dos caracteres en mayúsculas que representan a la provincia, un guion seguido de cuatro dígitos, para terminar con otro guion y una o dos letras mayúsculas. Ejemplo: SE-7182-AJ.
 - Histórica: hay vehículos que pueden solicitar este tipo de matrículas si tienen más de 30 años, su modelo ha dejado de fabricarse y está en estado original sin cambios fundamentales. Dicha matrícula empieza por una H, seguido de cuatro dígitos, y tres letras mayúsculas. Ejemplo: H1234BCD.
- El color puede ser una cadena de texto de una sola palabra en mayúsculas o minúsculas.
- En la capa de salida se mostrará una lista con los campos vacíos y otra con los campos con errores. En el caso de que todos los campos sean correctos se procesará el formulario.

Normas:

- Se debe codificar todo el código en los ficheros con extensión *js* ubicados en las carpetas llamadas *js* de cada ejercicio.
- No se puede modificar ningún fichero *html*.
- Entregar un único fichero *zip* con las carpetas de los proyectos implementados para solucionar cada uno de los tres ejercicios.
- Todo el código debe seguir el estándar ES6.