

# EXAMEN 2

## 2º DAW CFGS

### DWC

- NO SE PUEDEN USAR BUCLES DE NINGUN TIPO EN LOS CODIGOS A IMPLEMENTAR A MENOS QUE LO INDIQUE EL ENUNCIADO.
- SE DEBEN RESPETAR Y CONSEGUIR LAS ESPECIFICACIONES INDICADAS PARA CADA APARTADO.
- SE PUEDE USAR LA DOCUMENTACION DE LA WEB  
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- NO SE PUEDE USAR NINGUNA OTRA WEB PARA CONSULTA DURANTE EL EXAMEN PRACTICO.
- DEBES ENTREGAR SOLAMENTE EL ARCHIVO JAVASCRIPT CON EL RESULTADO DE TU IMPLEMENTACIÓN PARA ALCANZAR LA SOLUCIÓN.
- SI LA SOLUCION PROPUESTA NO CUMPLE CON LOS REQUISITOS O FALLA EN ALGUNO DE LOS REQUISITOS ES DADA COMO NO RESUELTA.

## Ejercicio 1 (1 punto)

Tenemos un array y la web en formato HTML, así como la hoja de estilos CSS de la web, que nos proporcionan y debemos hacer en JavaScript lo siguiente:

- ☐ Los datos que contiene el array, que ya está definido en JavaScript, los queremos guardar en una lista desordenada.
- ☐ Los datos en formato JSON, el formato JSON es una cadena de caracteres, se crean mediante unos datos que provienen de un array.
- ☐ Realizar una aplicación en JavaScript que con un botón llamado “Cargar ciclos” que al hacer clic en ese botón cargue los datos de dicho fichero (localStorage) en una lista desordenada en la página web.
- ☐ Esto quiere decir que tenemos dos eventos en este ejercicio, un evento que sucede cuando se carga todo el DOM y otro evento dentro de este último que solamente sucede cuando se hace clic en el botón oportuno.
- ☐ **La resolución del ejercicio esta completamente guiada en el archivo JavaScript indicándote hasta el nombre de las variables y el fin del bucle, función y evento necesario.**
- ☐ Solamente debes saber por ti mismo como crear un elemento para el localStorage.
- ☐ También debes saber cómo añadir un elemento a una lista desordenada en HTML.
- ☐ Y, finalmente, como ensamblar todas estas piezas del puzle que forma una aplicación web.

Datos en formato JSON de los ciclos de informática:

- SMR – Sistemas microinformáticos y redes
- ASIR – Administración de sistemas informáticos y redes
- DAW – Desarrollo de aplicaciones web
- DAM – Desarrollo de aplicaciones multimedia

## Ejercicio 2 (2 puntos)

Basándote en un formulario de introducción de datos personales, que se proporciona en la web creada para este ejercicio, crea la parte de JavaScript de una aplicación web con (Nombre, Apellidos, DNI, teléfono, e-mail) añade las siguientes funcionalidades. Disponemos de cuatro botones, “Guardar”, “Cancelar”, “Recuperar” y “Borrar almacenamiento”:

- ☐ Botón “Guardar”: Guardará los datos del formulario en el almacenamiento local (localStorage) en formato JSON.
- ☐ Recuerda que debes validar los campos antes de guardarlos.
- ☐ Al guardarlos eliminará los datos que hubieran guardados y guardará los nuevos.

- ☐ Botón “Recuperar”: Recuperará del almacenamiento local los datos almacenados y rellenará el formulario con estos datos. Podremos modificar los que queramos si hiciera falta.
- ☐ Botón “Cancelar”: Limpiará nuestro formulario sin modificar nada del almacenamiento local.
- ☐ Botón “Borrar almacenamiento”: Borrará los datos almacenados localmente.
- ☐ Recuerda que debes realizar el almacenamiento en localStorage y guardando los datos en formato JSON.
- ☐ Cuando acabes y veas que funciona correctamente la aplicación web que acabas de crear. Cierra la página una vez guardados los datos, vuélvela a abrir y prueba de cargar los datos. Describe con tus palabras el comportamiento de la página.

**Importante:** la validación de datos del formulario se hará simplemente comprobando que existe algún dato o no existe. Si sabes hacer algo más, adelante, pero con esa comprobación básica es suficiente.

Este ejercicio está guiado en parte, mucho menos que el ejercicio anterior.

### Ejercicio 3 (1 punto)

- ☐ Consulta la Api de Chuck Norris (<https://api.chucknorris.io/>)
- ☐ Crea un script JS que muestre por consola frases aleatorias de esa API cada 5 segundos.
- ☐ Debes mostrar la frase aleatoria en el intervalo indicado.
- ☐ Debes mostrar la URL de la frase, pero esta no debe ser para cada elemento, es decir, no debe sumarse por cada elemento una URL si no que solamente aparece en la línea de la URL una única URL que va cambiando al cambiar la frase.
- ☐ A la frase o chiste sobre Chuck Norris le ocurre lo mismo que a la URL de la frase, la frase anterior es sustituida por la frase actual cuando haya transcurrido cierto intervalo de tiempo.

### Ejercicio 4 ( puntos)

### Ejercicio 5 (1 punto)

Crea una aplicación web que basándote en la página web que se te proporciona para este ejercicio haga lo siguiente:

- ☐ Al colocar el cursor sobre un párrafo este se resalte (cambiando el fondo y el color de letra).
- ☐ Esto se puede realizar mediante CSS fácilmente pero el ejercicio te pide que lo hagas en JavaScript.
- ☐ Para conseguir este efecto fácilmente con JavaScript solamente debes poner o quitar una de las clases incluidas en el archivo CSS. En concreto la clase “focus”.
- ☐ **En este ejercicio se pueden usar BUCLES, pero solamente en este ejercicio.**

## Ejercicio 6 (1 punto)

Crea una aplicación web que basándote en la página web que se te proporciona para este ejercicio haga lo siguiente:

- ☐ Al pulsar sobre el primer botón los párrafos que ocupan las posiciones impares tendrán un fondo de un color y los párrafos pares de otro color.
- ☐ Al pulsar sobre el otro botón todos los párrafos vuelven al color inicial.
- ☐ Al igual que en otros ejercicios similares a este utiliza para ello las clases CSS que se te proporcionan, en este caso, necesitaras las clases “pares” e “impares”
- ☐ **En este ejercicio se pueden usar BUCLES, pero solamente en este ejercicio.**

## Ejercicio 7 (2 puntos)

Dada la página web en formato HTML incluida en el ejercicio que contiene varios párrafos, contiene también un input de tipo “text” para introducir el id del párrafo a modificar. Queremos conseguir con JavaScript la siguiente funcionalidad:

- ☐ 3 botones:
  - Agrandar: Ampliará el tamaño del párrafo seleccionado en 0.05em hasta un máximo de 2em.
  - Reducir: Reducirá el tamaño del párrafo seleccionado en 0.05em hasta un mínimo de 0.3em.
  - Original: Restablecerá el tamaño al tamaño original.
  - Si aumenta más de 2em mostrara una alerta que diga no puede ser mayor que la cantidad establecida.
  - Si disminuye a menos de 0.3em, igualmente, mostrará una alerta que diga que no puede ser menor que esa cantidad.

## Ejercicio 8 (2 puntos)

Se proporciona una web con dos botones y una lista ordenada. Se pide que se construya la aplicación JavaScript para que se consigan las siguientes funcionalidades:

- ☐ Se mostrarán una lista numerada de elementos.
- ☐ Se incluirán dos botones: añadir y eliminar (se adjunta captura).
- ☐ Al pulsar añadir dará de alta un nuevo elemento en la lista siguiendo el contador.
- ☐ Con la pulsación anteriormente mencionada se solicitará el texto a introducir (si no se introduce texto, deberá indicar que no puede estar el texto vacío y no se creará nada).
- ☐ Al pulsar el botón eliminar eliminará el último nodo.
- ☐ No se podrá quedar la lista vacía.
- ☐ Siempre quedará un nodo en ella (se avisará que no se puede quedar la lista sin datos).



*Ilustración 1: Ejemplo de web para ejercicio 7*