

Examen de Desarrollo de Interfaces

Tema 2. JavaScript

2º DAM

Alumno/a: _____ Nota: _____

Instrucciones generales:

- Duración: **2 horas y 15 minutos**.
- Crea una carpeta llamada **Examen_TuNombre** y dentro un archivo por ejercicio (ej: `ejercicio1.js`, `ejercicio2.html`, etc). Si hay varios archivos organiza en carpetas.
- Entrega en Moodle la carpeta comprimida.

1. Ejercicio 1 (3 puntos)

Implementa los siguientes apartados en el fichero `ejercicio1.js`:

- **a)** Crea una función que elimine los duplicados de un array y retorne un nuevo array. Debes usar un bucle. *(0,5 puntos)*
- **b)** Crea una función que cuente el número de vocales de una frase. *(0,5 puntos)*
- **c)** Crea una función que determine si una cadena es palíndromo, es decir, que se lee igual hacia delante que hacia atrás. *(0,5 puntos)*
- **d)** Crea un array mediante un bucle del 10 al 50 de cinco en cinco y después desestructura el primer elemento y el resto en dos constantes. *(0,25 puntos)*
- **e)** Mediante la función `map`, crea un array con los números del punto d) elevados al cuadrado. Después filtra para quedarte solo con los menores de 1000. Con la función `some` comprueba si algún número es par. *(0,25 puntos)*
- **f)** Crea una copia del array del punto d) mediante el operador `spread` y añade el número 60 al final. *(0,25 puntos)*

- **g)** Crea una función que no sea arrow que tenga el parámetro persona. Pasa un objeto con las propiedades nombre, apellidos, edad. Dentro de la función recorre el objeto mediante un bucle e imprime las keys y sus valores. *(0,5 puntos)*
- **h)** Desestructura el objeto persona del punto g) en dos constantes nombre y apellidos. *(0,25 puntos)*

2. Ejercicio 2 (2,75 puntos)

Implementa una galería de imágenes que muestre 2 imágenes a la vez y que tenga un botón anterior y un botón siguiente para cambiar entre las imágenes. Utiliza las imágenes proporcionadas en la carpeta del examen.

- **a)** Genera con un bucle un array de 20 urls con el siguiente patrón ["zorro_1.jpg", "zorro_2.jpg", ..., "zorro_20.jpg"]. *(0,25 puntos)*
- **b)** El layout está formado por 4 elementos en el siguiente orden: Un botón Anterior, 2 imágenes y un botón Siguiente. En un primer momento se mostrará la imagen de la posición 0 y 1 del array. Al pulsar el botón siguiente se mostrarán las imágenes 1 y 2 y así sucesivamente. *(0,75 puntos)*
- **c)** Implementa el comportamiento del botón Anterior. *(0,5 puntos)*
- **d)** Controla el botón anterior y siguiente en las posiciones extremas para que no se salgan del array. *(0,5 puntos)*
- **e)** Al hacer clic en la imagen se debe mostrar un modal con la imagen en grande y el fondo oscuro. Se proporciona el CSS. *(1 punto)*

3. Ejercicio 3 (4 puntos)

Desarrolla el muro de la red social FaceBug usando json-server como backend. La aplicación gestionará dos endpoints:

- `/posts`: Array de objetos con {id, id_autor, texto, likes}.
- `/users`: Array de objetos con {id, nombre}. (Esta colección es fija y sirve para elegir quién escribe).

Requisitos:

- **a)** Al entrar en la página se debe renderizar cada post mostrando el nombre del autor, el texto, el número de likes y un botón "Me gusta". *(1 punto)*

- **b)** Añade en la parte superior del muro un formulario de “Nuevo Post”, el campo de autor debe ser un desplegable() que muestre los nombres de los usuarios disponibles. Al pulsar “Publicar”, almacena el post en el backend y actualiza el muro. **(1,5 puntos)**
- **c)** Al hacer clic en el botón “Me gusta” de un post, se debe incrementar el número de likes en 1 y actualizar el post en el backend. Realiza una petición PATCH al endpoint /posts/id enviando únicamente el dato de likes. **(1,5 puntos)**