

Apellidos: .....  
Nombre: ..... UVUS: .....



Bases de datos (GE/DGME)


15/07/2024

Final segunda convocatoria - Parte 2 - Control 6

**Ejercicio 1 (2 puntos)** Crear el esqueleto básico de una página `ej01.html`, con cuerpo vacío, y un archivo `cliente.js`, y realizar lo siguiente:

- Crear en el fichero JS una función que reciba un array de números y devuelva un objeto simple de Javascript que incluya tres elementos: uno que indique el nº de elementos del array, y otros dos que devuelvan dos valores lógicos: uno indicando si el array contiene elementos pares y el otro indicando si hay elementos impares. Probarla con algún ejemplo en el propio archivo JS y posteriormente dejar el código de las llamadas comentado, para no interferir en los siguientes apartados.
- Abra en la página HTML un script de JS de modo que, al arrancar la misma, solicite al usuario, mediante un mensaje emergente de captura de datos, un número (que será para que este indique cuántos números aportar a continuación). Acto seguido, deberá pedir de nuevo al usuario tanto números (en sucesivos mensajes) como se haya indicado. Se deberán almacenar todos estos números en un array, e imprimir el mismo por consola, para comprobar que todo ha ido bien. A continuación, deberá realizar una llamada a la función declarada en el archivo JS auxiliar, capturar en una variable el resultado devuelto por la función, y generar una cadena de texto con la información sobre si hay o no elementos pares e impares en el array, para escribirla en la propia página HTML (con la función más simple posible, teniendo en cuenta que estamos en la carga inicial de la página).

**Ejercicio 2 (3,25 puntos)** Crear el esqueleto básico de una página `ej02.html`, con el cuerpo vacío. Querremos tratar en la misma con información sobre la cartelera de los cines de la ciudad, con información de este tipo:




**Bikeriders. La ley del asfalto**

116 min. | EE.UU. | Crimen | +16  
Jodie Comer, Austin Butler, Tom Hardy  
Dir.: Jeff Nichols

**Comprar entradas:**

11:50 12:00 17:45 19:35 20:15 22:10 22:40

FOTOS ▶




**Del revés 2 (Inside Out 2)**

96 min. | EE.UU. | Animación | TP  
Amy Poehler, Maya Hawke, Liza Lapira  
Dir.: Kelsey Mann

**Comprar entradas:**

12:15 12:20 16:50 17:15 17:35 17:45 18:15 18:50  
19:15 19:45 20:15 20:50 21:15 21:45 22:15 22:20  
22:50 23:15

FOTOS ▶



**El bus de la vida**

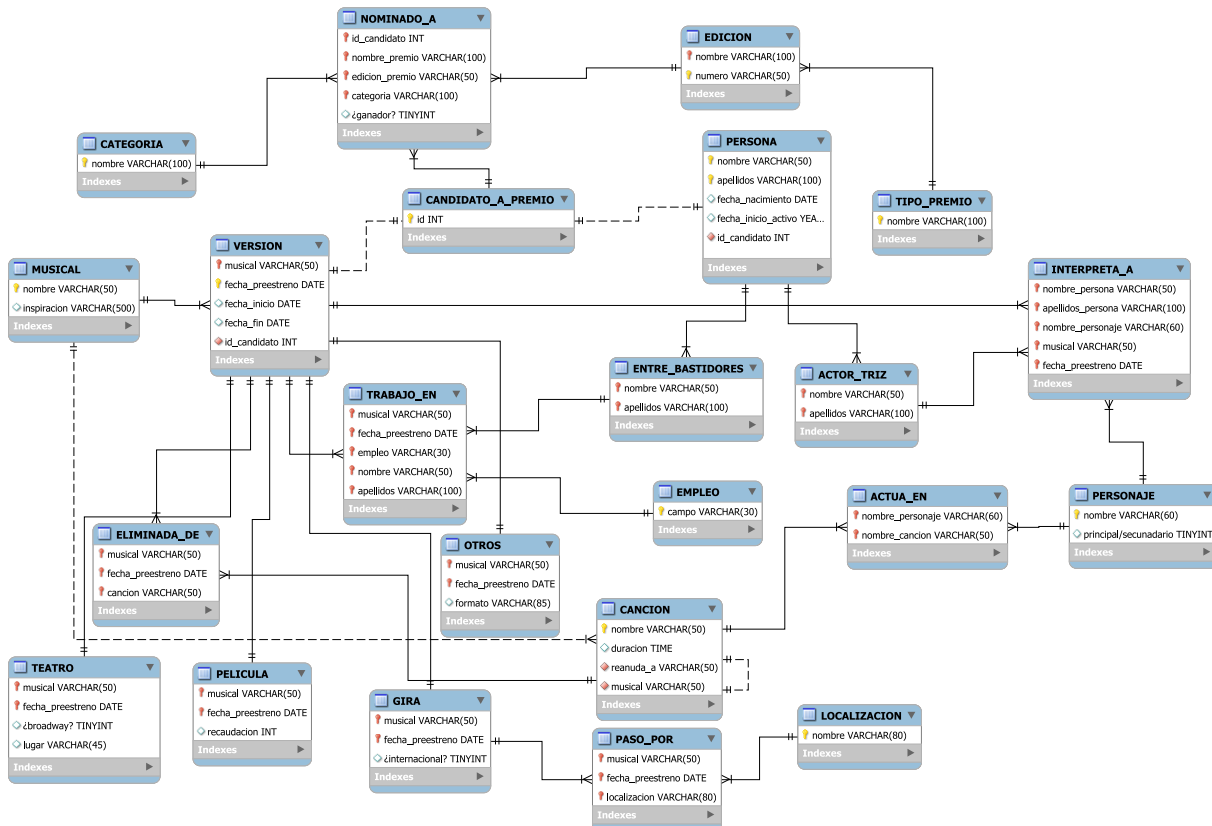
98 min. | España | Drama | +12  
Susana Abaitua, Roberto Álamo, Iria del Río  
Dir.: Ibon Cormenzana

FOTOS ▶

Se pide incorporar el código dinámico necesario para realizar lo siguiente:

- (0,5) Crear una estructura de JavaScript adecuada que almacene, para cada película, los datos de la misma, incluyendo título, duración, país, género, intérpretes, director y horarios. Almacene algunos ejemplos como los de la imagen.
- (1,25) Crear una función que reciba una estructura del tipo anterior, una hora de inicio y un intérprete, y realice lo siguiente: 1) Si no hay ninguna película que empiece a la hora indicada, indicarlo en un mensaje emergente; 2) Si el/la intérprete pasado/a como parámetro no participa en ninguna de las películas, indicar por consola que ya no quieres ir al cine; en ambos casos, se deberá devolver por parte de la función un valor -1; en caso contrario, si se dan ambas situaciones anteriores, devolver el listado de películas que empiezan en el horario indicado en las que interviene el/la intérprete indicado/a (podría ser más de una). Probar la función con algunos valores.
- (1,25) Crear de forma estática un desplegable con nombres de intérpretes y otro con horarios, y asociar manejadores de eventos tales que, al modificar cualquiera de los desplegables, se llame a la función anterior, pasándole los datos tomados de los controles. A continuación, en función de la respuesta de la función, generar en la página los siguientes elementos, haciendo uso de funciones de manipulación del DOM: 1) Si la función devolvió un -1, crear un elemento párrafo en el que indicar que no hay ninguna película que satisfaga todos sus requisitos. Asígnele mediante Js alguna clase que le asocie un estilo especial; 2) Si todo ha ido bien, crear una lista ordenada con un ítem por cada película devuelta por la función.

**Ejercicio 3 (5 puntos)** Cargue en su servidor local MySQL la BD musicales.sql<sup>1</sup>



<sup>1</sup>Agradecimientos a Magdalena Jáñez Vaz, que realizó durante el curso 2021/2022 el genial trabajo de bases de datos que hemos tomado como punto de partida para plantear este ejercicio.

, y realice lo siguiente:

- a) (0,75) Cree un archivo que haga una consulta con todos los campos de la tabla **CANCION**, para aquellas canciones del musical “Hadestown”. Devuelva los registros a un array asociativo e imprima de una vez el array obtenido. Pruébalo en su forma original directamente en el navegador, llamando a php sin parámetros. Cree una página HTML mínima que permita lanzar la consulta desde un botón de la misma, sin dinámica en cliente, de modo que su pulsación sea equivalente a haber lanzado directamente el php desde el navegador, y sin pasarle parámetros.
- b) (1) Prepare una segunda versión de dicha página, pero reemplazando el botón anterior por uno de otro tipo que llame a PHP de forma asíncrona mediante una petición POST, sin pasarle parámetros. En PHP se deberá imprimir una tabla con los datos de cada canción, en lugar de imprimir el array anterior. Recibida la respuesta de PHP, deberá plasmarla tal cual en el cuerpo de la página HTML, al final del cuerpo del documento o en algún contenedor creado a tal efecto, lo que prefiera.
- c) (1,5) Adapte el código PHP para que acepte ahora un parámetro con el nombre de un musical y devuelva las canciones correspondientes a ese musical. Desarrolle en el lado del cliente una función que reciba un nombre de musical y realice una petición al servidor GET de forma no manual, enviándole ese dato. La respuesta recibida de la petición anterior se deberá tratar igual que en el apartado previo. Incorporar un desplegable con nombres de musicales y un botón tal que, al pulsarlo, tome el dato del desplegable y llame a la función anteriormente creada pasándole ese dato.
- d) (1,75) Crear una nueva variante del apartado anterior que cambie tanto la manera de devolver los datos por parte del servidor como su tratamiento posterior en el cliente. Específicamente, PHP ya no deberá generar HTML sino que deberá devolver los datos obtenidos de la consulta como JSON, de modo que en el cliente recibamos una estructura con tres campos: uno que indique mediante un valor cierto que todo ha ido bien, otro que indique el número de datos devuelto y otro que devuelva en sí los resultados de la consulta (mediante un array de objetos, cada uno representando una fila de la consulta). De vuelta en el cliente, se deberá comprobar si todo ha ido bien, en cuyo caso se deberá generar una tabla (actuando sobre el DOM), con la información contenida sobre cada canción, y de manera que, cuando pasemos sobre cada una de las filas, se imprima por consola un mensaje con el título de la canción. Además, deberá garantizarse que, al cargar la página, se visualice por primera vez el resultado de la consulta, devolviendo todas las canciones del musical seleccionado por defecto.

## Notas:

- Escriba su nombre, apellidos y UVUS en la primera hoja del enunciado del examen.
- Incluya sus archivos en una carpeta **final1\_control6\_uvus** (todo en minúsculas), comprímala y súbala a **FileZilla**.