Gestión de Biblioteca 2

Examen desarrollo 2ª evaluación 2DAW 26Feb25

Basándose en el código generado y entregado por el alumno para la aplicación ***Gestión de Biblioteca***, se realizarán distintas tareas que modificarán o añadirán código a lo ya realizado.

Como base para este examen, se utilizará el fichero ***biblioteca_2.html***, que tiene pequeñas adaptaciones para poder realizar las tareas que se definirán a continuación. El alumno ***NO*** podrá modificar este fichero, salvo la parte entre ***<script>*** *y* ***</script>***. *T*odo el tratamiento del UI se realizará mediante Javascript.

Se entregará, además del fichero ***biblioteca_2.html***, un fichero Javascript de nombre ***biblioteca_2.js***, que se integrará dento del fichero HTML.

Tareas a realizar

1 - Importación de datos

Se proporcionan 3 ficheros en formato CSV que contienen:

- Datos de lectores
- Datos de libros
- Datos de préstamos

Antes de interactuar con cualquier otra sección del UI, se deberá realizar la carga de datos iniciales.

Para importar los datos, se seleccionarán los ficheros correspondientes de lectores, libros y préstamos que se entregan como parte de este ejercicio.

Al clickar en el botón Importar, se ejecutará la importación de los 3 ficheros.

No se realizará ninguna comprobación ni corrección de la información a importar; simplemente, se incluirá la información de los ficheros en las estructuras de datos ya creadas por el alumno.

Sólo se modificará, si es necesario, el formato de ***numPrestamo***, ***numSocio*** y/o ***codLibro*** para que sea compatible con el formato elegido por el alumno para estas propiedades de acuerdo al código de la actividad ya entregado antes del examen.

Debajo del botón de su sección (***Importar***), una vez clickado, aparecerá un mensaje diciendo si se ha producido algún error en la importación y cuál ha sido (texto en color rojo) o un mensaje diciendo que se ha realizado la importación de forma exitosa (texto en color verde). El mensaje aparecerá cada vez que se pulse el botón ***Importar***; si se vuelve a clickar el botón de esta sección, desaparecerá este mensaje antes de hacer otras tareas relacionadas con el click en tal botón.

2 - Gestión de eventos de entrada-salida del cursor de la sección *Vista Libros*

Se ha incluido en ***biblioteca 2.html*** un elemento ***p*** con ***id="cursor-sobre-vista-libros"***.

- Cuando el cursor entre en la sección ***Vista Libros***, aparecerá en el elemento ***p*** con ***id="cursor-sobre-vista-libros"*** el mensaje: “Entró en Vista Libros”.
- Cuando el cursor salga de la sección ***Vista Libros***, aparecerá en el elemento ***p*** con ***id="cursor-sobre-vista-libros"*** el mensaje: “Salío de Vista Libros”.

Se utilizarán los eventos: ***mouseenter*** (el cursor entra dentro de los límites de un elemento) y ***mouseleave*** (el cursor sale de los límites de un elemento).

3 - Tamaño y lenguaje de la ventana del navegador

Se ha incluido en ***biblioteca 2.html*** un elemento ***p*** con ***id="propiedades-navegador"***.

Se incluirá en este elemento el siguiente mensaje:

> Código del lenguaje: xx_XX.
>
> Alto de pantalla: xxxx pixels.
>
> Ancho de pantalla: xxxx pixels.

Para ello, se utilizarán las propiedades del BOM que correspondan para que la información del mensaje anterior señale:

- El código de lenguaje configurado en el navegador
- La altura de la pantalla del usuario completa
- La anchura de la pantalla del usuario completa

4 - Nuevo formato de teléfono del lector

En la especificación sobre la que ha trabajado el alumno, el teléfono del usuario (propiedad ***telefono***) tiene el siguiente formato:

> Texto. Contendrá 9 cifras.

Cambiamos el formato a:

> Texto. Contendrá 9 cifras. Podrá empezar o no por “+34 ” (caracteres “+34” seguido por un espacio).
>
> Ejemplos del nuevo formato: “+34 687339182”, “687339182”.

Realizar los cambios necesarios para adaptar el código a este nuevo formato.

5 - Redefinición de la estructura de datos con el listado de libros

Se creará una ***NUEVA*** estructura que contendrá el listado de todos los libros (***no se cambia la actual que ha definido en la actividad el alumno***). Se creará teniendo en cuenta lo siguiente:

- Si el alumno ha definido en la actividad entregada un ***array de objetos*** como estructura para la lista de libros: Se desarrollará una ***función*** llamada ***crearNuevaListaLibros*** que, teniendo como argumento la estructura que ya tiene creada el alumno, devuelva una ***nueva*** estructura llamada ***nuevoObjetoListaLibros*** que será un ***objeto de objetos***
- Si el alumno ha definido en la actividad entregada un ***objeto de objetos*** como estructura para la lista de libros: Se desarrollará una ***función*** llamada ***crearNuevaListaLibros*** que, teniendo como argumento la estructura que ya tiene creada el alumno, devuelva una ***nueva*** estructura llamada ***nuevoObjetoListaLibros*** que será un ***array de objetos***

6 – Modificación de la sección Vista Libros para incluir filtrado

En el código HTML de la sección Vista Libros se ha incluido un elemento ***<input type="text"> (id="vista-libros-incluye")***, donde el usuario puede introducir (o no) caracteres para visualizar el listado de libros cuyo título incluya los caracteres seleccionados. Estos caracteres pueden encontrarse en cualquier posición del string de la propiedad ***titulo*** de cada libro.

La lista con los libros se actualizará con el filtro al clickar en el botón ***Actualizar Libros*** (***id="vista-libros-boton"***)

Se propone utilizar el método ***includes()*** de string, aunque el alumno podrá utilizar otros mecanismos que considere adecuados.

7 - Modificación de la función solicitudPrestamo() utilizando promesas

La especificación para la función ***solicitudPrestamo()*** dice:

> Un préstamo se solicita introduciendo los datos ***numSocio*** y ***codLibro***. Se creará un nuevo préstamo, comprobándose antes si es posible realizarlo. Condiciones para realizar el préstamo: tanto el lector como el libro existen y no están dados de baja; hay un número de ejemplares mayor que **0** de dicho libro.
>
> Entrada y salida de datos:
>
> - Entrada de datos: A través del navegador
> - Salida de mensajes: A través del navegador según lo especificado en la sección correspondiente del UI

Se rediseñará esta función utilizando ***Promesas***.

solicitudPrestamo() contendrá 3 funciones, cada una de ellas con una promesa. Se tendrá que cumplir una promesa para poder pasar a la siguiente. Estas funciones se llamarán desde ***solicitudPrestamo()*** de forma asíncrona.

Estas funciones realizarán las siguientes tareas:

- ***comprobarLibro()***: Comprobará que el libro existe, que no está dado de baja y que hay un número de ejemplares mayor que 0
- ***comprobarLector()***: Comprobar que el lector existe y que no está dado de baja
- ***generarPrestamo()***: Creará un préstamo

Cada vez que se haga click en el botón ***Préstamo***, y como resultado del procesamiento del préstamo ,aparecerá, ***debajo de los botones*** de la sección, un mensaje en un elemento ***p*** diciendo si se ha producido algún error y cuál ha sido (texto en color rojo) o un mensaje diciendo que se ha prestado de forma exitosa y que incluya ***numPrestamo*** y ***fechaPrestamo*** (texto en color verde); si se vuelve a clickar cualquier botón de esta sección, desaparecerá este mensaje antes de hacer otras tareas relacionadas con tal clickado.

El alumno podrá utilizar cualquiera de las modalidades vistas en clase para el manejo de promesas:

- Métodos then(), catch(), finally()
- Async/Await

Normas

- *La base del desarrollo será la actividad ya realizada por el alumno y entregada como paso previo a la realización de este examen. Las especificaciones sobre las que se basó este desarrollo previo permanecen, de acuerdo al documento* ***Especificaciones actividad - Gestión de biblioteca.pdf****. Sólo cambiará lo especificado en el presente documento para poder realizar las nuevas funcionalidades y modificaciones en las tareas definidas anteriormente para este examen*
- *Se comentará el código convenientemente, pensando en una persona que pudiera tener que trabajar en este desarrollo varios meses después de entregarlo*
- *Se utilizará código lo más declarativo posible (y lo menos imperativo posible), utilizando funcionalidades específicas y avanzadas de Javascript*
- *El alumno tendrá que tomar decisiones técnicas respecto a lo no especificado en este documento. Lo hará de forma lógica y coherente, explicándolo en los comentarios del código*
- *Los mensajes que se presenten por consola tendrán un formato adecuado, legible, entendible y ordenado*
- *Adicionalmente a las funciones anteriores, el alumno podrá crear otras funciones a su conveniencia. Se penalizará desarrollar varias veces un código para realizar una misma tarea o muy parecida*