Gestión de Biblioteca

Esta actividad consiste en realizar parcialmente una aplicación para la gestión de una biblioteca.
Se desarrollará toda la lógica que se pide y sólo parte del interfaz de usuario recogida en ***biblioteca.html***.

Especificaciones

Estructura de datos y lógica de gestión

Se tendrán en cuenta los siguientes tipos de elementos en cuanto a la creación de una estructura de datos. Cada elemento será un objeto. El conjunto de elementos de tipología (***Lectores***, ***Libros***, …) se podrá estructurar como considere el alumno, pero basándose o en arrays o en objetos

Los tipos de elementos son:

- ***Lector***: Cada lector tendrá las siguientes propiedades:
  - ***numSocio*** - Texto. Formato a definir por el alumno
  - ***nombre*** - Texto. Podrá ser simple (una palabra) o compuesto (2 palabras) Se admiten todas las letras del alfabeto español con/sin acento en vocales y el signo "-"
  - ***apellido*** - Texto. Podrá ser simple (una palabra) o compuesto (2 palabras) Se admiten todas las letras del alfabeto español con/sin acento en vocales y el signo "-"
  - ***telefono*** – Texto. Contendrá 9 cifras
  - ***email*** – El alumno buscará y aplicará las reglas de un email válido, considerando que sólo se admiten los dominios: ***es***, ***com***, ***net***, ***eu***, ***org***

  La gestión de ***numSocio*** se realizará dentro del código de forma automática; cada alumno puede aplicar la política de numeración que considere, siempre que sea lógica y coherente. En los comentarios del código se explicará dicha política.
  Si se da de baja un usuario, se dispondrá de las propiedades:
  - ***bajaLector*** - Booleano
  - ***fechaBaja*** – Texto. Formato: ***dd/mm/aaaa***

  Las anteriores propiedades se podrán añadir al instanciar cada objeto o en el momento de tramitar la baja de un lector. Se conservará toda la información del lector a pesar de la baja para poder analizar los préstamos tanto antiguos como pendientes.
- ***Libro***: Habrá un listado de los libros disponibles, que se restringirá en este ejercicio a los clasificados como manga. De cada libro, tendrá la siguiente información (propiedades):
  - ***codLibro*** - Texto. Formato a definir por el alumno
  - ***isbn*** – Texto. Contendrá 13 caracteres
  - ***autor*** - Texto. Podrá ser simple (una palabra) o compuesto (2 palabras); Se admiten todas las letras del alfabeto español con/sin acento en vocales y el signo -
  - ***titulo*** - Podrá ser simple (una palabra) o compuesto (varias palabras). Se admiten todas las letras del alfabeto español con/sin acento en vocales, los números y los siguientes caracteres imprimibles: ***- _ ¡ ! @ # $ % & / ( ) ¿ ? € . : , ;***
  - ***editorial*** - Texto. Podrá ser simple (una palabra) o compuesto (2 palabras); se admiten todas las letras del alfabeto español con sus acentos y el signo -
  - ***ejemplares*** – Número entero. Valores posibles: del 0 al 9

La gestión de ***codLibro*** se realizará dentro del código de forma automática; cada alumno puede aplicar la política de numeración que considere, siempre que sea lógica y coherente. En los comentarios del código se explicará dicha política.
Si se da de baja un usuario, se dispondrá de las propiedades

- ***bajaLibro*** - Booleano
- ***fechaBaja*** - Texto. Formato: ***dd/mm/aaaa***

Las anteriores propiedades se podrán añadir al instanciar cada objeto o en el momento de tramitar la baja de un lector. Se conservará toda la información del libro a pesar de la baja para poder analizar los préstamos tanto antiguos como pendientes.

- ***Préstamo***: Habrá un listado con los préstamos de libros que han realizado los lectores de la biblioteca. Contendrá todos los préstamos, tanto activos como de libros ya devueltos. Cada préstamo incluirá la siguiente información (propiedades):
  - ***numPrestamo*** - Texto. Formato a definir por el alumno
  - ***numSocio*** – Definido anteriormente
  - ***codLibro*** - Definido anteriormente
  - ***fechaPrestamo*** - Texto. Formato: ***dd/mm/aaaa***
  - ***fechaDevolucion*** - Texto. Formato: ***dd/mm/aaaa***

  La gestión de ***numPrestamo*** se realizará dentro del código de forma automática; cada alumno puede aplicar la política de numeración que considere, siempre que sea lógica y coherente. En los comentarios del código se explicará dicha política.
- ***Clasificación de los libros***: Sólo tendremos en cuenta los de manga. Todos se encuentran en el ***pasillo*** 7, ***estantería*** 4, ***estante*** 6. Se creará un objeto con las propiedades y valores anteriores.

Se quiere disponer de las siguientes funciones referentes a los elementos anteriores:

- ***Lectores***
  - ***altaLector()***: Se proporcionarán los datos de un nuevo lector, se comprobará que están todos y que son correctos; a continuación, se dará de alta. Se realizará la verificación del formato de cada dato introducido.
    La función devolverá:
    numSocio – Si el alta ha sido correcta
    F – Si falta algún dato
    V – Si hay algún dato con formato incorrecto
    Entrada y salida de datos:
    - Entrada de datos: A través del método ***prompt()***
    - Salida de datos y errores: A través del método ***console.log()***
  - ***bajaLector()***: La baja se realizará añadiendo y/o actualizando propiedades de los datos de baja en la información del lector al que se da de baja. Contendrá: ***bajaLector*** (booleano) y ***fechaBaja*** (día, mes y año). Se conservará toda la información del lector a pesar de la baja para poder analizar los préstamos tanto antiguos como pendientes.
    La función devolverá:
    fechaBaja – Si el alta ha sido correcta
    E – Si se ha producido algún error
    - Entrada de datos: A través del método ***prompt()***
    - Salida de datos y errores: A través del método ***console.log()***

- ***modifLector()***: Se utilizará para modificar cualquier dato de un libro, sea porque ha cambiado o porque es incorrecto. Se preguntará por el ***numSocio*** sobre el que se realizará la modificación, se preguntará el dato a modificar, se comprobará si el formato es válido y se actualizará (en caso de que sea válido). Se generará un mensaje en la consola diciendo si la modificación ha sido exitosa o el error que se produzca.
    - Entrada de datos: A través del método ***prompt()***
    - Salida de datos y errores: A través del método ***console.log()***
- ***verificarEmail():*** Verificará si el formato de un email es válido. Argumento: dirección de email; devuelve: ***true***/***false***. Se llamará desde otra parte del código
- ***verificarTelefono():*** Verificará si el formato de un email es válido. Argumento: número de teléfono; devuelve: ***true***/***false***. Se llamará desde otra parte del código
- ***comprobarEmails()***: Se comprobará si las direcciones de email de TODOS los lectores tienen un formato correcto y se dará un listado de los que NO son válidos (***numSocio + nombre + apellido + telefono + email***.). Se sugiere realizar la función ***verificarEmail*** para verificar el formato de una dirección de email.
    - Entrada de datos: A través del método ***prompt()***
    - Salida de datos y errores: A través del método ***console.log(***
- ***comprobarTelefonos()***: Se comprobará si los números de teléfono de TODOS los lectores tienen un formato correcto y se dará un listado de los que no son válidos (***numSocio + nombre + apellido + telefono + email***.). Se sugiere realizar la función ***verificarTelefono*** para verificar el formato de un teléfono.
    - Entrada de datos: A través del método ***prompt()***
    - Salida de datos y errores: A través del método ***console.log()***

- ***Libros***
    - ***altaLibro()***: Se preguntará por los datos de un nuevo libro, se comprobará que están todos y que son correctos; a continuación, se dará de alta. Si el libro ya existe, se generará un mensaje de error advirtiéndolo (ver último párrafo de esta función). Se realizará la verificación del formato de cada dato introducido.
      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
    - ***bajaLibro()***: La baja se realizará añadiendo y/o actualizando propiedades de los datos de baja en la información del libro al que se da de baja. Contendrá: ***bajaLibro*** (booleano) y ***fechaBaja*** (día, mes y año). Se conservará toda la información del lector a pesar de la baja para poder analizar los préstamos tanto antiguos como pendientes. Se generará un mensaje en la consola diciendo si la modificación ha sido exitosa o el error que se produzca.
        - Entrada de datos: A través del método ***prompt()***
        - Salida de datos y errores: A través del método ***console.log()***
    - ***modifLibro()***: Se utilizará para modificar cualquier dato de un libro, sea porque ha cambiado o porque es incorrecto. Se preguntará por el ***codLibro*** sobre el que se realizará la modificación, se preguntará el dato a modificar, se comprobará si el dato introducido tiene formato válido y se actualizará (en caso de que sea válido). Se generará un mensaje en la consola diciendo si la modificación ha sido exitosa o el error que se produzca.

        - Entrada de datos: A través del método ***prompt()***
        - Salida de datos y errores: A través del método ***console.log()***
    - ***hayLibro()***: se podrá buscar por ***codLibro*** o por ***isbn***. El libro no deberá estar dado de baja. Cada alumno la podrá implementar a su gusto. Si está/no está en la lista de libros, se devolverá: ***true***/***false***. Se llamará desde otra parte del código
    - ***prestamoLibro()***: Se comprobará si existe el libro (por ***codLibro***) y si hay ejemplares disponibles; en caso afirmativo, se actualizarán los datos del libro reflejando el préstamo y se devolverá ***true***; en caso contrario, no se podrá prestar el libro y de devolverá ***false***. Se llamará desde otra parte del código o desde otra función
    - ***devolucionLibro()***: Se comprobará que existe el libro (por ***codLibro***); en caso afirmativo, se aumentará el número de ejemplares disponibles y se devolverá ***true***. Si no existe el libro, se devolverá ***false.*** Se llamará desde otra parte del código
    - ***dondeLibro()***: Se preguntará por un ***codLibro*** y se devolverá por consola un mensaje conteniendo: pasillo, estantería y estante.
        - Entrada de datos: A través del método ***prompt()***
        - Salida de datos y errores: A través del método ***console.log()***
- ***Préstamos***
    - ***listadoTotalPrestamos()***: Se elaborará un listado con todos los préstamos, tanto vivos como devueltos. Se proporcionará un listado con toda la información de cada préstamo en una línea. Si hay algún error, se generará un mensaje de error en la consola.
        - Entrada de datos: A través del método ***prompt()***
        - Salida de datos y errores: A través del método ***console.log()***
    - ***listadoPrestamosVivos()***: Se elaborará un listado de todos los préstamos vivos (no devueltos). Se proporcionará un listado con toda la información de cada préstamo en una línea. Si hay algún error, se generará un mensaje de error en la consola.
        - Entrada de datos: A través del método ***prompt()***
        - Salida de datos y errores: A través del método ***console.log()***
    - ***solicitudPrestamo()***: Un préstamo se solicita introduciendo los datos ***numSocio*** y ***codLibro***. Se creará un nuevo préstamo se comprobará 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
    - ***devolucionPrestamo()***: Un préstamo se solicita introduciendo los datos ***numSocio*** y ***codLibro***. Se actualizará el préstamo (***fechaDevolucion***) si es posible realizarlo. Condiciones para realizar la devolución: tanto el lector como el socio existen, aunque estén dados de baja; además, existe un ***numPrestamo*** y en ese préstamo coinciden tanto el ***numSocio*** como el ***codLibro***.

      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

Interfaz de usuario

En algunas de las funciones anteriores, se ha especificado cómo se llevará a cabo su interacción con el navegador. En este apartado, se detalla todo el UI.

Se provee a los alumnos del fichero ***biblioteca.html*** con el código sobre el que se desarrolla el UI. De este fichero, sólo se puede modificar la parte entre <***script***> y <***/script***>.

Todo el tratamiento del UI se realizará a través de Javascript, utilizando las estructuras de datos y la lógica presentadas anteriormente en este documento.

El UI se compone de las siguientes secciones:

- ***Importar***: Se proporcionan los ficheros en formato CSV que contienen la carga inicial de datos de lectores y de libros. 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 y de libros. Al clickar en el botón ***Importar***, se ejecutará la importación. No se realizará ninguna comprobación ni corrección de la información; simplemente, se incluirá la información de los ficheros en las estructuras de datos ya creadas.

  Debajo del botón de su sección (***Importar***), una vez clickado, aparecerá un mensaje diciendo si se ha producido algún error y cuál ha sido (texto en color rojo) o un mensaje diciendo que se ha realizado la importación de forma exitosa. 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.
- ***Vista libros***: Se presentará la lista de todos los libros que no estén dados de baja, con los campos que se incluyen en ***biblioteca.html***. Se presentará en formato tabla con cada campo debajo del título correspondiente. Se actualizará cada vez que se presione el botón ***Actualizar libros***.

  Añadir paddings para una buena presentación y con outline para cada campo. El fondo de la línea con los títulos de los campos será de color ***#BB61F0***. El fondo de cada campo será de color ***#C398EB***. Texto en toda la tabla en color #***1B1B1B***.

- ***Vista lectores***: Se presentará la lista de todos los lectores que no estén dados de baja, con los campos que se incluyen en ***biblioteca.html***. Se presentará en formato tabla con cada campo debajo del título correspondiente. Se actualizará cada vez que se presione el botón ***Actualizar lectores***. Añadir paddings para que tenga una buena presentación y con outline en cada campo.

  Añadir paddings para una buena presentación y con outline para cada campo. El fondo de la línea con los títulos de los campos será de color ***#BB61F0***. El fondo de cada campo será de color ***#C398EB***. Si el teléfono del lector o su dirección de email no tiene el formato correcto, en este caso, cada campo de email o teléfono erróneo será de color ***#EA9E90***. Texto en toda la tabla en color #***1B1B1B***.

- ***Alta libro***: Se introducirán TODOS los campos del formulario y se verificará cada dato según formato definido en cada propiedad. Se utilizará la función ***altaLibro()***, junto con el código adicional que se requiera.

Cada vez que se clicke el botón de su sección (***Alta***), aparecerá un mensaje (con texto en color rojo) debajo del botón diciendo si se ha producido algún error y cuál ha sido o un mensaje diciendo que se ha dado de alta de forma exitosa y conteniendo ***codLibro***; si se vuelve a clickar en el botón para dar de alta el libro, desaparecerá este mensaje antes de hacer otras tareas. El mensaje anterior desaparecerá cada vez que se pulse el botón ***Alta*** y aparecerá el nuevo mensaje.

- ***Devolución/Préstamo de libros***: A través de esta sección, se podrá tanto devolver un libro prestado como solicitar el préstamo de un libro. Se utilizarán, aparte del código adicional que se requiera, las funciones ***solicitudPrestamo()*** y ***devolucionPrestamo()***.

  Se ejecutará la devolución o el préstamo al pulsar el botón correspondiente en esta sección.

  Cada vez que se clicke el botón ***Préstamo***, y como resultado del procesamiento de la devolución, aparecerá debajo de los botones un mensaje 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 rojo); si se vuelve a clickar cualquier botón de esta sección, desaparecerá este mensaje antes de hacer otras tareas.
  Cada vez que se clicke el botón ***Devolución***, y como resultado del procesamiento del préstamo, aparecerá debajo de los botones un mensaje diciendo si se ha producido algún error y cuál ha sido (texto en color rojo) o un mensaje diciendo que se ha devuelto de forma exitosa y que incluya ***numPrestamo*** y ***fechaDevolucion*** (texto en color rojo); si se vuelve a clickar cualquier botón de esta sección, desaparecerá este mensaje antes de hacer otras tareas.

Notas

- Las funciones especificadas anteriormente se desarrollarán obligatoriamente. El alumno elegirá entre hacerlas como funciones independientes, como métodos dentro de un objeto, agrupando funciones en prototipos o de otra manera que explicará el alumno en los comentarios del código
- 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
- Los mensajes que se presenten por consola tendrán un formato adecuado, legible, entendible y ordenado
- 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
- Sólo se tendrán en cuenta las especificaciones recogidas en este documento. Se descartarán las proporcionadas anteriormente
- El fichero ***biblioteca.html*** es el que contiene el esqueleto del UI. Se utilizará la última versión, que es el que acompaña al presente documento dentro de la actividad del Aula Virtual . No se modificará el código del fichero ***biblioteca.html***, salvo entre ***<script>*** y ***</script>***
- 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