



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
EXÁMEN PROGRAMACIÓN WEB

Escenario: Una biblioteca digital necesita mejorar su sistema de gestión de préstamos y devoluciones de libros. Actualmente, los usuarios pueden buscar libros, pero no hay una forma eficiente de reservarlos, devolverlos o verificar su disponibilidad en tiempo real. Además, el sistema no cuenta con alertas para recordar a los usuarios la fecha de devolución ni notificaciones sobre la disponibilidad de libros reservados

Se requiere que los estudiantes analicen y propongan una solución para mejorar el sistema de gestión de la biblioteca digital, aplicando los conceptos de JavaScript aprendidos en clase. En su análisis, deberán considerar lo siguiente:

1. Estructura del sistema de préstamos:

El sistema de préstamos organiza los libros en dos estructuras principales:

- Un array llamado libros, que contiene todos los libros registrados, tanto disponibles como prestados.
- Un array secundario llamado librosPrestados, que almacena exclusivamente los libros que han sido prestados.

La manipulación de estos arrays se realiza utilizando métodos integrados de JavaScript como:

- **push:** Para agregar libros al final del array librosPrestados cuando un libro es prestado.
- **splice:** Para eliminar un libro de librosPrestados y regresarlo al array principal libros cuando se realiza una devolución.

2. Filtrado y búsquedas dinámicas:

El sistema utiliza el método filter de JavaScript para realizar búsquedas dinámicas en el array de libros. Este enfoque permite filtrar libros basándose en atributos específicos como el título, el autor o el género.

- **Implementación en el código:** La función filtrarLibros recibe dos parámetros: el criterio de búsqueda (como título, autor o genero) y el valor que se desea buscar. Esta función devuelve un array de libros que cumplen con el criterio especificado.



Ventajas de este enfoque:

- Flexibilidad para realizar búsquedas en múltiples campos.
- Permite realizar búsquedas dinámicas, como cuando el usuario escribe en un campo de texto.

3. Interacción con el usuario:

El sistema utiliza métodos de manipulación del DOM para actualizar dinámicamente las listas de libros en la interfaz de usuario. Estas listas se muestran en dos secciones: libros disponibles y libros prestados.

- **Técnicas empleadas:**
 - **getElementById:** Se utiliza para acceder a los elementos `` donde se insertan los libros.
 - **innerHTML:** Se emplea para limpiar y actualizar el contenido de las listas antes de agregar nuevos elementos.
 - **createElement y appendChild:** Se usan para crear elementos `` que representan los libros y añadirlos dinámicamente a las listas.

4. Alertas y recordatorios:

El sistema incluye una funcionalidad para enviar recordatorios automáticos sobre la devolución de libros prestados. Esto se logra mediante el uso de `setInterval`.

- **Funcionamiento:**
 - La función `enviarRecordatorios` recorre el array `librosPrestados` y genera un mensaje para cada libro, recordando al usuario que debe devolverlo.
 - `setInterval` se configura para ejecutar esta función automáticamente cada 10 segundos.

Ventaja de esta implementación:

- Automatiza los recordatorios, simulando un sistema de notificaciones en tiempo real.

5. Eventos y usabilidad:

El sistema incorpora eventos para mejorar la interacción con el usuario:

- **Eventos utilizados:**
 - **oninput:** Permite activar la búsqueda de libros en tiempo real mientras el usuario escribe en el campo de búsqueda.

6. Funciones avanzadas:

El sistema utiliza funciones avanzadas para gestionar tareas complejas:



- **Funciones anónimas:** Son empleadas en métodos como `setTimeout` para encapsular lógica específica.
- **Funciones `async/await`:** Facilitan la gestión de promesas, especialmente en operaciones como la reserva y devolución de libros.

7. Simulación de procesos asíncronos:

El sistema utiliza promesas y `setTimeout` para simular tiempos de espera en operaciones como la reserva y devolución de libros:

- **Reservar un libro:**
 - La función `prestarLibro` verifica la disponibilidad del libro y, si está disponible, lo mueve al array `librosPrestados`.