



UNIVERSIDAD NACIONAL DEL ALTIPLANO

**FACULTAD DE
INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



TRABAJO ENCARGADO

CURSO:

ESTRUCTURAS DE DATOS AVANZADOS

DOCENTE:

ING.FREDY COLLANQUI MARTÍNEZ

PRESENTADO POR:

BELTRAN EDWIN MAMANI MAMANI

PUNO- PERU

2024

FUNCIONES CONTÍNUAS

SEMANA 16 (julio 15, 17 ,19)

SISTEMA DE BIBLIOTECA CON BST

Implementa un sistema básico de gestión de biblioteca utilizando un Árbol Binario de Búsqueda BST.

Permite buscar, insertar y eliminar libros en un BST a través de una interfaz web. La implementación está dividida en dos partes principales: el archivo HTML que define la estructura y el archivo JavaScript que gestiona la lógica del BST y las interacciones del usuario.

Estructura del Código

```
1      <html lang= "es" >
2      <head>
3          <meta charset= "UTF-8" >
4          <meta name= "viewport" content= "width=device-width, initial-scale=1.0" >
5          <title> Sistema de Biblioteca con BST </title>
6          <link href=
7      "https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap" rel=
8      "stylesheet" >
9          <link rel= "stylesheet" href= "styles.css" >
10         <style>
11             cuerpo { familia-fuente : Arial , sans-serif ; ancho-máximo : 800px ; margen : 0
12 automático ; relleno : 20px ; }
13             entrada , botón { margen : 5px 0 ; }
14             #resultado { margen-superior : 20px ; }
15         </style>
dieciséis </head>
17     <body>
18         <h1> Sistema de Biblioteca con BST </h1>
19
20         <h2> Buscar Libro </h2>
21         <input type= "number" id= "buscarId" placeholder= "ID del libro" deshabilitado >
22         <button onclick= "buscarLibro()" deshabilitado > Buscar Libro </button>
23
24         <h2> Insertar libro </h2>
```

```

25     <input type= "number" id= "insertId" placeholder= "ID del libro" deshabilitado >
26     <input type= "text" id= "insertTitulo" placeholder= "Título del libro" deshabilitado >
27     <input type= "text" id= "insertAutor" placeholder= "Autor del libro" deshabilitado >
28     <button onclick= "insertarLibro()" deshabilitado > Insertar libro </button>
29
30     <h2> Eliminar Libro </h2>
31     <input type= "number" id= "eliminarId" placeholder= "ID del libro" deshabilitado >
32     <button onclick= "eliminarLibro()" deshabilitado > Eliminar Libro </button>
33
34     <div id= "resultado" ></div>
35
36     <script>
37         // Código JavaScript aquí
38     </script>
39 </body>
40 </html>

```

Descripción:

- **<head>**: Define los metadatos de la página, incluyendo el juego de caracteres, la vista adaptable, el título, y los enlaces a fuentes y hojas de estilo.
- **<body>**: Contiene la estructura principal de la interfaz con campos de entrada para buscar, insertar y eliminar libros, y un área para mostrar resultados.

JAVASCRIPT: DEFINICIÓN DE CLASES Y FUNCIONES

Definición de Clases:

```

1     class Nodo {
2         constructor(libro) {
3             este .libro = libro;
4             esto .izquierda = null ;
5             este .derecha = null ;
6         }
7     }
8
9     class ArbolBinarioBusqueda {
10         constructor() {
11             este .raiz = null ;
12         }
13
14         insertar(libro) {
15             libro.id = parseInt (libro.id);
dieciséis             este .raiz = este ._insertarRecurso(este .raiz, libro);
17         }
18
19         _insertarRecurso(nodo, libro) {

```

```

20         if (nodo === null ) {
21             return nuevo Nodo(libro);
22         }
23
24         si (libro.id < nodo.libro.id) {
25             nodo.izquierda = this ._insertarRecursivo(nodo.izquierda, libro);
26         } de lo contrario si (libro.id > nodo.libro.id) {
27             nodo.derecha = this ._insertarRecursivo(nodo.derecha, libro);
28         }
29
30         retorno nodo;
31     }
32
33     buscar(id) {
34         devuelve este ._buscarRecursivo( este .raiz, parseInt (id));
35     }
36
37     _buscarRecursivo(nodo, id) {
38         if (nodo === null || nodo.libro.id === id) {
39             return nodo ? nodo.libro : nulo ;
40         }
41
42         if (id < nodo.libro.id) {
43             return this ._buscarRecursivo(nodo.izquierda, id);
44         }
45         devuelve este ._buscarRecursivo(nodo.derecha, id);
46     }
47
48     eliminar(id) {
49         este .raiz = este ._eliminarRecursivo( este .raiz, parseInt (id));
50     }
51
52     _eliminarRecursivo(nodo, id) {
53         si (nodo === null ) {
54             devolver null ;
55         }
56
57         si (id < nodo.libro.id) {
58             nodo.izquierda = this ._eliminarRecursivo(nodo.izquierda, id);
59         } de lo contrario si (id > nodo.libro.id) {
60             nodo.derecha = this ._eliminarRecursivo(nodo.derecha, id);
61         } else {
62             if (nodo.izquierda === null ) {
63                 return nodo.derecha;
64             } else if (nodo.derecha === null ) {
65                 return nodo.izquierda;
66             }
67
68             nodo.libro = this ._encontrarMinimo(nodo.derecha);
69             nodo.derecha = this ._eliminarRecursivo(nodo.derecha, nodo.libro.id);
70         }
71
72         retorno nodo;
73     }
74
75     _encontrarMinimo(nodo) {
76         let actual = nodo;

```

```

77         while (actual.izquierda !== nulo) {
78             actual = actual.izquierda;
79         }
80         devuelve actual.libro;
81     }
82 }

```

Descripción:

- **Nodo:** Representa un nodo del árbol con un libro y punteros a nodos izquierdo y derecho.
- **ArbolBinarioBusqueda:** Implementa las operaciones básicas del BST (insertar, buscar, eliminar), utilizando métodos recursivos.

Interacción con la Interfaz de Usuario:

```

1     const bst = new ArbolBinarioBusqueda();
2
3     fetch( 'obtener_libros.php' )
4         .then(respuesta => respuesta.json())
5         .then(datos => {
6             datos.forEach(libro => {
7                 libro.id = parseInt( libro.id );
8                 bst.insertar(libro);
9             });
10            console.log( 'Libros cargados en BST:', data.length );
11            habilitarUI();
12        })
13        .catch( error => console.error( 'Error:', error ) );
14
15    función habilitarUI() {
dieciséis    documento . getElementById( 'buscarId' ). disabled = false ; documento . getElementById(
17    'insertId' ). disabled = false ;
18    documento . getElementById( 'insertTitulo' ) .
19    disabled = false ; documento .
20    getElementById ( 'insertAutor' ). disabled = false ;
21    documento . getElementById( 'eliminarId' ). disabled = false ;
22    documento . querySelector ( ' button[onclick="buscarLibro()"]' ). disabled = false ;
23    documento . querySelector( 'button[onclick="insertarLibro()"]' ). disabled = false ;
24    documento . querySelector( 'button[onclick="eliminarLibro()"]' ). disabled = false ;
25    }
26
27    function mostrarResultado(mensaje) {
28        document .getElementById( 'resultado' ).innerHTML = mensaje;
29    }
30
31    function buscarLibro() {
32        const id = parseInt ( document .getElementById( 'buscarId' ).value );
33        const libro = bst.buscar(id);

```

```

34
35     si (libro) {
36         mostrarResultado( Libro encontrado : ID ${libro.id}, Título : ${libro.titulo}, Autor : $
37         {libro.autor} );
38     } demás {
39         mostrarResultado( No se encontró ningún libro con el ID ${id} );
40     }
41 }
42
43 función insertarLibro() {
44     const id = parseInt ( documento . getElementById( 'insertId' ). value);
45     const titulo = documento . getElementById( 'insertTitulo' ). value;
46     const autor = documento . getElementById( 'insertAutor' ). value;
47
48     if(id && titulo && autor) {
49         const libro = {id, titulo, autor};
50         bst.insertar(libro);
51         mostrarResultado( Libro insertado : ID ${id}, Título : $ { titulo }, Autor : ${autor} );
52     } demás {
53         mostrarResultado( 'Error: Todos los campos son obligatorios' );
54     }
55 }
56
57 función eliminarLibro() {
58     const id = parseInt ( document . getElementById( 'eliminarId' ). value);
59     const libroAEliminar = bst. buscar(id);
60     if(libroAEliminar) {
61         bst.eliminar(id);
62         mostrarResultado( Libro con ID ${id} eliminado correctamente );
63     } demás {
64         mostrarResultado( No se encontró ningún libro con el ID ${id} );
65     }
66 }

```

- **Inicialización del BST:** Se crea una instancia del BST y se cargan los libros desde un archivo PHP (obtener_libros.php). La interfaz de usuario se habilita después de que los libros se cargan correctamente.
- **Funciones de Interacción:**
 - ❖ **habilitarUI:** Activa los elementos de entrada y botones una vez que los libros se cargan.
 - ❖ **mostrarResultado:** Muestra mensajes en el área de resultados.
 - ❖ **buscarLibro, insertarLibro, eliminarLibro:** Manejan la búsqueda, inserción y eliminación de libros en el BST, respectivamente.

Este código implementa un sistema de gestión de biblioteca basado en un Árbol Binario de Búsqueda. Utiliza HTML para la estructura de la interfaz y JavaScript para manejar la lógica del árbol y las interacciones del usuario.

La separación entre la lógica de negocio y la presentación permite una gestión eficiente de los libros y una interfaz web intuitiva.

Sistema de Biblioteca con BST

Buscar Libro

<input type="text" value="ID del libro"/>	<input type="button" value="Buscar Libro"/>
---	---

Insertar Libro

<input type="text" value="ID del libro"/>	<input type="text" value="Título del libro"/>	<input type="text" value="Autor del libro"/>	<input type="button" value="Insertar Libro"/>
---	---	--	---

Eliminar Libro

<input type="text" value="ID del libro"/>	<input type="button" value="Eliminar Libro"/>
---	---