

# **CATSBOOKS – PROYECTO 1**

## **MANUAL TÉCNICO**

**PROGRAMA REALIZADO POR EL ALUMNO ALLEN  
GIAN KARLO ROMÁN VÁSQUEZ CARNET 202004745 PARA  
EL CURSO ESTRUCTURAS DE DATOS DE LA ESCUELA DE  
CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA  
DE LA UNIVERSIDAD SAN CARLOS DE GUATEMALA**

## Programa Utilizado

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.



## Lenguaje Utilizado

JavaScript es un robusto lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla.

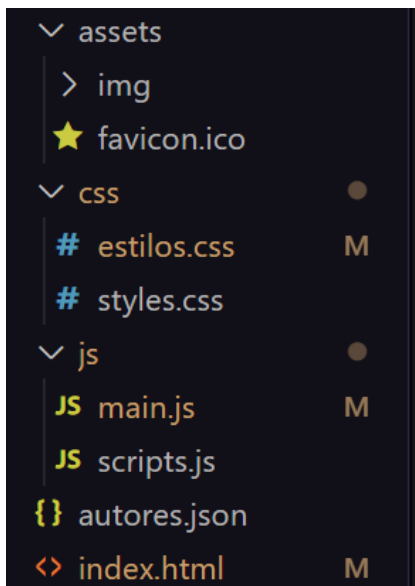


## Código

### Clases

El programa cuenta con tres carpetas, assets para imágenes, css para el entorno gráfico utilizado con archivos tipo .css y js para los archivos de javascript y el index.html que es la página principal, el archivo principal de

main.js cuenta con 17 clases y 21 funciones implementada que serán descritas a continuación de una forma superficial.



## class NodoCola

El NodoCola es donde se encuentra el Nodo de la Lista\_Cola y a su vez su método constructor.

```
//Cola
class NodoCola{
    constructor(libro,cliente){
        this.libro = libro;
        this.cliente = cliente;
        this.codigoCliente = 0;
        this.codigoLibro = null;
        this.siguiente = null;
    }
}
```

## class Lista\_Cola

Lista\_Cola es donde se encuentra la lista que guarda los nodos de NodoCola, se encuentra su constructor y sus métodos para insertar a la Cola o para graficarCola que realiza un gráfico en Graphviz.

```
class Lista_Cola{
    constructor(){
        this.primeros = null;
        this.ultimo = null;
        this.contador = 0;
    }
    > insertar(libro, cliente){ ...
    }
    > graficarCola(){ ...
    }
}
```

## class NodoPila

NodoPila es donde se encuentra el Nodo de la Pila de la Lista\_Pila con su método constructor.

```
//Pila
class NodoPila{
    constructor(libro){
        this.libro = libro;
        this.siguiente = null;
    }
}
```

## class Lista\_Pila

Lista\_Pila es la clase de donde se almacenan los NodoPila en el que se encuentra su constructor y sus métodos de insertar Nodo, eliminar Nodo y contar los Nodos en la Pila.

```
class Lista_Pila{
    constructor(){ ...
    }

    insertar(Libro){ ...
    }

    eliminar(){ ...
    }

    contar(){ ...
    }
}
```

## class Autor

Autor es la clase Nodo del árbol de búsqueda binario, en ella se encuentra su método constructor, su método de insertar en el árbol binario, su método obtener graphviz que crea el código de graphviz ayudado del método de getCodigoInterno de forma recursiva.

```
//Arbol de busqueda
class Autor{
>   constructor(dpi, nombre_autor, correo, telefono, direccion, biografia){...
>   }
>   insertar(dpi, nombre_autor, correo, telefono, direccion, biografia){...
>   }
>   obtenerGraphviz(){ ...
>   }
>   getCodigoInterno(){...
>   }
}
```

## class Arbol

Árbol es la clase que almacena los Nodo de Autor en el que se encuentra su método constructor, su método insertar recursivo, su método para graficarArbol que genera un graphviz, métodos de inorden e inordenAux que recorren el árbol binario búsqueda para llenar la tabla con la información de Autores cuando y buscar con buscarAux se busca alguno en específico y se configura en la tabla.

```
class Arbol{
    constructor(){
        this.raiz = null;
    }
    > insertar(dpi, nombre_autor, correo, telefono, direccion, biografia){ ...
    }
    > graficarArbol(){ ...
    }
    > inorden(){ ...
    }
    > inordenAux(nodo, res){ ...
    }
    > buscar(nombre){ ...
    }
    > buscarAux(nodo, nombre){ ...
    }
}
```

## class Nodo\_Cabecera

Nodo\_Cabecera es la clase de Nodo que se genera para filas o columnas que luego son utilizados en la Matriz dependiendo lo que se requiera y sus métodos de getAcceso, setAcceso y getId.

```
//Matriz Dispersa
class Nodo_Cabecera{
>     constructor(id){ ...
    }
>     getAcceso(){ ...
    }
>     setAcceso(nuevo_acceso){ ...
    }
>     getId(){ ...
    }
}
```

## class Lista\_Cabecera

Lista\_Cabecera es donde se almacenan los Nodos\_Cabecera.

```
class Lista_Cabecera{
    constructor(tipo){
        this.primeros = null;
        this.ultimo = null;
        this.tipo = tipo; //si son columnas o filas
        this.size = 0;
    }
    > getPrimeros(){ ...
    }
    > insertar_nodoCabecera(nuevo){ ...
    }
    > mostrarCabeceras(){ ...
    }
    > getCabecera(id){ ...
    }
}
```

## class Nodo\_Celda

Nodo\_Celda es donde se almacena cada nodo de la Matriz con su información, entre ellos una Pila como atributo, con sus métodos.

```
class Nodo_Celda{
    > constructor(x,y,libro){ ...
    }
    > agregarPila(pila){ ...
    }
    > setArriba(arriba){ ...
    }
    > setAbajo(abajo){ ...
    }
    > setIzquierda(izquierda){ ...
    }
    > setDerecha(derecha){ ...
    }
    > getArriba(){ ...
    }
    > getAbajo(){ ...
    }
    > getIzquierda(){ ...
    }
    > getDerecha(){ ...
    }
}
```

## class Matriz\_Dispersa

Matriz\_Dispersa con su método constructor y sus métodos de insertar, imprimir, modificar, agregarPilaMatriz, comprarLibro, buscar y graficarNeato.

```
class Matriz_Dispersa{
    constructor(){ ...
    }
    insertar(pos_x,pos_y,Libro){ ...
    }

    imprimir(){ ...
    }

    modificar(fila,columna,Libro){ ...
    }

    agregarPilaMatriz(fila,columna,Libro,cantidad){ ...
    }

    comprarLibro(Libro,cliente){ ...
    }

    buscar(Libro){ ...
    }

    graficarNeato(){ ...
    }
}
```

## class Usuario y LibrosUsuarios

Usuario el método constructor que guarda los Nodos de la Lista de Usuarios entre ellos el apuntador para LibroUsuarios y LibroUsuarios que almacena cada Nodo que es un libro de algún Usuario.

```
class Usuario{
    constructor(dpi,nombre_completo,nombre_usuario,correo,rol,contrasenia,telefono){
        this.dpi=dpi;
        this.nombre_completo=nombre_completo;
        this.nombre_usuario=nombre_usuario;
        this.correo=correo;
        this.rol=rol;
        this.contrasenia=contrasenia;
        this.telefono=telefono;
        this.siguiente=null;
        this.abajo=null;
    }
}

//Libros Usuarios
class LibrosUsuarios{
    constructor(nombre,usuario){
        this.nombre=nombre;
        this.usuario=usuario;
        this.cantidad=0;
        this.siguiente=null;
    }
}
```



## class ListaUsuarios

Es donde se almacenan los Nodos de LibrosUsuarios y Usuarios, y sus métodos para InsertarUsuario, InsertarLibros, BuscarUsuario, RecorrerMenu que genera la validación del Menú, imprimirUsuarios, graficarExtra que es un método para ahorrar código del método de graficarUsuarios.

```
class ListaUsuarios{
>   constructor(){ ...
>   }
>   //Agregar Usuario
>   InsertarUsuario(dpi,nombre_completo,nombre_usuario,correo,rol,contrasenia,telefono){ ...
>   }
>   //Insertar Libros
>   InsertarLibros(nombre,usuario){ ...
>   }
>   //Buscar Usuario
>   BuscarUsuario(usuario){ ...
>   }
>   RecorrerMenu(usuario,contrasenia){ ...
>   }
>   imprimirUsuarios(){ ...
>   }
>   graficarExtra(actual){ ...
>   }
>   graficarUsuarios(){ ...
>   }
}
```

## class Top5Usuario y ListaTop5

Top5Usuario es el Nodo de usuarios con libros comprados y ListaTop5 su lista donde almacenan los métodos entre ellos el constructor, InsertarUsuario, Buscar, CambiarTop5, ordenarTop5, graficarTop5 y TablaTop5.

```
class Top5Usuario{
    constructor(nombre, usuario){ ...
    }
}

class ListaTop5{
    constructor(){ ...
    }
    InsertarUsuario(nombre, usuario){ ...
    }

    Buscar (usuario){ ...
    }

    CambiarTop5(usuario){ ...
    }

    ordenarTop5(){ ...
    }

    graficarTop5(){ ...
    }

    TablaTop5(){ ...
    }
}
```

## class Libro y ListaLibros

Libro es el Nodo de ListaLibros que cuenta con sus métodos de InsertarLibro, Tabla\_Libros que lo refleja en el html, Bubble que ordena los libros, imprimir, Partición y Sort que son utilizados para ordenar los libros.

```
//Libros
class Libro{
>   constructor(isbn, nombre_autor, nombre_Libro, paginas, categoria){...
>   }
> }

class ListaLibros{
>   constructor(){...
>   }
>   InsertarLibro(isbn, nombre_autor, nombre_Libro, paginas, categoria ){...
>   }
>   Tabla_Libros(){...
>   }
>   Bubble(){...
>   }
>   imprimir(){...
>   }
>   Particion(L,h){...
>   }
>   Sort(cabeza,cola){...
>   }
> }
```

## Funciones

Principalmente utilizadas para la acción de los botones, pero también para leer y cargar los archivos de tipo .json.

## Diagrama de Clases

Diagrama de Clases Proyecto 1

