



```
if ($(window).scrollTop() > header1_initialDistance) {  
  if (parseInt(header1.css('padding-top'), 10) > header1_initialPadding) {  
    header1.css('padding-top', '' + $(window).scrollTop() - header1_initialPadding + 'px');  
  }  
} else {  
  header1.css('padding-top', '' + header1_initialPadding + 'px');  
}  
  
if ($(window).scrollTop() > header2_initialDistance) {  
  if (parseInt(header2.css('padding-top'), 10) > header2_initialPadding) {  
    header2.css('padding-top', '' + $(window).scrollTop() - header2_initialPadding + 'px');  
  }  
} else {  
  header2.css('padding-top', '' + header2_initialPadding + 'px');  
}
```

MANUAL TÉCNICO

Edwin Sandoval López - 202010856

Guatemala, Abril 2023

ÍNDICE

INTRODUCCIÓN.....	2
OBJETIVOS.....	3
REQUERIMIENTOS.....	4
FUNCIONES PRINCIPALES.....	5
ESTRUCTURAS.....	7

INTRODUCCIÓN

Este manual técnico tiene como finalidad dar a conocer al lector que pueda requerir hacer modificaciones futuras al software del desarrollo de la aplicación indicando el IDE utilizado para su creación, su versión, requerimientos del sistema, etc.

La aplicación tiene como objetivo mostrar información de grado académico por medio de una página web de gestor de archivos por medio de archivos json, que contendrá toda la información para que luego la aplicación pueda cumplir con todas las especificaciones requeridas, permitiendo visualizar toda la información así como la capacidad de mostrar toda esta información utilizando estructuras de datos.

OBJETIVOS

General

Brindar al lector una guía que contenga la información del manejo de clases, atributos, métodos y del desarrollo de la aplicación desarrollada en el lenguaje de programación Javascript para facilitar futuras actualizaciones y modificaciones realizadas por terceros.

Específicos

- Proporcionar al lector una idea más precisa de los métodos y clases creadas para el desarrollo de la aplicación.
- Dar más información al lector de las herramientas utilizadas para el desarrollo de la aplicación.

REQUERIMIENTOS

SO:

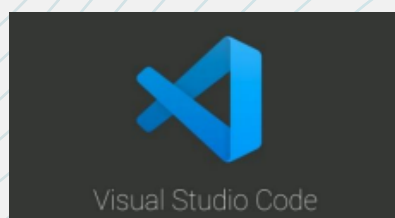
Las especificaciones del equipo utilizado para el desarrollo de la práctica se varan reflejadas en la siguiente imagen:

Device specifications	
Device name	DESKTOP-U9EBU00
Processor	12th Gen Intel(R) Core(TM) i7-12700K 3.61 GHz
Installed RAM	32.0 GB (31.7 GB usable)
Device ID	38C5838E-88A8-4D8F-84B1-2B72AA3B730A
Product ID	00326-10000-00000-AA899
System type	64-bit operating system, x64-based processor
Pen and touch	Pen support

JAVASCRIPT: Para realizar este proyecto se recomienda utilizar intérprete para ejecutar código Javascript, y el intérprete que se utiliza con frecuencia es nuestro navegador de internet.



IDE: Para el desarrollo del proyecto se trabajó con Visual Studio Code, queda a su discreción el ide a utilizar, sin embargo, por su fácil uso se recomienda.



FUNCIONES PRINCIPALES

Login(): Dicha función permite ingresar como administrador o como usuario normal dependiendo de las credenciales de cada usuario.

```
function login() {  
  let carnet = document.getElementById("carnet").value;  
  let password = document.getElementById("password").value;  
  let estudiante = avlTree.search(parseInt(carnet));  
  
  if (carnet.toLowerCase() == "admin" && password.toLowerCase() == "admin") {  
    localStorage.setItem("estudiante", JSON.stringify({ carnet: "Admin", nombre: "Administrador" }));  
    window.location.href = "../views/admin.html";  
  } else {  
    if (estudiante != null) {  
      if (estudiante.password == password) {  
        // set local storage estudiante  
        localStorage.setItem("estudiante", JSON.stringify(estudiante));  
        window.location.href = "../views/usuario.html";  
      } else {  
        alert("Contraseña incorrecta");  
      }  
    } else {  
      alert("El carnet no existe");  
    }  
  }  
}
```

generarAvlDot(): Esta función permite graficar el AVL

```
function generarAvlDot() {  
  let dot = "";  
  switch (selectOrden) {  
    case 1:  
      dot = avlTree.threeGraphInOrder();  
      break;  
    case 2:  
      dot = avlTree.threeGraphPreOrder();  
      break;  
    case 3:  
      dot = avlTree.threeGraphPostOrder();  
      break;  
    default:  
      dot = avlTree.threeGraphInOrder();  
      break;  
  }  
  
  const svg = Viz(dot, { format: "svg" });  
  const graphContainer = document.getElementById("graph");  
  graphContainer.innerHTML = svg;  
}
```

ESTRUCTURAS

Árbol AVL: Es un árbol binario de búsqueda (ABB), ampliado con un campo que indica el factor de equilibrio de cada nodo.

```
1  class AvlNode {
2      constructor(item) {
3          this.item = item;
4          this.left = null;
5          this.right = null;
6          this.height = 0;
7      }
8  }
9
10 class MatrizDispersa {
11     constructor() {
12         this.data = {};
13     }
14
15     insertar(row, col, value) {
16         if (!this.data[row]) {
17             this.data[row] = {};
18         }
19         this.data[row][col] = value;
20     }
21
22     obtener(row, col) {
23         if (this.data[row] && this.data[row][col]) {
24             return this.data[row][col];
25         } else {
26             return 0;
27         }
28     }
29
30     eliminar(row, col) {
31         if (this.data[row] && this.data[row][col]) {
32             delete this.data[row][col];
33             if (Object.keys(this.data[row]).length === 0) {
```

Matriz Dispersa: Estructura de datos utilizada para representar matrices en la que la mayoría de sus elementos tienen un valor de cero.

```
10 class MatrizDispersa {
11     constructor() {
12         this.data = {};
13     }
14
15     insertar(row, col, value) {
16         if (!this.data[row]) {
17             this.data[row] = {};
18         }
19         this.data[row][col] = value;
20     }
21
22     obtener(row, col) {
23         if (this.data[row] && this.data[row][col]) {
24             return this.data[row][col];
25         } else {
26             return 0;
27         }
28     }
29
30     eliminar(row, col) {
31         if (this.data[row] && this.data[row][col]) {
32             delete this.data[row][col];
33             if (Object.keys(this.data[row]).length === 0) {
```