



Resolución a problemas algorítmicos aplicando estructuras de almacenamiento

🔍 Status	Completed
☰ Codigo de guia	GA3-220501093-AA3-EV02.
☑ Entregada	<input type="checkbox"/>
📅 Semana	Semana 6
📎 Archivo resuelto	https://github.com/JahazielHernandezHoyos/GA3-220501093-AA3-EV02 GA3-220501093-AA3-EV02.zip
👤 Encargado	Jahaziel

Page test:

<https://algoritmos-one.vercel.app/>

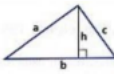
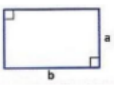
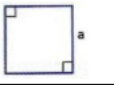

Github code:

<https://github.com/JahazielHernandezHoyos/GA3-220501093-AA3-EV02>

Ejercicios:

Utilizando el lenguaje JavaScript desarrollar un programa que dé solución a los siguientes problemas:

1. Desarrollar un programa que permita calcular el área o perímetro de algunas figuras planas según la siguiente tabla:

Figura	Perímetro	Área
	$a + b + c$	$\frac{b \times h}{2}$
	$2 \times (b + a)$	$b \times a$
	$4 \times a$	a^2
	$2 \times \pi \times r^2$	$\pi \times r^2$

```

/*
RESPUESTA
*/

function areaTriangulo() {
    alert("Bienvenido al programa para calcular el área de un triangulo");
    const a = prompt("Ingrese el valor de a");
    const b = prompt("Ingrese el valor de b");
    const c = prompt("Ingrese el valor de c");
    const area = parseFloat(a) + parseFloat(b) + parseFloat(c);
    alert("El area del triangulo es: " + area);
}

function perimetroCirculo() {
    alert("Bienvenido al programa para calcular el perimetro de un circulo");
    const pi = 3.1416;
    const r = prompt("Ingrese el valor de r");
    const perimetro = 2 * pi * parseFloat(r)**2;
    alert("El perimetro del circulo es: " + perimetro, "dado que r es igual a 10");
}

```

1. Desarrollar un programa que permita almacenar las edades de un grupo de 10 personas en un vector de enteros y luego determine la cantidad de personas que son menores de edad, mayores de edad, cuántos adultos mayores, la edad más baja, la edad más alta y el promedio de edades ingresadas. Para el ejercicio anterior suponga que un adulto mayor debe tener una edad igual o superior a 60. Debe validar para cada ingreso que los valores estén en un rango entre 1 y 120 años. En caso de error deberá notificar y solicitar un nuevo valor.

```

/*
RESPUESTA
*/

function almacenDeEdades(){
    alert("Hola bienvenido al organigrama de edades");
    const grupoEdades = [];
    let contadorMenores = 0;
    let contadorMayores = 0;
    let contadorAdultosMayores = 0;

    for (let i = 1; i <= 10; i++){
        let edad = parseInt(prompt("Introduzca la edad de la persona numero " + i));
        if (edad >= 1 && edad <= 120){
            grupoEdades.push(edad);
        } else {
            alert("La edad ingresada no es valida, ingrese de nuevo el valor");
            i--;
            continue;
        }
        if (edad <= 18 && edad <= 60){
            contadorMenores++;
        } else {

```

```

        contadorMayores++;
    }
    if (edad >= 60){
        contadorAdultosMayores++
    }
}

let edadMasAlta = Math.max.apply(null, grupoEdades);
let edadMasBaja = Math.min.apply(null, grupoEdades);
let promedioEdades = 0
for(let i of grupoEdades) promedioEdades+=i;
promedioEdades = promedioEdades/grupoEdades.length;

alert("-La cantidad de menores de edad es: " + contadorMenores +
      "\n-La cantidad de mayores de edad es: " + contadorMayores +
      "\n-La cantidad de adultos mayores es: " + contadorAdultosMayores +
      "\n-La edad mas alta es: " + edadMasAlta +
      "\n-La edad mas baja es: " + edadMasBaja +
      "\n-El promedio de edades es: " + promedioEdades);
}

```

2. Escriba un programa que lea dos vectores de números enteros ordenados ascendentemente y luego produzca la lista ordenada de la mezcla de los dos, por ejemplo, si los dos arreglos tienen los números 1 3 6 9 17 y 2 4 10 17, respectivamente, la lista de números en la pantalla debe ser 1 2 3 4 6 9 10 17 17. Limite los vectores a un tamaño de 5 y debe validar en cada ingreso que realmente se estén ingresando los datos de forma ascendente.

```

/*
RESPUESTA
*/

function mezclaDeVectores(){
    alert("Bienvenido al programa de mezcla de vectores");
    const vector1 = [];
    const vector2 = [];
    const vectorMezclado = [];
    let i = 0;
    let j = 0;
    let k = 0;
    let numeroIngresado = 0;
    let numeroAnterior = 0;
    let numeroAnterior2 = 0;

    while (i < 5){
        numeroIngresado = parseInt(prompt("Ingrese el numero " + (i + 1) + " del vector 1"));
        if (numeroIngresado > numeroAnterior){
            vector1.push(numeroIngresado);
            numeroAnterior = numeroIngresado;
            i++;
        } else {
            alert("El numero ingresado no es valido, ingrese de nuevo el valor");
            continue;
        }
    }

    while (j < 5){
        numeroIngresado = parseInt(prompt("Ingrese el numero " + (j + 1) + " del vector 2"));
        if (numeroIngresado > numeroAnterior2){
            vector2.push(numeroIngresado);
            numeroAnterior2 = numeroIngresado;
            j++;
        } else {
            alert("El numero ingresado no es valido, ingrese de nuevo el valor");
            continue;
        }
    }

    while (k < 10){
        if (vector1[0] < vector2[0]){
            vectorMezclado.push(vector1[0]);
            vector1.shift();
        } else {
            vectorMezclado.push(vector2[0]);
            vector2.shift();
        }
        k++;
    }
}

```

```

    alert("El vector mezclado es: " + vectorMezclado);
}

```

3. Una emisora con presencia en diferentes ciudades desea conocer el rating de canciones y cantantes más escuchados (sonados) en este semestre del año. Por lo tanto, se ha pedido a estudiantes del SENA del programa de tecnólogo en análisis y desarrollo de software desarrollar una solución que permita conocer la respuesta de 6 personas con relación a sus gustos musicales. Con fines administrativos y realizar una rifa entre las personas encuestadas, la emisora desea poder registrar de las personas entrevistadas su nombre, número de identificación (cédula), fecha de nacimiento, correo electrónico, ciudad de residencia, ciudad de origen. Además, se deberá poder almacenar el artista y título de hasta 3 canciones favoritas en GFPI-F-135 V01 cada una de las personas que se ingrese, teniendo en cuenta lo anterior, se sugiere que la solución deberá mostrar un menú que permite las siguientes opciones:

- Agregar una persona con los datos que se listan anteriormente.
- Mostrar la información personal de una persona particular por medio de su posición en el vector.

```

function encuestaDeMusica(){
    alert("Bienvenido al programa de encuesta de musica");
    const personas = [];
    let opcion = 0;

    do {
        opcion = parseInt(prompt("Ingrese la opcion que desea realizar: \n1. Agregar una persona \n2. Mostrar informacion de una person
        switch(opcion){
            case 1:
                agregarPersona(personas);
                break;
            case 2:
                mostrarPersona(personas);
                break;
            case 3:
                alert("Gracias por usar el programa");
                break;
            default:
                alert("La opcion ingresada no es valida, ingrese de nuevo el valor");
                break;
        }
    } while (opcion != 3);
}

function agregarPersona(personas){
    const persona = {
        nombre: "",
        identificacion: 0,
        fechaNacimiento: "",
        correo: "",
        ciudadResidencia: "",
        ciudadOrigen: "",
        cancionesFavoritas: []
    }

    persona.nombre = prompt("Ingrese el nombre de la persona");
    persona.identificacion = parseInt(prompt("Ingrese la identificacion de la persona"));
    persona.fechaNacimiento = prompt("Ingrese la fecha de nacimiento de la persona");
    persona.correo = prompt("Ingrese el correo de la persona");
    persona.ciudadResidencia = prompt("Ingrese la ciudad de residencia de la persona");
    persona.ciudadOrigen = prompt("Ingrese la ciudad de origen de la persona");

    let i = 0;
    while (i < 3){
        let cancion = {
            artista: "",
            titulo: ""
        }
        cancion.artista = prompt("Ingrese el artista de la cancion " + (i + 1));
        cancion.titulo = prompt("Ingrese el titulo de la cancion " + (i + 1));
        persona.cancionesFavoritas.push(cancion);
        i++;
    }

    personas.push(persona);
}

```

```
function mostrarPersona(personas){
    let posicion = parseInt(prompt("Ingrese la posicion de la persona que desea mostrar"));
    if (posicion > personas.length){
        alert("La posicion ingresada no es valida, ingrese de nuevo el valor");
        return;
    }
    alert("La persona en la posicion " + posicion + " es: " + personas[posicion - 1].nombre);
}
```