2020

SISTEMAS NUMÉRICOS: Introducción a la Informática

GIAN FRANCO POSSO GIRALDO/ DANIEL ENRIQUE VILLA ARIAS

DICIEMBRE DE 2020

UTP | Pereira

1 CONTENIDO

2 PRESENTACIÓN

La presente monografía presenta las actividades realizadas durante la tercera previa, que se basa prioritariamente en el manejo de los arrays.

En los siguientes párrafos se presenta una descripción básica de que es y como se utilizan los arrays.

AUTORES: Gian Franco Posso Giraldo/ Daniel Enrique Villa Arias

CÓDIGO: 100468162/ 1010003883

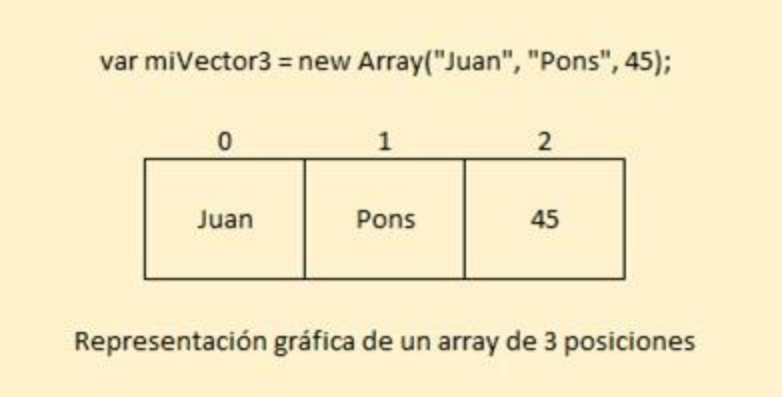
CORREO: f.posso@utp.edu.co/ daniel.villla2@upt.edu.co

GITHUB:

3 ARRAYS I

En este punto se da una introducción al manejo de arrays.

Los arrays son objetos similares a una lista cuyo prototipo proporciona métodos para efectuar operaciones de recorrido y de mutación. Tanto la longitud como el tipo de los elementos de un array son variables. Dado que la longitud de un array puede cambiar en cualquier momento, y los datos se pueden almacenar en ubicaciones no contiguas, no hay garantía de que los arrays de JavaScript sean densos; esto depende de cómo se elijan usarlos.



Representación gráfica de un array de 3 posiciones

A continuación, se presenta un código en el que se muestran varias formas de cómo utilizar los arrays.

```
<!DOCTYPE HTML>

<html>

<head>
  <title>Arrays-01</title>
</head>

<body>

<script>

////////////////////////////////////////
// ARREGLOS EN JAVASCRIPT - PARTE I //
////////////////////////////////////////

// Función para imprimir texto en la pantalla
function texto( cadena, lineas_post = 0 ) {
  if (typeof(cadena) == "string") {
    if (cadena.indexOf("(::)") == 0) {
      document.write("============================================================<br>");
    }
  }
  document.write( cadena );
  for (var i = 0; i < lineas_post; i++) {
    document.write("<br>");
  }
}

texto("(::) TALLER SOBRE MANEJO INTEGRAL DE ARRAYS EN JAVASCRIPT", 2);

////////////////////////////////////
// Asignando valores a un array //
////////////////////////////////////

// Definiendo un array vacío
var numeros = [];

// Asignando valores a cada posición del array, mediante un ciclo
for (var i = 0; i < 100; ++i) {
  numeros[i] = i+1;
}

// Mostrando el array en la pantalla
texto("(::) Array con los números desde el 1 hasta el 100", 2)
for (var i = 0; i < 100; i++) {
  texto(numeros[i] + "-");
}

//////////////////////////////////////////////////
// Acceso a cada elemento individual del array. //
// Luego, se suman los elementos                //
//////////////////////////////////////////////////
```

```
var numeros = [1,2,3,4,5];

var suma = numeros[0] + numeros[1] + numeros[2] + numeros[3] + numeros[4];

texto("", 2);
texto("(::) Suma manual de los elementos del array: " + numeros, 2)
texto("Suma = " + suma, 2);

////////////////////////////////////
// Suma de los elementos del array //
// Utilizando un ciclo            //
////////////////////////////////////

var numeros = [1,2,3,5,8,13,21];

var suma = 0;

for (var i = 0; i < numeros.length; ++i)
{
  suma += numeros[i];
}

texto("(::) Suma de elementos del array: " + numeros + " utilizando ciclos", 2);
texto("Suma = " + suma, 2);

//////////////////////////////////
// Diviendo un array en partes //
//////////////////////////////////

var frase = "la programación orientada a objetos en javascript";

var palabras = frase.split(" ");

texto("(::) Se divide una frase en palabras. La frase es = '" + frase + "'", 2);
for (var i = 0; i < palabras.length; ++i)
{
  texto("Palabra: " + i + ": " + palabras[i], 1);
}

texto("", 1);

////////////////////////////////////////////////
// La asignación simple de arrays no produce //
// dos arrays diferentes. Ambos señalan a la //
// misma zona de memoria                     //
////////////////////////////////////////////////

var numeros = [];

for (var i = 0; i < 10; ++i)
{
  numeros[i] = i + 1;
}
```

```
// Los array numeros y copia_numeros, son en realidad el mismo array
var copia_numeros = numeros;

numeros[0] = 400;

texto("(::) Al cambiar la posición cero de 'numeros', también se cambia la posición 0 de 'copia_numeros'", 2);

texto("(-) numeros[0] = " + numeros[0], 1);
texto("(-) copia_numeros[0] = " + copia_numeros[0], 2)

//////////////////////////////////////////
// Se utiliza una funciónj de copia para //
// que los dos arrays sean diferentes    //
//////////////////////////////////////////

function copiar(arr1, arr2) {
  for (var i = 0; i < arr1.length; ++i) {
    arr2[i] = arr1[i];
  }
}

var numeros = [];
for (var i = 0; i < 100; ++i) {
numeros[i] = i+1;
}
var copia_numeros = [];

// Se copia el array numeros en copia_numeros
copiar(numeros, copia_numeros);

numeros[0] = 400;

texto("(::) Se verifica que al cambiar la posición cero en numeros: ", 1);
texto("NO se cambia la posición cero en el array copia_numeros", 2);

texto("(-) numeros[0] = " + numeros[0], 1);
texto("(-) copia_numeros[0] = " + copia_numeros[0], 2);

//////////////////////////////////////////////////////
// Se muestra aquí como buscar un dato en un array //
//////////////////////////////////////////////////////

var nombres = ["David", "Sofía", "Ramón", "Carlos", "María"];
texto("(::) Lista de nombres = ", 1);
texto("(-) " + nombres, 2);

nombre = "Ramón";

texto("(-) Se está buscando el nombre: Ramón", 1);

var posicion = nombres.indexOf(nombre);

if (posicion >= 0) {
  texto("(-) Encontrado " + nombre + " en la posición " + posicion, 2);
```

```
}
else {
  texto("(-) " + nombre + " no encontrado en el array.", 2);
}

//////////////////////////////////////////////////
// Buscando el primero y el último de los elementos //
//////////////////////////////////////////////////

var nombres = ["David", "Miguel", "Sofía", "Miguel", "Teresa", "Miguel", "Ramón", "Carlos", "Miguel", "María"];

texto("(::) Se busca un nombre en la lista = ", 1);
texto("(-) " + nombres, 2);

var nombre = "Miguel";

texto("(-) El nombre a buscar es: " + nombre, 1);

var primeraPosicion = nombres.indexOf(nombre);

texto("(-) Encontrado " + nombre + " en la primera posición = " + primeraPosicion, 1);

var ultimaPosicion = nombres.lastIndexOf(nombre);

texto("(-) Encontrado " + nombre + " en la última posición = " + ultimaPosicion, 2);

//////////////////////////////////////////////////
// Unión de elementos de un array utilizando join //
//////////////////////////////////////////////////

var nombres = ["David", "Sofía", "Ramón", "Carlos", "Miguel", "María"];

var nombres_union = nombres.join();

texto("(::) Unión de un array utilizando join. Todos sus elementos se unen en una sola cadena: ", 2)
texto(nombres_union, 2)

//////////////////////////////////////////////////////////////////
// Mostrar un array utilizando directamente el nombre del Array //
// Se utiliza el método: toString()                             //
//////////////////////////////////////////////////////////////////

nombres_union = nombres.toString();
texto("(::) Utilización del método toString para obtener el string equivalente de un array", 2);
texto(nombres_union, 2);

//////////////////////////////////////////////
// Concatenacion de los elementos de un array //
//////////////////////////////////////////////

var cadena_1 = ["Miguel", "Carlos", "Antonio", "Daniel", "María"];
var cadena_2 = ["Ramón", "Sofía", "Bernardo"];
```

```
texto("(::) Concatenación de arrays utilizando concat",  2);

texto("(-) Primer array  = " + cadena_1, 1);
texto("(-) Segundo array = " + cadena_2, 2);

texto("(-) Primera forma de concatenacion. cadena_2 se concatena a cadena_1", 1);
var concatenacion = cadena_1.concat(cadena_2);
texto("(-) " + concatenacion, 2);

concatenacion = cadena_2.concat(cadena_1);

texto("(-) Segunda forma de concatenacion. cadena_1 se concatena a cadena_2", 1);
texto("(-) " + concatenacion, 2);

//////////////////////////////////////////
// Utilización de la instrucción: split //
//////////////////////////////////////////

var concatenacion = ["Miguel","Carlos","Antonio","Ramón","Sofía","Daniel","María"];

texto("(::) Ilustración del método de división por partes utilizando splice", 2)
texto("(-) El array original = " + concatenacion, 1);

// Extrae una parte del array y la coloca en 'unaParte'
var unaParte = concatenacion.splice(3,3);

// En concatenacion queda el excedente, al cual se le denomina 'otraParte'
var otraParte = concatenacion;


texto("(-) Una parte splice(3,3) = " + unaParte, 1); // Ramón,Sofía,Daniel
texto("(-) Otra parte (lo que sobra) = " + otraParte, 2); // Miguel,Carlos,Antonio,María

////////////////////////////////////////////
// Agregando elementos al final de un array //
////////////////////////////////////////////

var numeros = [1,2,3,4,5];
texto("(::) Agregando elementos a un array (al final)", 2)
texto("(-) Array antes de agregar un elemento al final: " + numeros, 1); // 1,2,3,4,5
numeros.push(6);
texto("(-) Array después de agregar el elemento: " + numeros, 2); // 1,2,3,4,5,6

////////////////////////////////////////////////
// Agregando elementos al comienzo de un array //
// Método manual ineficiente                   //
////////////////////////////////////////////////

var numeros = [2,3,4,5];

texto("(::) Agregando nuevo número al comienzo del array (método ineficiente)", 2);
texto("(-) Array inicial: " + numeros, 1);
```

```
var nuevo_numero = 1;
var N = numeros.length;
for (var i = N; i >= 0; --i) {
numeros[i] = numeros[i-1];
}
numeros[0] = nuevo_numero;

texto("(-) Adición del valor 1 al comienzo del array. Números = " + numeros, 2); // 1,2,3,4,5

//////////////////////////////////////////
// Agregando datos al comienzo del array //
// Método más eficiente: uhshift        //
//////////////////////////////////////////

texto("(::) Agregando elementos al comienzo de un array. Utlización de unshift", 2);

var numeros = [2,3,4,5];
texto("(-) Números inicial = " + numeros, 1); // 2,3,4,5

var nuevo_numero = 1;
numeros.unshift(nuevo_numero);
texto("(-) Agregado el 1: " + numeros, 2); // 1,2,3,4,5

numeros = [3,4,5];

texto("(-) Números inicial = " + numeros, 1); // 3,4,5

numeros.unshift(1,2);
texto("(-) Agregados dos valores, el 1 y el 2. Nuevo array = " + numeros, 2); // 1,2,3,4,5

////////////////////////////////////////////////
// Eliminando elementos del final de un array //
// Instrucción pop()                          //
////////////////////////////////////////////////

var numeros = [1,2,3,4,5,9];
texto("(::) Removiendo un dato al final del array", 2);

texto("(-) Array original = " + numeros, 1);

numeros.pop();
texto("(-) Array sin el último elemento: " + numeros, 2); // 1,2,3,4,5

//////////////////////////////////////////////
// Removiendo datos del inicio de un array //
// Instrucción: shift                      //
//////////////////////////////////////////////

var numeros = [9,1,2,3,4,5];
texto("(::) Removiendo elementos del comienzo de un array. Se utiliza 'unshift'", 2);

texto("(-) Valor original de Números = " + numeros, 1);
numeros.shift();
texto("(-) Se elimina el primer elemento del array: " + numeros, 2); // 1,2,3,4,5
```

```
//////////////////////////////////////////
// Agregando datos en medio de un array //
//////////////////////////////////////////

var numeros = [1,2,3,7,8,9];

texto("(::) Agregando elementos en la mitad de un array. ", 2);
texto("(-) Array Números (antes) = " + numeros, 1); // 1,2,3,7,8,9

var nuevos_elementos = [4,5,6];
numeros.splice(3,0,nuevos_elementos);

texto("(-) Números (después). Se agrega [4,5,6] a partir de la tercera posición = " + numeros, 2); // 1,2,3,4,5,6,7,8,9

// Una variante. Se agregan directamente los elementos
var numeros = [1,2,3,7,8,9];

texto("(::) Agregando elementos en la mitad de un array. Método directo, agregando los valores en la instrucción: ", 2);
texto("(-) Números (antes) = " + numeros, 1); // 1,2,3,7,8,9

numeros.splice(3,0,4,5,6);

texto("(-) Números (después) = " + numeros, 2); // 1,2,3,4,5,6,7,8,9

//////////////////////////////////////////
// Eliminando datos en medio de un array //
//////////////////////////////////////////

var numeros = [1,2,3,100,200,300,400,4,5];

texto("(::) Eliminando elementos en la mitad de un array. Se utiliza splice, según se aprecia en la instrucción", 2);
texto("(-) Números (antes) = " + numeros, 1); // 1,2,3,100,200,300,400,4,5

numeros.splice(3,4);

texto("(-) Números (después) = " + numeros, 2); // 1,2,3,4,5

////////////////////////
// Invirtiendo arrays //
////////////////////////

var numeros = [1,2,3,4,5];
texto("(::) Invirtiendo arrays: Número = " + numeros, 2);

numeros.reverse();

texto("(-) Array invertido: " + numeros, 2); // 5,4,3,2,1

///////////////////////////////////
// Ordenando un array (strings) //
///////////////////////////////////

var nombres = ["David","Miguel","Carlos","María","Bernardo","Ramón"];
```

```
texto("(::) Ordenando el array = " + nombres, 2);

nombres.sort();
texto("(-) Array ordenado = " + nombres, 2);
// Bernardo, Carlos, David, María, Miguel, Ramón

//////////////////////////////////////////
// Ordenando un array (números) - Erróneo //
//////////////////////////////////////////

texto("(::) Ordenamiento de números. En su forma básica, los números se ordenan mal", 2);

var numeros = [3,1,2,100,4,200];
texto("(-) Array original = " + numeros, 1);

numeros.sort();
texto("(-) Array ordenado = " + numeros, 2); // 1,100,2,200,3,4

//////////////////////////////////////////
// Ordenando un array (números) - Corregido //
//////////////////////////////////////////

// Se debe utilizar una función de comparación
function comparar(num1, num2) {
return num1 - num2;
}

texto("(::) Ordenamiento numérico corregido", 2);

var numeros = [3,1,2,100,4,200];

texto("(-) Array original = " + numeros, 1);

numeros.sort(comparar);

texto("(-) Array ordenado = " + numeros, 2);   // 1,2,3,4,100,200

//////////////////////////////////
// ITERADORES - Función forEach //
//////////////////////////////////

function cuadrado(numero) {
    texto(numero.toString() + " --- " + (numero * numero), 1);
}
var numeros = [1,2,3,4,5,6,7,8,9,10];

texto("(::) Cuadrados, utilizando forEach", 2);

numeros.forEach(cuadrado);

//////////////////////////////////
// ITERADORES - Función every   //
//////////////////////////////////
```

```
function esPar(numero) {
    return numero % 2 == 0;
}

var numeros = [2,4,6,8,10];
var par = numeros.every(esPar);

texto("(::) Detección números pares con el iterador: every", 2)

if (par) {
    texto("(-) Todos los números son de tipo par", 2);
}
else {
    texto("(-) No todos los números son de tipo par", 2);
}

//////////////////////////////
// ITERADORES: Función some //
//////////////////////////////

function esPar(numero) {
    return numero % 2 == 0;
}

var numeros = [1,2,3,4,5,6,7,8,9,10];
var algunoPar = numeros.some(esPar);

texto("(::) Verificación de si algún número es par", 2)

texto("(-) Array de números = " + numeros, 1);
if (algunoPar) {
    texto("(-) Algunos números son par", 2);
}
else {
    texto("(-) Ningún número es par", 2);
}
numeros = [1,3,5,7,9];
algunoPar = numeros.some(esPar);

texto("(-) Array de números = " + numeros, 1);
if (algunoPar) {
    texto("(-) Algunos números son par", 2);
}
else {
    texto("(-) Ningún número es par", 2);
}

//////////////////////////////////
// ITERADORES: Función reduce //
//////////////////////////////////

function agregar(avance, valorActual) {
    return avance + valorActual;
}
```

```
var numeros = [1,2,3,4,5,6,7,8,9,10];
var suma = numeros.reduce(agregar);

texto("(::) La función reduce genera un ciclo 'oculto'", 2);

texto("(-) Array original = " + numeros, 1);
texto("(-) Suma = " + suma, 2); // muestra 55

/*
CICLO OCULTO:

agregar(1,2) -> 3
agregar(3,3) -> 6
agregar(6,4) -> 10
agregar(10,5) -> 15
agregar(15,6) -> 21
agregar(21,7) -> 28
agregar(28,8) -> 36
agregar(36,9) -> 45
agregar(45,10) -> 55

*/

////////////////////////////////////////
// ITERADORES: Función reduce - String //
////////////////////////////////////////

function concatenar(stringAcumulado, elemento) {
    return stringAcumulado + elemento;
}

texto("(::) Concatenación de cadenas utilizando 'reduce'", 2);

var palabras = ["La ", "programación ","en ", "Javascript "];
var frase = palabras.reduce(concatenar);

texto("(-) Frase = " + frase, 2); // muestra "La programación en Javascript"

////////////////////////////////////////////////
// ITERADORES: Función reduce - String (inverso) //
////////////////////////////////////////////////

function concatenar(stringAcumulado, elemento) {
    return stringAcumulado + elemento;
}

texto("(::) Concatenación de cadenas utilizando 'reduce', de Derecha a Izquierda", 2);

var palabras = ["La ", "programación ","en ", "Javascript "];
var frase = palabras.reduceRight(concatenar);

texto("(-) Frase invertida = " + frase, 2); // muestra "La programación en Javascript"
```

```
////////////////////////////////////
// ITERADORES que retornan un ARRAY //
////////////////////////////////////

////////////////////////////
// ITERADORES: Función map //
////////////////////////////

function curva(grado) {
    return grado += 5;
}

var grados = [77, 65, 81, 92, 83];

texto("(::) Utilización de la función 'map'", 2);

texto("(-) Array original = " + grados, 1);

var nuevos_grados = grados.map(curva);

texto("(-) Nuevos grados = " + nuevos_grados, 2); // 82, 70, 86, 97, 88

////////////////////////////
// ITERADORES: Función map //
////////////////////////////

function primero(palabra) {
    return palabra[0];
}
var palabras = ["para", "su", "información"];
var acronimo = palabras.map(primero);

texto("(::) Frase original = " + palabras, 2);

texto("(-) Acrónimo = " + acronimo.join(""), 2); // muestra 'psi'

//////////////////////////////
// ITERADORES: Función filter //
//////////////////////////////

function esPar(numero) {
    return numero % 2 == 0;
}

function esImpar(numero) {
    return numero % 2 != 0;
}

var numeros = [];

for (var i = 0; i < 20; ++i) {
    numeros[i] = i + 1;
}
```

```
texto("(::) Detección de pares e impares con 'filter'", 2);

texto("(-) Array = " + numeros, 2);

var pares = numeros.filter(esPar);

texto("(-) Números Pares: ", 1);

texto(pares, 2);

var impares = numeros.filter(esImpar);

texto("(-) Números Impares: ", 1);
texto(impares, 2);

////////////////////////////////////////////////
// ITERADORES: Función filter - Otra aplicación //
////////////////////////////////////////////////

function corte(numero) {
    return numero >= 60;
}
var grados = [];

for (var i = 0; i < 20; ++i) {
    grados[i] = Math.floor(Math.random() * 101);
}

texto("(::) Filtro de grados mayores a 60", 2);

texto("(-) Grados = " + grados, 2);

var corteSuperior = grados.filter(corte);

texto("(-) Corte Superior = ");
texto(corteSuperior, 2);

////////////////////////////////////////////////
// ITERADORES: Función filter - Otra aplicación //
////////////////////////////////////////////////

function afterc(str) {
    if (str.indexOf("cie") > -1) {
        return true;
    }
    return false;
}
var palabras = ["recieve","deceive","percieve","deceit","concieve"];

texto("(::) Detecta palabras mal escritas. En este caso, las que tengan las letras 'cie'", 2);

texto("(-) Palabras origen: " + palabras, 1);

var mal_escrito = palabras.filter(afterc);

texto("(-) Palabras mal escritas = " + mal_escrito, 2); // muestra: recieve,percieve,concieve

</script>
</body>

</html>
```

Cuando se ejecuta este código quedará el siguiente diseño de pantalla.

```
============================================================
(::) TALLER SOBRE MANEJO INTEGRAL DE ARRAYS EN JAVASCRIPT

============================================================
(::) Array con los números desde el 1 hasta el 100

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-
21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-
41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-
61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-
81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-

============================================================
(::) Suma manual de los elementos del array: 1,2,3,4,5

Suma = 15

============================================================
(::) Suma de elementos del array: 1,2,3,5,8,13,21 utilizando ciclos

Suma = 53

============================================================
(::) Se divide una frase en palabras. La frase es = 'la programación orientada a objetos en javascript'

Palabra: 0: la
Palabra: 1: programación
Palabra: 2: orientada
Palabra: 3: a
Palabra: 4: objetos
Palabra: 5: en
Palabra: 6: javascript

============================================================
(::) Al cambiar la posición cero de 'numeros', también se cambia la posición 0 de 'copia_numeros'

(-) numeros[0] = 400
(-) copia_numeros[0] = 400

============================================================
(::) Se verifica que al cambiar la posición cero en numeros:
NO se cambia la posición cero en el array copia_numeros

(-) numeros[0] = 400
(-) copia_numeros[0] = 1

============================================================
(::) Lista de nombres =
(-) David,Sofía,Ramón,Carlos,María

(-) Se está buscando el nombre: Ramón
(-) Encontrado Ramón en la posición 2
```

```
============================================================
(::) Se busca un nombre en la lista =
(-) David,Miguel,Sofía,Miguel,Teresa,Miguel,Ramón,Carlos,Miguel,María

(-) El nombre a buscar es: Miguel
(-) Encontrado Miguel en la primera posición = 1
(-) Encontrado Miguel en la última posición = 8

============================================================
(::) Unión de un array utilizando join. Todos sus elementos se unen en una sola cadena:

David,Sofía,Ramón,Carlos,Miguel,María

============================================================
(::) Utilización del método toString para obtener el string equivalente de un array

David,Sofía,Ramón,Carlos,Miguel,María

============================================================
(::) Concatenación de arrays utilizando concat

(-) Primer array = Miguel,Carlos,Antonio,Daniel,María
(-) Segundo array = Ramón,Sofía,Bernardo

(-) Primera forma de concatenacion. cadena_2 se concatena a cadena_1
(-) Miguel,Carlos,Antonio,Daniel,María,Ramón,Sofía,Bernardo

(-) Segunda forma de concatenacion. cadena_1 se concatena a cadena_2
(-) Ramón,Sofía,Bernardo,Miguel,Carlos,Antonio,Daniel,María

============================================================
(::) Ilustración del método de división por partes utilizando splice

(-) El array original = Miguel,Carlos,Antonio,Ramón,Sofía,Daniel,María
(-) Una parte splice(3,3) = Ramón,Sofía,Daniel
(-) Otra parte (lo que sobra) = Miguel,Carlos,Antonio,María

============================================================
(::) Agregando elementos a un array (al final)

(-) Array antes de agregar un elemento al final: 1,2,3,4,5
(-) Array después de agregar el elemento: 1,2,3,4,5,6

============================================================
(::) Agregando nuevo número al comienzo del array (método ineficiente)

(-) Array inicial: 2,3,4,5
(-) Adición del valor 1 al comienzo del array. Números = 1,2,3,4,5
```

```
============================================================
(::) Agregando elementos al comienzo de un array. Utlización de unshift

(-) Números inicial = 2,3,4,5
(-) Agregado el 1: 1,2,3,4,5

(-) Números inicial = 3,4,5
(-) Agregados dos valores, el 1 y el 2. Nuevo array = 1,2,3,4,5

============================================================
(::) Removiendo un dato al final del array

(-) Array original = 1,2,3,4,5,9
(-) Array sin el último elemento: 1,2,3,4,5

============================================================
(::) Removiendo elementos del comienzo de un array. Se utiliza 'unshift'

(-) Valor original de Números = 9,1,2,3,4,5
(-) Se elimina el primer elemento del array: 1,2,3,4,5

============================================================
(::) Agregando elementos en la mitad de un array.

(-) Array Números (antes) = 1,2,3,7,8,9
(-) Números (después). Se agrega [4,5,6] a partir de la tercera posición = 1,2,3,4,5,6,7,8,9

============================================================
(::) Agregando elementos en la mitad de un array. Método directo, agregando los valores en la instrucción:

(-) Números (antes) = 1,2,3,7,8,9
(-) Números (después) = 1,2,3,4,5,6,7,8,9

============================================================
(::) Eliminando elementos en la mitad de un array. Se utiliza splice, según se aprecia en la instrucción

(-) Números (antes) = 1,2,3,100,200,300,400,4,5
(-) Números (después) = 1,2,3,4,5

============================================================
(::) Invirtiendo arrays: Número = 1,2,3,4,5

(-) Array invertido: 5,4,3,2,1

============================================================
(::) Ordenando el array = David,Miguel,Carlos,María,Bernardo,Ramón

(-) Array ordenado = Bernardo,Carlos,David,María,Miguel,Ramón

============================================================
(::) Ordenamiento de números. En su forma básica, los números se ordenan mal

(-) Array original = 3,1,2,100,4,200
(-) Array ordenado = 1,100,2,200,3,4
```

```
============================================================
(::) Ordenamiento numérico corregido

(-) Array original = 3,1,2,100,4,200
(-) Array ordenado = 1,2,3,4,100,200

============================================================
(::) Cuadrados, utilizando forEach

1 --- 1
2 --- 4
3 --- 9
4 --- 16
5 --- 25
6 --- 36
7 --- 49
8 --- 64
9 --- 81
10 --- 100
============================================================
(::) Detección números pares con el iterador: every

(-) Todos los números son de tipo par

============================================================
(::) Verificación de si algún número es par

(-) Array de números = 1,2,3,4,5,6,7,8,9,10
(-) Algunos números son par

(-) Array de números = 1,3,5,7,9
(-) Ningún número es par

============================================================
(::) La función reduce genera un ciclo 'oculto'

(-) Array original = 1,2,3,4,5,6,7,8,9,10
(-) Suma = 55

============================================================
(::) Concatenación de cadenas utilizando 'reduce'

(-) Frase = La programación en Javascript

============================================================
(::) Concatenación de cadenas utilizando 'reduce', de Derecha a Izquierda

(-) Frase invertida = Javascript en programación La

============================================================
(::) Utilización de la función 'map'

(-) Array original = 77,65,81,92,83
(-) Nuevos grados = 82,70,86,97,88
```

```
==========================================================
(::) Frase original = para,su,información

(-) Acrónimo = psi

==========================================================
(::) Detección de pares e impares con 'filter'

(-) Array = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

(-) Números Pares:
2,4,6,8,10,12,14,16,18,20

(-) Números Impares:
1,3,5,7,9,11,13,15,17,19

==========================================================
(::) Filtro de grados mayores a 60

(-) Grados = 7,100,8,10,34,99,42,2,4,4,79,34,86,5,88,36,51,39,55,34

(-) Corte Superior = 100,99,79,86,88

==========================================================
(::) Detecta palabras mal escritas. En este caso, las que tengan las letras 'cie'

(-) Palabras origen: recieve,deceive,percieve,deceit,concieve
(-) Palabras mal escritas = recieve,percieve,concieve
```

4 ARRAYS II

Se continúa realizando ejercicios sobre el manejo de arrays.

A continuación, se muestra un código que muestra una manera de cómo utilizar arrays.

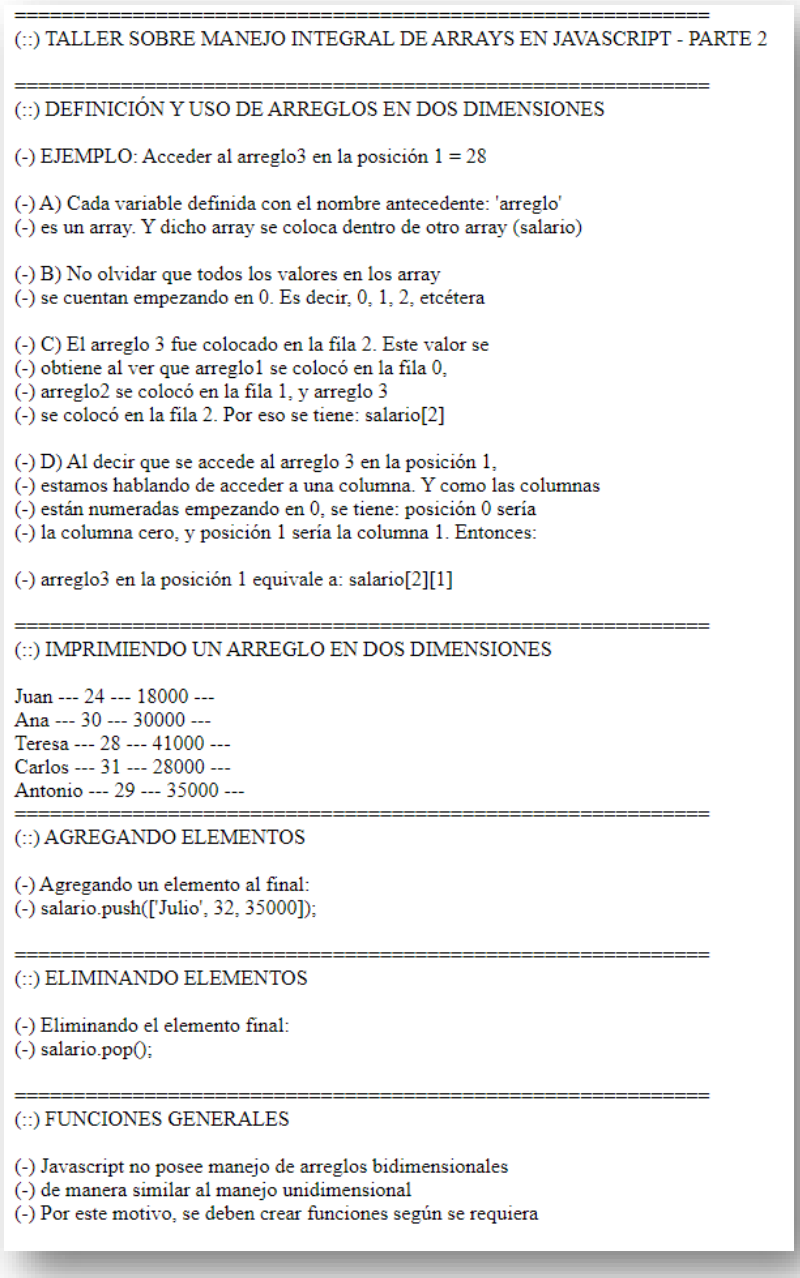
```
<!DOCTYPE HTML>

<html>

<head>
    <title>Arrays-02</title>
    <meta charset="UTF-8"/>
</head>

<body>

<script>

////////////////////////////////////
// ARREGLOS EN JAVASCRIPT - PARTE I //
////////////////////////////////////

// Función para imprimir texto en la pantalla
function texto( cadena, lineas_post = 0 ) {
  if (typeof(cadena) == "string") {
    if (cadena.indexOf("(::)") == 0) {
      document.write("============================================================<br>");
    }
  }
  document.write( cadena );
  for (var i = 0; i < lineas_post; i++) {
    document.write("<br>");
  }
}

function tabla( arr ) {
    for (var i = 0; i < arr.length; i++) {
        for (var j = 0; j < arr[i].length; j++) {
            texto( arr[i][j] + " -- ");
        }
        texto("", 1);
    }
}

texto("(::) TALLER SOBRE MANEJO INTEGRAL DE ARRAYS EN JAVASCRIPT - PARTE 2", 2);

texto("(::) DEFINICIÓN Y USO DE ARREGLOS EN DOS DIMENSIONES", 2);

var arreglo1 = ["Juan", 24, 18000];     // Fila 0

// Juan : Columna 0, 24 : Columna 1, 18000 : Columna 2

// Juan está en: [Fila 0][Columna 0]

// Correctamente escrito: Juan está en: [0][0]

// 24 está en: [0][1]
// 18000 está en: [0][2]
```

```
var arreglo2 = ["Ana", 30, 30000];      // Fila 1
var arreglo3 = ["Teresa", 28, 41000];   // Fila 2
var arreglo4 = ["Carlos", 31, 28000];   // Fila 3
var arreglo5 = ["Antonio", 29, 35000];  // Fila 4

var salario = [arreglo1, arreglo2, arreglo3, arreglo4, arreglo5];

var notas = [

    ['Juan', 4],
    ['Sofía', 5],
    ['Carlos', 3],
    ['Alberto', 5],
    ['María', 5]

];

////////////////////////////////////////////////////////////////////

texto("(-) EJEMPLO: Acceder al arreglo3 en la posición 1 = " + salario[2][1], 2);
texto("(-) A) Cada variable definida con el nombre antecedente: 'arreglo'", 1);
texto("(-) es un array. Y dicho array se coloca dentro de otro array (salario)", 2);

texto("(-) B) No olvidar que todos los valores en los array", 1);
texto("(-) se cuentan empezando en 0. Es decir, 0, 1, 2, etcétera", 2);

texto("(-) C) El arreglo 3 fue colocado en la fila 2. Este valor se", 1);
texto("(-) obtiene al ver que arreglo1 se colocó en la fila 0, ", 1);
texto("(-) arreglo2 se colocó en la fila 1, y arreglo 3", 1);
texto("(-) se colocó en la fila 2. Por eso se tiene: salario[2]", 2);

texto("(-) D) Al decir que se accede al arreglo 3 en la posición 1,", 1);
texto("(-) estamos hablando de acceder a una columna. Y como las columnas ", 1);
texto("(-) están numeradas empezando en 0, se tiene: posición 0 sería", 1);
texto("(-) la columna cero, y posición 1 sería la columna 1. Entonces:", 2);

texto("(-) arreglo3 en la posición 1 equivale a: salario[2][1]", 2);

////////////////////////////////////////////////////////////////////

texto("(::) IMPRIMIENDO UN ARREGLO EN DOS DIMENSIONES", 2);

// Este ciclo es para las filas
for (var fila = 0; fila < salario.length; fila++) {

    // Este ciclo es para las columnas
    for (var columna = 0; columna < salario[fila].length; columna++) {

        // Aquí se accede a cada elemento del array
        texto( salario[fila][columna] + " --- " );

    }
    texto("", 1);
}
```

```
texto("(::) AGREGANDO ELEMENTOS", 2);

texto("(-) Agregando un elemento al final:", 1);
texto("(-) salario.push(['Julio', 32, 35000]); ", 2);

salario.push(['Julio', 32, 35000]);

texto("(::) ELIMINANDO ELEMENTOS", 2);
texto("(-) Eliminando el elemento final:", 1);
texto("(-) salario.pop();", 2);

salario.pop();

texto("(::) FUNCIONES GENERALES", 2);
texto("(-) Javascript no posee manejo de arreglos bidimensionales", 1);
texto("(-) de manera similar al manejo unidimensional", 1);
texto("(-) Por este motivo, se deben crear funciones según se requiera", 2);

////////////////////////////////////////////////////////////////////////7

texto("(::) O B J E T O S", 2);

// Un objeto se define en Javascript utilizando una función

// El siguiente objeto es un punto, con sus coordenadas (x, y)
// en el plano cartesiano

// Los nombres de los objetos tienen la primera letra en mayúscula
// (esto es una convención, no es obligatorio)

// Todos los objetos poseen en su interior datos y otras funciones
// denominados métodos

// Para el caso inicial que nos ocupa, únicamente utilizaremos dos datos:
// La coordenada x
// La coordenada y

function Punto(x,y) {
    // Explicación: this.x es el nombre de las variables dentro del objeto
    //
    // Los nombres de todas las variables internas de un objeto deben
    // poseer la frase: this.

    this.x = x;
    this.y = y;
}

// Se crean cuatro puntos
var p1 = new Punto(1,2);
var p2 = new Punto(3,5);
var p3 = new Punto(2,8);
var p4 = new Punto(4,4);
```

```
163   // Los cuatro puntos se colocan dentro de un array
164   // (Ver la imagen que se anexa en Classroom)
165   var puntos = [p1,p2,p3,p4];
166
167   // Se muestran los cuatro puntos
168   // NOTA: un poco más abajo se escribe una función general para mostrar puntos
169   for (var i = 0; i < puntos.length; ++i) {
170       texto("Punto " + parseInt(i+1) + " = " + puntos[i].x + ", " +
171       puntos[i].y, 1);
172   }
173
174   texto("", 1);
175
176   // La función: dibujarPuntos, presenta los puntos contenidos
177   // en un array (el cual está conformado por puntos)
178   function dibujarPuntos(arr) {
179       // Se recorren todos los puntos contenidos en el arreglo recibido
180       for (var i = 0; i < arr.length; ++i) {
181           // Se muestra el punto iésimo en la pantalla
182           texto("Punto " + parseInt(i+1) + "= " + arr[i].x + ", " + arr[i].y, 1);
183       }
184   }
185
186   // Se agrega un nuevo punto, al final del arreglo 'puntos'
187
188   // Se define el nuevo punto. El punto es p5 con la coordenada (12, -3)
189   var p5 = new Punto(12,-3);
190
191   // Agrega el punto p5 al final del arreglo
192   puntos.push(p5);
193
194   texto("Después de agregar el punto (12, -3) ", 2);
195
196   // Se muestran todos los puntos en la pantalla de la computadora
197   // En el array ya se incluye el último punto adicionado
198   dibujarPuntos(puntos);
199
200   texto("", 1);
201
202   // Con esta instrucción se elimina el primer elemento del arreglo 'puntos'
203   puntos.shift();
204
205   texto("Después de extraer el primero ", 2);
206
207   // Se muestra el array con el punto eliminado
208   dibujarPuntos(puntos);
209
210   </script>
211   </body>
212
213   </html>
```

Cuando se ejecuta este código queda el siguiente diseño de pantalla.



```
============================================================
(::) TALLER SOBRE MANEJO INTEGRAL DE ARRAYS EN JAVASCRIPT - PARTE 2

============================================================
(::) DEFINICIÓN Y USO DE ARREGLOS EN DOS DIMENSIONES

(-) EJEMPLO: Acceder al arreglo3 en la posición 1 = 28

(-) A) Cada variable definida con el nombre antecedente: 'arreglo'
(-) es un array. Y dicho array se coloca dentro de otro array (salario)

(-) B) No olvidar que todos los valores en los array
(-) se cuentan empezando en 0. Es decir, 0, 1, 2, etcétera

(-) C) El arreglo 3 fue colocado en la fila 2. Este valor se
(-) obtiene al ver que arreglo1 se colocó en la fila 0,
(-) arreglo2 se colocó en la fila 1, y arreglo 3
(-) se colocó en la fila 2. Por eso se tiene: salario[2]

(-) D) Al decir que se accede al arreglo 3 en la posición 1,
(-) estamos hablando de acceder a una columna. Y como las columnas
(-) están numeradas empezando en 0, se tiene: posición 0 sería
(-) la columna cero, y posición 1 sería la columna 1. Entonces:

(-) arreglo3 en la posición 1 equivale a: salario[2][1]

============================================================
(::) IMPRIMIENDO UN ARREGLO EN DOS DIMENSIONES

Juan --- 24 --- 18000 ---
Ana --- 30 --- 30000 ---
Teresa --- 28 --- 41000 ---
Carlos --- 31 --- 28000 ---
Antonio --- 29 --- 35000 ---
============================================================
(::) AGREGANDO ELEMENTOS

(-) Agregando un elemento al final:
(-) salario.push(['Julio', 32, 35000]);

============================================================
(::) ELIMINANDO ELEMENTOS

(-) Eliminando el elemento final:
(-) salario.pop();

============================================================
(::) FUNCIONES GENERALES

(-) Javascript no posee manejo de arreglos bidimensionales
(-) de manera similar al manejo unidimensional
(-) Por este motivo, se deben crear funciones según se requiera
```

```
============================================================
(::) O B J E T O S

Punto 1 = 1, 2
Punto 2 = 3, 5
Punto 3 = 2, 8
Punto 4 = 4, 4

Después de agregar el punto (12, -3)

Punto 1= 1, 2
Punto 2= 3, 5
Punto 3= 2, 8
Punto 4= 4, 4
Punto 5= 12, -3

Después de extraer el primero

Punto 1= 3, 5
Punto 2= 2, 8
Punto 3= 4, 4
Punto 4= 12, -3
```

5 MAPA DE CALOR

Más ejercicios sobre el manejo de arrays.

Una empresa quiere saber cómo son las ganancias a lo largo de cuatro[4] semanas en cuatro[4] de sus sucursales. Para ello utilizamos el siguiente código.

```
function texto(cadena, lineas) {
  document.write(cadena);
  for (var i = 0; i < lineas; i++) {
    document.write("<br>");
  }
}

var ganancias = [
  [7, 8, 3, 5],
  [4, 6, 4, 7],
  [5, 7, 4, 2],
  [3, 1, 6, 8],

];

texto("", 1);
texto("GANANCIAS", 2);
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 4; c++) {
    texto(ganancias[f][c] + " ", 0);
  }
  texto("", 1);
}

var mayor = ganancias[0][0];
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 4; c++) {
    if (ganancias[f][c] > mayor)
      mayor = ganancias[f][c];
  }
}


var menor = ganancias[0][0];
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 4; c++) {
    if (ganancias[f][c] < menor)
      menor = ganancias[f][c];
  }
}

texto("", 1);

texto("Ganancia Mayor = " + mayor, 1);
texto("Ganancia Menor = " + menor, 2);

for (var f = 0; f < 4; f++) {
  texto("Semana " + (f + 1) + " = ", 0);
  for (var c = 0; c < 4; c++) {
    texto(ganancias[f][c] + " ", 0);
  }
  texto("", 1);
}
```

```
54
55  texto("Ganancias por semana", 2);
56
57  for (var f = 0; f < 4; f++) {
58    texto("Semana " + (f + 1) + " = ", 0);
59    for (var c = 0; c < 4; c++) {
60      texto(ganancias[f][c] + " ", 0);
61    }
62    texto("", 1);
63  }
64
65  texto("", 1);
66  texto("Ganancias por semana, totalizadas", 2);
67
68  for (var f = 0; f < 4; f++) {
69    texto("Semana " + (f + 1) + " = ", 0);
70    for (var c = 0; c < 4; c++) {
71      texto(ganancias[f][c] + " ", 0);
72    }
73    suma = 0;
74    for (var k = 0; k < 4; k++) {
75      suma = suma + ganancias[f][k];
76    }
77    texto(" Total = " + suma, 1);
78  }
79
80  texto("", 1);
81  texto("Ganancias por sucursal, totalizadas", 2);
82
83  for (var c = 0; c < 4; c++) {
84    texto("Sucursal " + (c + 1) + " = ", 0);
85    for (var f = 0; f < 4; f++) {
86      texto(ganancias[f][c] + " ", 0);
87    }
88    suma = 0;
89    for (var k = 0; k < 4; k++) {
90      suma = suma + ganancias[k][c];
91    }
92    texto(" Total = " + suma, 1);
93  }
94
95  texto("", 1);
96
97  var c = document.getElementById("miCanvas");
98  var ctx = c.getContext("2d");
99
100 for (var f = 0; f < 4; f++) {
101   for (var c = 0; c < 4; c++) {
102     ctx.strokeRect(f*25, c*25, 25, 25);
103     ctx.fillText(ganancias[f][c], f*25+10, c*25+15);
104   }
105 }
```

Cuando ejecutamos este código quedara el siguiente diseño de pantalla.

7	4	5	3
8	6	7	1
3	4	4	6
5	7	2	8

GANANCIAS

7 8 3 5
4 6 4 7
5 7 4 2
3 1 6 8

Ganancia Mayor = 8
Ganancia Menor = 1

Semana 1 = 7 8 3 5
Semana 2 = 4 6 4 7
Semana 3 = 5 7 4 2
Semana 4 = 3 1 6 8
Ganancias por semana

Semana 1 = 7 8 3 5
Semana 2 = 4 6 4 7
Semana 3 = 5 7 4 2
Semana 4 = 3 1 6 8

Ganancias por semana, totalizadas

Semana 1 = 7 8 3 5 Total = 23
Semana 2 = 4 6 4 7 Total = 21
Semana 3 = 5 7 4 2 Total = 18
Semana 4 = 3 1 6 8 Total = 18

Ganancias por sucursal, totalizadas

Sucursal 1 = 7 4 5 3 Total = 19
Sucursal 2 = 8 6 7 1 Total = 22
Sucursal 3 = 3 4 4 6 Total = 17
Sucursal 4 = 5 7 2 8 Total = 22

Con esto concluimos que en la semana “1” fue donde más ganancias hubo, mientras que en la semana “3” y semana “4” fue donde menos ganancias se registran, y en las sucursales “4” y “2” fue quienes más ganancias registraron, pero en la sucursal “1” fue quien menos ganancias tuvo.

La misma empresa ahora quiere saber cómo son las ganancias a lo largo de cuatro[4] semanas en seis[6] de sus sucursales. Para ello utilizamos el siguiente código.

```
function texto(cadena, lineas) {
  document.write(cadena);
  for (var i = 0; i < lineas; i++) {
    document.write("<br>");
  }
}

var ganancias = [
  [7, 8, 3, 5, 9, 5],
  [4, 6, 4, 7, 6, 4],
  [5, 7, 4, 2, 4, 6],
  [3, 1, 6, 8, 5, 8],

];

texto("", 1);
texto("GANANCIAS", 2);
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 6; c++) {
    texto(ganancias[f][c] + " ", 0);
  }
  texto("", 1);
}

var mayor = ganancias[0][0];
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 6; c++) {
    if (ganancias[f][c] > mayor)
      mayor = ganancias[f][c];
  }
}


var menor = ganancias[0][0];
for (var f = 0; f < 4; f++) {
  for (var c = 0; c < 6; c++) {
    if (ganancias[f][c] < menor)
      menor = ganancias[f][c];
  }
}

texto("", 1);

texto("Ganancia Mayor = " + mayor, 1);
texto("Ganancia Menor = " + menor, 2);

for (var f = 0; f < 4; f++) {
  texto("Semana " + (f + 1) + " = ", 0);
  for (var c = 0; c < 6; c++) {
    texto(ganancias[f][c] + " ", 0);
  }
  texto("", 1);
}
```

```
54
55  texto("Ganancias por semana", 2);
56
57  for (var f = 0; f < 4; f++) {
58    texto("Semana " + (f + 1) + " = ", 0);
59    for (var c = 0; c < 6; c++) {
60      texto(ganancias[f][c] + " ", 0);
61    }
62    texto("", 1);
63  }
64
65  texto("", 1);
66  texto("Ganancias por semana, totalizadas", 2);
67
68  for (var f = 0; f < 4; f++) {
69    texto("Semana " + (f + 1) + " = ", 0);
70    for (var c = 0; c < 6; c++) {
71      texto(ganancias[f][c] + " ", 0);
72    }
73    suma = 0;
74    for (var k = 0; k < 4; k++) {
75      suma = suma + ganancias[f][k];
76    }
77    texto(" Total = " + suma, 1);
78  }
79
80  texto("", 1);
81  texto("Ganancias por sucursal, totalizadas", 2);
82
83  for (var c = 0; c < 6; c++) {
84    texto("Sucursal " + (c + 1) + " = ", 0);
85    for (var f = 0; f < 4; f++) {
86      texto(ganancias[f][c] + " ", 0);
87    }
88    suma = 0;
89    for (var k = 0; k < 4; k++) {
90      suma = suma + ganancias[k][c];
91    }
92    texto(" Total = " + suma, 1);
93  }
94
95  texto("", 1);
96
97  var c = document.getElementById("miCanvas");
98  var ctx = c.getContext("2d");
99
100 for (var f = 0; f < 4; f++) {
101   for (var c = 0; c < 6; c++) {
102     ctx.strokeRect(f*25, c*25, 25, 25);
103     ctx.fillText(ganancias[f][c], f*25+10, c*25+15);
104   }
105 }
```

Cuando ejecutamos este código queda el siguiente diseño de pantalla.

7	4	5	3
8	6	7	1
3	4	4	6
5	7	2	8
9	6	4	5
5	4	6	8

GANANCIAS

7 8 3 5 9 5
4 6 4 7 6 4
5 7 4 2 4 6
3 1 6 8 5 8

Ganancia Mayor = 9
Ganancia Menor = 1

Semana 1 = 7 8 3 5 9 5
Semana 2 = 4 6 4 7 6 4
Semana 3 = 5 7 4 2 4 6
Semana 4 = 3 1 6 8 5 8
Ganancias por semana

Semana 1 = 7 8 3 5 9 5
Semana 2 = 4 6 4 7 6 4
Semana 3 = 5 7 4 2 4 6
Semana 4 = 3 1 6 8 5 8

Ganancias por semana, totalizadas

Semana 1 = 7 8 3 5 9 5 Total = 23
Semana 2 = 4 6 4 7 6 4 Total = 21
Semana 3 = 5 7 4 2 4 6 Total = 18
Semana 4 = 3 1 6 8 5 8 Total = 18

Ganancias por sucursal, totalizadas

Sucursal 1 = 7 4 5 3 Total = 19
Sucursal 2 = 8 6 7 1 Total = 22
Sucursal 3 = 3 4 4 6 Total = 17
Sucursal 4 = 5 7 2 8 Total = 22
Sucursal 5 = 9 6 4 5 Total = 24
Sucursal 6 = 5 4 6 8 Total = 23

Con esto concluimos que la semana “1” fue donde hubo más ganancias, mientras que la semana “3” y semana “4” fue donde menos ganancias hubo, también que la sucursal “5” fue quien más ganancias obtuvo, en cambio la sucursal “3” fue donde menos ganancias se obtuvieron.

6 CONCLUSIONES

El aprendizaje sobre el manejo de arrays en el lenguaje JavaScript prueba ser una herramienta de gran valor a la hora de crear código en dicho lenguaje, ya que facilita algunos procedimientos y crea nuevas posibilidades de desarrollo.

7 BIBLIOGRAFÍA

https://repl.it