

**MATRICES Y VECTORES**

**SANTIAGO AMARILES RESTREPO**

**INSTITUCION EDUCATIVA LAS NIEVES**

**MEDIATECNICA**

**10-A**

**ANDERSON**

**20/08/2024**

## INTRODUCCION

Una matriz en programación es una función matemática, vector o matriz con una sola fila o columna.

Las matrices se representan en JavaScript con coordenadas asignadas.

En programación, las coordenadas comienzan desde cero, a diferencia de las matemáticas tradicionales.

Para acceder a los datos en una matriz, se utilizan las coordenadas correspondientes.

En JavaScript, para acceder a un elemento como la manzana en una matriz, se utiliza la notación de corchetes con las coordenadas.

Una matriz es una disposición de información que se puede acceder utilizando corchetes y comas para separar la información.

Las matrices se utilizan para diversos propósitos, como enviar información a través de JSON en servidores web.

Las imágenes también se pueden representar como matrices, con el número de píxeles dentro de ellas.

Las matrices también pueden representar los pesos sinápticos de una red neuronal.

Los vectores son representados como arcos unidimensionales, ya sea horizontal o vertical.

Las matrices son ejemplos de arreglos bidimensionales, con filas y columnas de celdas.

Se menciona que las matrices pueden ser vistas como un conjunto de vectores para poder recorrerlas.

Los vectores pueden ser de tipo entero, real, lógico o caracter.

Cada posición de un vector debe contener el mismo tipo de dato.

Es importante declarar el tipo de dato y el nombre del vector al crearlo.

## Matrices:

Una matriz en programación es una estructura de datos que se utiliza para almacenar una colección de elementos relacionados. Estos elementos pueden ser números, letras, objetos o cualquier otro tipo de dato. Lo que hace que las matrices sean poderosas es que los elementos se organizan en filas y columnas, lo que facilita el acceso y la manipulación de los datos de manera estructurada.

En términos más simples, puedes pensar en una matriz como una tabla o una cuadrícula en la que cada celda contiene un valor. Cada celda está identificada por su posición en la fila y la columna.

## EJEMPLO:

Algoritmo MostrarAlReves

Dimension datos[6]

Para i <- 1 Hasta 6 Hacer

    Escribir "Dime el dato numero ", i

    Leer datos[i]

FinPara

Escribir "Los datos al revés son: "

Para i <- 6 Hasta 1 Con Paso -1 Hacer

    Escribir datos[i]

FinPara

FinAlgoritmo

## Vectores:

Los vectores son estructuras de datos similares a los arreglos, pero más desarrollados, ya que entre otras cosas, crecen y decrecen dinámicamente, según se necesite. En algunos lenguajes, el tamaño de un arreglo queda fijo en tiempo de compilación. En otros lenguajes, la dimensión del arreglo, queda fijada en tiempo de ejecución. No obstante, una vez fijada, no puede alterarse. La real necesidad es que la estructura de datos pueda ajustar su capacidad dinámicamente durante todo el tiempo de ejecución. En inglés esta estructura de datos es *vectors*.

La estructura de datos que puede crecer y decrecer dinámicamente, en todo momento, según las exigencias de ejecución se llama vector. La estructura de datos vector representa un conjunto de objetos. El conjunto de objetos es de tamaño variable. Se incorporan objetos hasta colmar la capacidad total del vector. Cuando se requiera incorporar un nuevo objeto en un vector lleno, el vector se expande automáticamente, según la capacidad incremental definida. La capacidad inicial, por default, es 10. La capacidad incremental, por default, es el doble de la existente en el momento de la expansión. Si no se desea este crecimiento exponencial, se puede especificar la capacidad incremental, y en esa capacidad crecerá el vector cada vez que se expanda. Cuando la capacidad del vector está agotada, el vector se redimensiona y se reubica automáticamente. La redimensión del vector acontece como se mencionó más arriba, según se haya explicitado o no una capacidad incremental. En el momento de la redimensión se crea un nuevo vector. A continuación se reubican todos los elementos del vector viejo, copiándolos en el nuevo, para finalmente ubicar el nuevo elemento que provocó la expansión, en el vector nuevo.

#### EJEMPLO:

```

1. ALGORITMO Vectores_1;
2. VAR
3.     ENTERO mi_vector[10];
4.     ENTERO contador;
5. INICIO
6.
7.     PARA contador DESDE 1 HASTA 10
8.         ESCRIBIR("Dime un número");
9.         LEER( mi_vector[contador] );
10.    FIN_PARA
11.
12.    PARA contador DESDE 1 HASTA 10
13.        ESCRIBIR( "El valor de la posición [", contador, "] es: ",
mi_vector[contador] );
14.    FIN_PARA

```

15.

16. FIN

## CONCLUSION

- Pueden ser unidimensionales o bidimensionales.
- Se comparan con cajas vacías donde se puede guardar y reemplazar información.
- Las matrices son ejemplos de arreglos bidimensionales, con filas y columnas de celdas.
- Se menciona que las matrices pueden ser vistas como un conjunto de vectores para poder recorrerlas.
- Cada posición de un vector debe contener el mismo tipo de dato.
- Es importante declarar el tipo de dato y el nombre del vector al crearlo.
- Se pueden asignar valores a los arreglos, representando cada valor como si estuviera en una caja.
- Los arreglos permiten realizar diversas operaciones, como recorrerlos y ordenarlos, al igual que con las variables.
- Los vectores permiten almacenar múltiples valores del mismo tipo en una sola variable.

- Se menciona que en la siguiente clase se verá cómo recorrer, intercambiar y manipular valores en los vectores.

EJM:

1. Si queremos acceder al elemento que está en

la fila 0, columna 1 hacemos:

```
>>>M[0][1]
```

```
>>>2
```

2. Si quiero una matriz de 2\*2 inicializada en ceros .

```
>>>M=([0,0],[0,0])
```

```
>>>M[0]
```