

# MATRICES Y VECTORES

Anderson Camacho

Jaider Álvarez

Media técnica 10°A

**¿Qué es una matriz?:** Una matriz es un arreglo bidimensional de números o símbolos organizados en filas y columnas. Se representa con una letra mayúscula (por ejemplo,  $A$ ) y sus elementos se agrupan entre paréntesis o corchetes. Cada elemento de la matriz se denota con un doble subíndice: el primero indica la fila y el segundo la columna.

**¿Qué ES UN VECTOR?:** Un **vector** es una estructura de datos dinámica que puede crecer y decrecer automáticamente según sea necesario, a diferencia de los arreglos que tienen un tamaño fijo o predefinido. A medida que se añaden elementos a un vector y se llena su capacidad, el vector se expande automáticamente. Por defecto, la capacidad inicial de un vector es 10 y se duplica en cada expansión, aunque esto puede ajustarse a una capacidad incremental específica. Durante la expansión, se crea un nuevo vector con mayor capacidad, se copian los elementos del vector antiguo y se añade el nuevo elemento. A diferencia de los arreglos que son de un solo tipo, los vectores pueden almacenar objetos de diferentes tipos, permitiendo una mayor flexibilidad.

## **¿Para Qué Sirve una Matriz en Programación?**

Las matrices son herramientas versátiles y se utilizan en una variedad de aplicaciones en programación:

### **1. Almacenamiento de Datos Tabulares**

Las matrices se utilizan para almacenar datos tabulares, como hojas de cálculo. Pueden representar información en forma de filas y columnas, lo que facilita la gestión de datos estructurados.

### **2. Procesamiento de Imágenes y Gráficos**

En aplicaciones de procesamiento de imágenes y gráficos, las matrices se utilizan para representar imágenes y aplicar operaciones como el filtrado y la transformación.

### **3. Juegos y Simulaciones**

En juegos y simulaciones, las matrices se utilizan para representar el estado de un tablero, la posición de los personajes y otros elementos del juego.

### **4. Álgebra Lineal**

En matemáticas y ciencias de la computación, las matrices se

utilizan en álgebra lineal para resolver sistemas de ecuaciones lineales y realizar transformaciones lineales.

## ¿Para Qué Sirve un vector en Programación?

En programación, un **vector** (también conocido como **array dinámico** o **vector dinámico**) sirve para varias finalidades importantes debido a su capacidad de ajustar su tamaño de manera automática. Aquí te detallo algunas de sus principales aplicaciones y ventajas:

### 1. Almacenamiento de Datos Dinámicos

- **Capacidad de Ajuste Automático:** Los vectores pueden crecer y decrecer según sea necesario, permitiendo almacenar una cantidad variable de elementos sin necesidad de definir un tamaño fijo previamente.

### 2. Manipulación y Acceso a Elementos

- **Acceso Aleatorio:** Permiten acceder a cualquier elemento mediante un índice, facilitando operaciones de lectura y escritura rápidas.
- **Inserción y Eliminación:** Facilitan la inserción y eliminación de elementos en cualquier posición, con la capacidad de reorganizar automáticamente su estructura.

### 3. Flexibilidad en el Manejo de Tipos de Datos

- **Tipos de Datos Diversos:** En muchos lenguajes, los vectores pueden contener elementos de distintos tipos, lo que proporciona flexibilidad para manejar diversos tipos de datos en una sola estructura.

### 4. Eficiencia en la Implementación de Algoritmos

- **Algoritmos Eficientes:** Son útiles en la implementación de varios algoritmos y estructuras de datos, como listas dinámicas, colas y pilas, que requieren un manejo eficiente del tamaño de los datos.

### 5. Aplicaciones en Gráficos y Computación Numérica

- **Representación de Datos:** En gráficos y computación numérica, los vectores pueden representar una serie de datos que necesitan ser manipulados y procesados, como en cálculos de transformaciones o en la representación de puntos y vectores en gráficos 2D y 3D.

### 6. Almacenamiento de Secuencias de Datos

- **Listas de Elementos:** Son ideales para almacenar secuencias de elementos que necesitan ser procesados en conjunto, como en el caso de colecciones de objetos, resultados de consultas o datos leídos de archivos.

## 7. Estructuras de Datos de Alto Nivel

- **Implementación de Otras Estructuras:** Pueden ser utilizados para implementar otras estructuras de datos más complejas, como listas enlazadas, tablas hash y otros contenedores de dato

### Ejemplo pseudocodigo matriz

// Definición de una matriz de tamaño 3x3

Definir matriz A[3][3]

Definir matriz B[3][3]

Definir matriz C[3][3]

// Inicializar las matrices A y B con valores

Para i desde 0 hasta 2 hacer

Para j desde 0 hasta 2 hacer

$A[i][j] = (i + 1) * (j + 1)$  // Asignar valores a A

$B[i][j] = (i + 1) + (j + 1)$  // Asignar valores a B

Fin Para

Fin Para

// Sumar las matrices A y B y almacenar el resultado en C

Para i desde 0 hasta 2 hacer

Para j desde 0 hasta 2 hacer

$C[i][j] = A[i][j] + B[i][j]$  // Sumar los elementos correspondientes

Fin Para

Fin Para

```
// Mostrar el resultado de la matriz C

Para i desde 0 hasta 2 hacer

    Para j desde 0 hasta 2 hacer

        Imprimir "C[" , i, "]" , j, "] = " , C[i][j]

    Fin Para

Fin Para
```

### **Ejemplo pseudocodigo vector**

```
ALGORITMO Vectores_1;

VAR

    ENTERO mi_vector [10];

    ENTERO contador;

INICIO

    PARA contador DESDE 1 HASTA 10

        ESCRIBIR("Dime un número");

        LEER( mi_vector[contador] );

    FIN_PARA

    PARA contador DESDE 1 HASTA 10

        ESCRIBIR( "El valor de la posición [" , contador, "] es: " , mi_vector[contador] );

    FIN_PARA

FIN
```

