

Arrays Bidimensionales o Matrices

Definición

Un **array bidimensional** es una estructura de datos que almacena valores en una **tabla** de filas y columnas.

Imaginate una **matriz** (como una hoja de Excel) con filas y columnas.

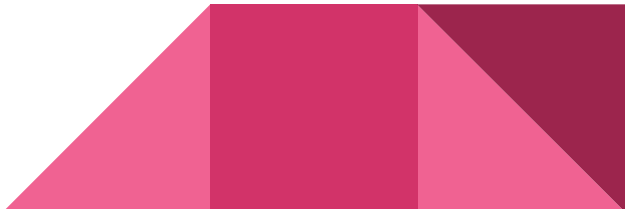
Visual: Mostrar un ejemplo gráfico de una matriz 2x3.

| 1 | 2 | 3 |

| 4 | 5 | 6 |

Conceptos Clave:

- **Filas:** representadas por el primer índice.
- **Columnas:** representadas por el segundo índice.



Declaración

C permite arrays de varias dimensiones, típicamente 2D (pueden ser más).

```
int matriz[2][3] = {  
    {1, 2, 3},  
    {4, 5, 6}  
};
```

En el primer corchete ponemos la cantidad de filas y en el segundo la cantidad de columnas

visualmente:

Fila 0: [1 2 3]

Fila 1: [4 5 6]

*El número de filas y columnas se debe especificar.



Acceso a los elementos de la matriz

La sintaxis para acceder a un elemento es:

`matriz[filas][columna]`

Por ejemplo si tenemos:

```
int matriz[2][3] = {
```

```
    {1, 2, 3},
```

```
    {4, 5, 6}
```

```
};
```

```
printf("%d", matriz[1][2]); *** ¿qué imprime?
```



Cómo lo recorreremos...

```
for (int i = 0; i < 2; i++) {           //el for de afuera se mete en la fila...  
    for (int j = 0; j < 3; j++) {       //el de adentro se mete en la columna...  
        printf("%d ", matriz[i][j]);    //i es la fila j la columna...  
    }  
    printf("\n");                       //este salto de línea es clave  
}
```

esto va a imprimir nuestra matriz :

1 2 3

4 5 6

para cargar datos es igual, solo que debemos usar scanf**

Pasando una matriz a funciones

```
void procesarMatriz(int m[ ][4], int filas);
```

Importante: ¡Debe indicarse el número de columnas!

¿Por qué se necesitan las columnas?

Porque la memoria en C es lineal, y al acceder a `m[i][j]`, el compilador necesita saber cuántos elementos hay por fila para calcular la posición real en memoria:

```
void miFuncion(int m[ ][ ]);          // !ERROR! Falta información.
```

