

FUNDAMENTOS DE PROGRAMACIÓN

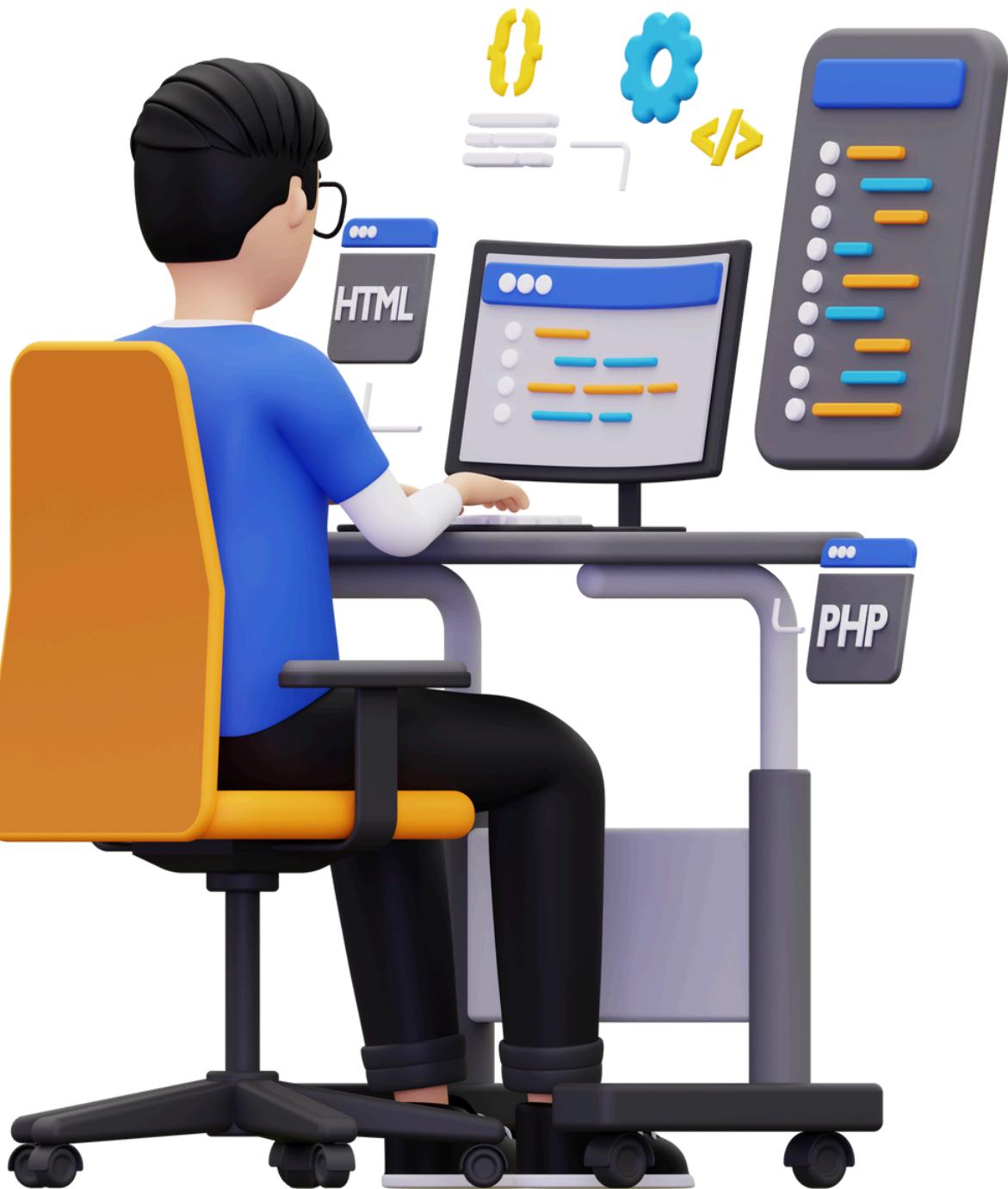


FUNDAMENTOS DE PROGRAMACIÓN



¿QUÉ VAMOS A VER?

- Arreglos bidimensionales.
- Asignar valores a una matriz.
- Ingreso masivo de datos.
- Lectura de los datos de una matriz.





ARREGLOS BIDIMENSIONALES

- También conocidos como **matriz**.
- Un arreglo bidimensional almacena los datos de manera general $N \times M$, en dónde N corresponde al número de filas y M al número de columnas.
- De forma gráfico una matriz se puede visualizar como una tabla en dónde cada dato almacenado va a depender directamente de la fila y la columna en que fue almacenado.

Vendedor 1
Vendedor 2
Vendedor 3
Vendedor 4
Vendedor 5
Vendedor 6
Vendedor 7
Vendedor n

	Dia 1	Dia 2	Dia 3	Dia 4	Dia M
Vendedor 1	Y	YYY			
Vendedor 2	XXXX				
Vendedor 3			ZZZZ		
Vendedor 4					
Vendedor 5		AAA			
Vendedor 6					
Vendedor 7					
Vendedor n					

```
1 Proceso EjemploMatriz
2
3     Definir datos Como Entero;
4
5     Dimension datos[3, 3];
6
7 FinProceso
```



ASIGNAR VALORES A UNA MATRIZ

- Es necesario indicar el índice de la fila y columna que se quiere modificar.
- Importante a tener en cuenta de que el dato que se le quiera almacenar debe ser igual al tipo de dato que se definió para la matriz.
- **Ejemplo:**
 - Crearemos una matriz de 3 filas y 3 columnas.
 - En primera instancia le asignaremos valores de manera directa.

```
1  Proceso EjemploMatriz
2
3      Definir datos Como Entero;
4      Dimension datos[3,3];
5
6      datos[0,0] = 5;
7      datos[0,1] = 10;
8      datos[0,2] = 15;
9
10     datos[1,0] = 20;
11     datos[1,1] = 25;
12     datos[1,2] = 30;
13
14     datos[2,0] = 35;
15     datos[2,1] = 40;
16     datos[2,2] = 45;
17
18     Escribir "El primer dato de la segunda columna es: ", datos[0,1];
19     Escribir "El último dato de la tercera columna es: ", datos[2,2];
20
21 FinProceso
```

	0	1	2
0	5	10	15
1	20	25	30
2	35	40	45



INGRESO MASIVO DE DATOS

- La asignación manual o directa de una matriz se puede tornar muy engorrosa a medida que la dimensión va creciendo.
- La mejor manera de solucionar esa problemática es utilizando ciclos anidados para poder recorrer correctamente ambas dimensiones de la matriz.
- **Ejemplo:**
 - Realizar un llenado celda a celda masivo de una matriz de 3 filas y 4 columnas.

```
1 Proceso EjemploMatriz
2
3     Definir matriz, f, c Como Entero;
4     Dimension matriz[3,4];
5
6     Para f = 0 Hasta 2 Con Paso 1 Hacer
7         Para c = 0 Hasta 3 Con Paso 1 Hacer
8             Escribir "Ingresa un valor para la fila ", f, " y columna ", c, ": ";
9             Leer matriz[f,c];
10            FinPara
11        FinPara
```



LECTURA DE LOS DATOS DE UNA MATRIZ

- Al igual que para llenar de datos una matriz, lo mejor para recorrerla y leer sus datos, es utilizar ciclos anidados.
- Normalmente se prefiere el ciclo “Para... Hacer” por la simplicidad en su sintaxis.
- **Ejemplo:**
 - Despliegue de datos de una matriz de 3 filas y 4 columnas.

```
13  Para f = 0 Hasta 2 Con Paso 1 Hacer
14    Para c = 0 Hasta 3 Con Paso 1 Hacer
15      Escribir "Fila ", f, ", Columna ", c, ": ", matriz[f,c];
16    FinPara
17  FinPara
```



BÚSQUEDA EN UNA MATRIZ

- Para buscar un dato en específico lo mejor siempre es recorrer la matriz para validar paso a paso la similitud.

```
1  Proceso EjemploMatriz
2
3      Definir f, c, cantidad Como Entero;
4      Definir letras, buscado Como Caracter;
5      Dimension letras[5,3];
6
7      Para f = 0 Hasta 4 Con Paso 1 Hacer
8          Para c = 0 Hasta 2 Con Paso 1 Hacer
9              Leer letras[f,c];
10             FinPara
11         FinPara
12
13     buscado = "0";
14     cantidad = 0;
15
16     f = 0;
17     Repetir
18         c = 0;
19         Repetir
20             Si letras[f,c] = buscado Entonces
21                 cantidad = cantidad + 1;
22                 FinSi
23                 c = c + 1;
24             Hasta Que c = 3;
25             f = f + 1;
26         Hasta Que f = 5;
27
28     Escribir "La cantidad de veces que se encontró el caracter ", buscado, " es: ", cantidad;
29
30 FinProceso
```



EJERCICIO PRACTICO

- Desarrolle un algoritmo en pseudocódigo que rellene automáticamente una matriz de $N*M$ con la suma de sus índices, es decir, el elemento $A[1,1] = 1 + 1 = 2$.
- Una vez la matriz se encuentre completa, se debe mostrar por pantalla.





EJERCICIO PRACTICO

- En una tabla de 4 filas y 4 columnas se guardan las notas de 4 alumnos de secundaria.
- Cada fila corresponde a las notas y al promedio de cada alumno.
- Se necesita un programa que permita a un profesor cargar, en las 3 primeras posiciones (columnas) de cada fila, las notas del alumno y que en la última columna se calculen los promedios.
- Una vez realizados los cálculos, se desea mostrar las 3 notas de cada alumno y el promedio correspondiente recorriendo la matriz.





EJERCICIO PRACTICO

- Crea una matriz de $N*M$ tipo Entero.
- Aprovecha las cualidades la función AZAR para llenar la matriz en su totalidad de datos.
- Una vez los datos estén almacenados, recorre la matriz y muestra por pantalla el número mayor y menor almacenado.

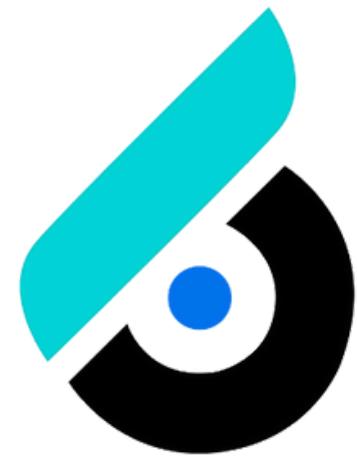




EJERCICIO PRACTICO

- Crea un arreglo bidimensional que contenga la tabla de multiplicar del 1 al 9.
- Ten en cuenta que tanto la primera fila como la primera columna, deben contener los números del 0 al 9 respectivamente.





AWAKELAB

#programmingbootcamp

nodovirtual.awakelab.cl

 jELOU futurejob by  Adalid Chile