

UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA
VICERRECTORADO ACADÉMICO
DECANATO DE DOCENCIA
DEPARTAMENTO DE ING. EN INFORMÁTICA

UNIDAD CURRICULAR: **COMPUTACIÓN I**

(Código: 0415102T)

Clase No. 8

*Unidad IV: Arreglos en Lenguaje C
(Arreglos Bidimensionales)*

Profesor: Armando Carrero

UNIDAD No. IV

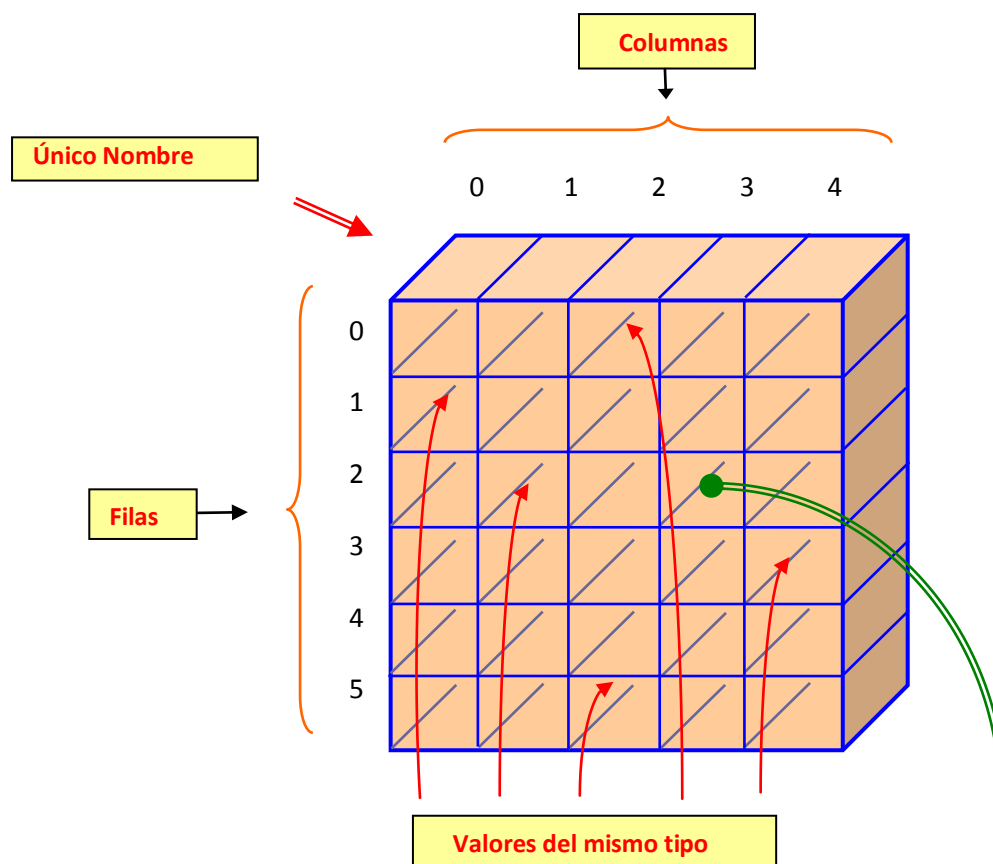
ARREGLOS EN LENGUAJE C

Arreglos Bidimensionales

También se les conoce como **Tabla** o **Matriz**.

Representación Gráfica de un Arreglo Unidimensional

Una variable tipo matriz, se puede representar gráficamente como se muestra a continuación:



Cada uno de los valores se identifica mediante: el **nombre**, el **subíndice** de la **fila** y el de la **columna**, así por ejemplo:

NOMBRE [2][3]

Observación: En lenguaje C, los subíndices para acceder a la fila y a la columna de un arreglo bidimensional debe iniciar en cero.

DECLARACIÓN DE UNA VARIABLE TIPO ARREGLO BIDIMENSIONAL

Formato para declarar arreglos bidimensionales (Matriz) de: Enteros, Reales, Caracteres

tipo **nombre de la variable** $\left[\begin{array}{c} \text{Dimensión} \\ \text{de} \\ \text{Filas} \end{array} \right] \left[\begin{array}{c} \text{Dimensión} \\ \text{de} \\ \text{Columnas} \end{array} \right] ;$

Para declarar un arreglo bidimensional de cadena de caracteres:

char **nombre de la variable** $\left[\begin{array}{c} \text{Dimensión} \\ \text{de} \\ \text{Filas} \end{array} \right] \left[\begin{array}{c} \text{Dimensión} \\ \text{de} \\ \text{Columnas} \end{array} \right] \left[\begin{array}{c} \text{Cantidad} \\ \text{Máxima de} \\ \text{caracteres}+1 \end{array} \right] ^{\circ})$

Nota: Esta declaración es solo a nivel informativo, ya que no se trataran problemas donde se trabaje con matrices de cadenas de caracteres.

ACCESO A UNA VARIABLE TIPO ARREGLO BIDIMENSIONAL

Para acceder a cualquier valor de un arreglo bidimensional, debe hacerse referencia al nombre de la variable, al subíndice que posiciona la fila y al subíndice que posiciona la columna, así:

nombre de variable **[no. fila][no. columna]** ;

Ejemplo: Se tiene la producción que cada uno de 10 operarios, produjo en cada uno de los días 5 días de la semana. Declarar una variable tipo matriz para almacenar ordenadamente para cada operador las producciones en cada uno de los días.

Por conveniencia se considera que los operadores define el número de filas y los días, el número de las columnas. De igual forma supongamos, que la producción se mide en kilogramos, es decir que las cantidades pueden tener parte decimal.

Declaración:

```
float produccion [ 10 ][ 5 ] ;
```

La variable producción tipo matriz, puede visualizarse así:

produccion

días

		0	1	2	3	4
o p e r a d o r	0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					

CARGA O LECTURA DE UN ARREGLO BIDIMENSIONAL

Es necesario utilizar dos estructuras de repetición, anidadas, una para generar el subíndice de las filas y la otra para generar el subíndice de las columnas. Así, existen dos formas:

FORMA A: Procedimiento de carga por FILAS

```
for ( i = 0 ; i < 10 ; i ++ )
{
    for ( k = 0 ; k < 5 ; k ++ )
    {
        printf ( "\n Ingrese la producción del operario No. %d el día %d ", i+1 , k+1 ) ;
        scanf ( " %f " , &producción [ i ] [ k ] ) ;
    }
}
```

Ciclo Externo para manejar las filas

Ciclo Interno para manejar las columnas

Clave: Ciclo externo, genera el subíndice de las filas

FORMA B: Procedimiento de carga por COLUMNAS

```
for ( k = 0 ; k < 5 ; k ++ )
{
    for ( i = 0 ; i < 10 ; i ++ )
    {
        printf ( "\n Ingrese la producción del operario No. %d el día %d ", i+1 , k+1 ) ;
        scanf ( " %f " , &producción [ i ] [ k ] ) ;
    }
}
```

Ciclo Externo para manejar las columnas

Ciclo Interno para manejar las filas

Clave: Ciclo externo, genera el subíndice de las columnas

PROBLEMA DIDACTICO

El centro de meteorología legal, ha recopilado la presión en Kg/cm^2 , en distintas lugares del estado Táchira, a distintas horas del día (Hora militar 0-24). Los datos han sido arreglados de la forma siguiente:

Lugares	Horas				
	8:30	9:15	10:45	13:25.....	18:20
Táriba	xx	xx	xx	xx	xx
Michelena	xx	xx	xx	xx	xx
Orope	xx	xx	xx	xx	xx
Coloncito	xx	xx	xx	xx	xx
Siberia	xx	xx	xx	xx	xx
.
.
El Zumbador	Xx	xx	xx	xx	xx

Codifique un programa en Lenguaje C, que utilice arreglos unidimensionales (vectores) y bidimensionales (Matrices), para :

- Cargar en arreglos los datos conocidos, es decir: lugares (Vector), horas (Vector) y presiones (Matriz).
- Calcular y mostrar la cantidad total de lecturas tomadas.
- Mostrar las presiones de: el lugar ubicado en primera y última posición del arreglo y que fue tomada en la segunda lectura.
- Calcule una presión promedio general, para el estado, en base a todas las lecturas.
- Calcular y mostrar la presión promedio en cada hora.
- Cargar en un vector la presión promedio para cada lugar.
- Determine si en Lobatera se midió la presión, de ser cierto encuentre la hora a la que se registró la menor presión.
- Encontrar el lugar o lugares en los que ocurrió la presión mas alta, medida a una hora determinada por el usuario.
- Permitir la consulta de la presión de un lugar y una hora de interés por parte del usuario.

PROGRAMA CODIFICADO EN LENGUAJE C

//Programa centro de meteorología legal

//Autor: Armando Carrero Fecha: Lugar :

//Archivos de Cabecera

```
#include <stdio.h>
#include <ctype.h>
#include <windows.h>
```

```
#define F 25
```

```
#define C 10
```

//Codigo para usar gotoxy en DevC

```
void gotoxy(int x,int y)
{
    HANDLE hCon;
    hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x;
    dwPos.Y = y;
    SetConsoleCursorPosition(hCon,dwPos);
}
```

```
int main ( )
{
    //Declaración de variables, se asume un máximo de 25 lugares y 10
    // horas distintas para las mediciones

    char    lugar [ F ] [ 16 ] , hora [ C ] [ 6 ] , resp , p ;

    int     z = 0 , nh , i , j , h , a , m , k ;

    float   presion [ F ] [ C ] , suma_presion = 0 , presion_promedio , suma_hora ;

    float   pres_prom_hora , pres_prom_lugar [ F ] = { 0 } ;
```

// a.- Carga o Lectura de datos

// Carga de Lugares.

```
do
{
    printf ( "\n Ingrese el nombre del lugar No. %d ", z + 1 );
    gets ( lugar [ z ] );
    z ++ ;
    printf ( "\n ¿ Otro lugar por almacenar ? S o N " );
    resp = getchar ( );
    resp = toupper ( resp );
} while ( resp == 'S' && z < F );
```

// Luego de ésta lectura, el valor de la variable z contiene el número exacto de
// lugares, es decir, el número exacto de filas de la matriz

// Carga de las horas. Usando otra modalidad: Se solicita el numero de // horas de medición, se valida que no exceda el máximo de 10 y se carga // el vector usando un ciclo automático)

```
do // ciclo para solicitar y validar el número de horas
{
    printf ( "\n Ingrese el número de horas en que se midió la presión, máximo : %d ", C );
    scanf ( "%d" , &nh );
    if ( nh < 1 || nh > C )
        printf ( "\n Su valor no es válido, recuerde máximo : %d ", C );
} while ( nh < 1 || nh > C );

for ( i = 0 ; i < nh ; i ++ )
{
    printf ( "\n Ingrese la hora de medición No. %d ", i + 1 );
    gets ( hora [ i ] );
}
```


// Carga de la matriz presión, (Por Filas)

// Numero de Filas = z ; Numero de Columnas = nh

```
for ( i = 0 ; i < z ; i ++ )
    for ( j = 0 ; j < nh ; j ++ )
    {
        printf ( "\n Ingrese la presión medida en %s a las %s ", lugar [ i ] , hora [ j ] );
        scanf ( "%f" , & presion [ i ] [ j ] );
    }

printf ( "\n\n" ); system ( "pause" );
```

// b.- Total de lecturas tomadas

```
system ( "cls" ); gotoxy(12, 12) ;
printf ( " Total de Lecturas Tomadas = %d ", z * nh );
printf ( "\n\n" ); system ( "pause" );
```

// c.- Mostrar las presiones

**// Presion medida en el lugar ubicado en primera posición del arreglo y que
// fue tomada en la segunda lectura.**

```
printf ( "\n Presión medida en el primer lugar y que fue tomada en la " );
printf ( " segunda lectura = %.2f " , presion [ 0 ] [ 1 ] );
```

```
printf ( "\n\n" ); system ( "pause" );
```

**// Presión medida en el lugar ubicado en última posición del arreglo y que fue
// tomada en la segunda lectura.**

```
printf ( "\n Presión medida en el último lugar y que fue tomada en la " );
printf ( " segunda lectura = %.2f " , presion [ z - 1 ] [ 1 ] );
```

```
printf ( "\n\n" ); system ( "pause" );
```

// d.- Presión promedio general, para el estado

```
for ( i = 0 ; i < z ; i ++ )           // acumulación de todas las presiones
    for ( j = 0 ; j < nh ; j ++ )
        suma_presion += presion [ i ] [ j ];

presion_promedio = suma_presion / ( z * nh );
printf ( "\n La Presión Promedio para el estado es de = %.3f " , presion_promedio );
printf ( "\n\n" );    system ( "pause" );
```

// e.- Presión promedio en cada hora.

```
system ( "cls" );

gotoxy ( 20, 15 );
printf ( "HORA" );

gotoxy ( 30, 15 );
printf ( "PRESIÓN PROMEDIO" );

for ( j = 0 ; j < nh ; j ++ )    //recorrido de la matriz por columnas
{
    suma_presion = 0 ;

    for ( i = 0 ; i < z ; i ++ )
        suma_presion += presion [ i ] [ j ];

    pres_prom_hora = suma_presion / z ;
    gotoxy ( 20, 17 + j );
    printf ( " %s " , hora [ j ] );
    gotoxy ( 30, 17 + j );
    printf ( " %10.2f " , pres_prom_hora );
}

printf ( "\n\n" );
system ( "pause" );
```

// f.- Presión promedio en cada Lugar.

```
system ("cls");

for ( i = 0 ; i < z ; i ++ )      //recorrido de la matriz por filas
{
    for ( j = 0 ; j < nh ; j ++ )
        pres_prom_lugar [ i ] += presion [ i ] [ j ] ;
    pres_prom_lugar [ i ] = pres_prom_lugar [ i ] / nh ;
}

gotoxy ( 20, 15 ) ;
printf ("LUGAR" ) ;
gotoxy ( 30, 15 ) ;
printf ("PRESIÓN PROMEDIO" ) ;

for ( i = 0 ; i < z ; i ++ )
{
    gotoxy ( 20, 17 + i ) ;
    printf (" %s " , lugar [ i ] ) ;
    gotoxy ( 30, 17 + i ) ;
    printf (" %10.2f " , pres_prom_lugar [ i ] ) ;
}

printf ("\n\n");
system ("pause") ;
```

**// g.- Determine si en Lobatera se midió la presión, de
// ser cierto encuentre la hora a la que se registró la
// menor presión.**

```
p = 'A' ;           // Centinela o suiche en estado Abierto
a = 0;              // Contador Inicializado

do                  // búsqueda de Lobatera en el vector lugar
{
    if ( strcmpi ( "Lobatera" , lugar [ a ] ) == 0)
    {
        p = 'C';      // Centinela o suiche Cerrado
        m = a ;        // Se guarda el subíndice que representa la fila de
                        // la matriz
    }
    a ++;
} while ( a < z && p == 'A' );

if ( p == 'C' )
{
    h = presion [ m ] [ 0 ];
    for( k=0; k < nh ; k++)
        if ( presion [ m ] [ k ] < h)
            h = presion [ m ] [ k ] ;

    printf ("\n\n"); system ("pause") ;

    printf ("\n\n\t\t Hora o horas , en donde se registro la menor presión" ) ;
    printf (" en Lobatera");

    for( k=0 ; k < nh ; k++)
        if ( presion [ m ] [ k ] == h)
            printf ("\n\t\t %s " , hora [ k ] ) ;

}
else
    printf ("\n\t DISCULPE, EN LOBATERA NO SE MIDIO LA PRESION");

printf ("\n\n"); system ("pause") ;
return 0;

} // Fin de la función main. Fin del programa.
```