

# PROGRAMACIÓN

**Unidad 5 - Parte 2:** Tipos de datos derivados: arreglos. Uso de funciones de la biblioteca estándar. Arreglos y funciones.

# Repasemos lo visto



# Cadenas –Arreglos de caracteres



- Este es el tipo de arreglo mas popular en código C.
- La ultima celda del vector de caracteres se reserva para el carácter que marca el fin de la cadena, que es el carácter nulo: “\0”.

# Inicialización explícita de una cadena

Los valores a asignar deben estar encerrados entre llaves y separados por comas (lista).

```
1  #include <stdio.h>
2  #define MAX 45
3
4  int main(void)
5  {
6      char apellido[MAX] = {'P','e','r','e','z','\0'};
7      // agrega al final el correspondiente cero de terminación al final de la cadena
8      char nombre[MAX] = "Marcelo";
9      char unt[12]= {'U','n','i','v','e','r','s','i','d','a','d'};
10
11      ...
12
```

# Funciones de biblioteca y arreglos

## Función de entrada para cadena de caracteres:

**gets(arre):** almacena datos ingresados desde stdin a la cadena denominada arre. Un carácter ingresado `\n` de nueva línea se convierte en un cero de terminación (`\0`)

## Función de salida para cadena de caracteres:

**puts(arre):** encamina la cadena arre hacia stdout. Un cero de terminación (`\0`) al final de la cadena se convierte en un carácter de nueva línea (`\n`).

# Ejemplos

```
1  #include <stdio.h>
2  #define TAMA 50
3  int main()
4  {
5      char nombre[TAMA];
6
7      printf("Ingrese su nombre: ");
8      gets(nombre);
9
10     printf("El nombre ingresado es:");
11     puts(nombre);
12
13     return 0;
14 }
```

Ingrese su nombre: Ana Sofia  
El nombre ingresado es: Ana Sofia

# Funciones de biblioteca <string.h>

Mencionaremos solo algunas funciones.

Prototipo	Descripción
<code>int strlen(cad1)</code>	Retorna la longitud de cad1
<code>int strcmp(cad1, cad2)</code>	Compara cad1 con cad2, carácter a carácter
<code>char *strcat(cad1, cad2)</code>	Concatena la cad2 al final de la cad1. Retorna cad1.

# Funciones y Arreglos

```
1  #include <stdio.h>
2  #define TAMA 10
3
4  void Inicializar(int arre[TAMA] , int cant);
5  void CargarArreglo(int arre[TAMA], int cant);
6
7  int main()
8  {
9      int arre[TAMA];
10     int cant;
11
12     printf("Ingrese la cantidad de Elementos (<10):");
13     scanf("%d", &cant);
14
15     Inicializar(arre, cant);
16     CargarArreglo(arre, cant);
17
18     for(int i=0; i< cant; i++)
19     {
20         printf("\n arre[%d] = %d", i, arre[i]);
21     }
22
23     return 0;
24 }
```

```
void CargarArreglo(int arre[TAMA], int cant)
{
    int i;
    for(i=0; i<cant; i++)
    {   arre[i]= i * 3; }
}

void Inicializar(int arre[TAMA] , int cant)
{
    int i;
    for(i=0; i<cant; i++)
    {   arre[i]=0;      }
}
```



# A modo de resumen



Los arreglos, como una unidad, no admiten:

- ❑ Asignación directa entre ellos
- ❑ Operaciones aritméticas directas
- ❑ Comparaciones directas
- ❑ Devolución como valor de devolución de una función