



# 11074 – Programación I

División Computación – Departamento de Ciencias Básicas



## ARREGLOS DE CADENAS

El tipo de dato **cadena**, al menos en los estándares de **C**, no existe.

Una cadena **es un arreglo de caracteres**, o sea, del tipo **char**.

Por lo tanto, para declarar un arreglo de cadenas o strings en C necesitamos un **arreglo de caracteres bidimensional**, porque tendrá arreglos dentro de un arreglo.

Veamos cómo **declararlo, calcular su tamaño, recorrerlo y pasarlo como argumento a una función en C**.

# Declaración de un arreglo de cadena

Definimos la máxima longitud que puede tener la cadena, por ejemplo

```
#define MAXIMA_LONGITUD_CADENA 50
```

Y declarar el arreglo así:

```
char arreglo[ ][MAXIMA_LONGITUD_CADENA] = {"Arsenal", "Bautizo", "Comadreja", "Consulta"};
```

De esta manera puedes cambiar la longitud rápidamente.

# Calcular tamaño de arreglo de cadenas en C

Igualmente usamos **sizeof** para calcular el tamaño, como lo vimos hace algún tiempo.

Para la longitud de un arreglo de cadenas nada cambia, así se calcula:

```
int longitudDelArreglo = sizeof(arreglo) / sizeof(arreglo[0]);
```

Un programa que imprime su longitud y que nos sirve de ejemplo es este:

```
// Trabajando con arreglos de cadenas en C

#include <stdio.h>
#define MAXIMA_LONGITUD_CADENA 50
int main(){
    char arreglo[][MAXIMA_LONGITUD_CADENA] = {"Arsenal", "Bautizo", "Comadreja",
    "Consulta", "Dinosaurio", "Zapato"};
    int longitudDelArreglo = sizeof(arreglo) / sizeof(arreglo[0]);
    printf("El arreglo mide: %d\n", longitudDelArreglo);
}
```

# Recorrer arreglo de cadenas en C

Igual que en los otros arreglos vistos anteriormente. Aquí el código necesario, es un ciclo **for** a partir de que ya tenemos el tamaño o longitud del arreglo. Y dentro del arreglo hay cadenas.

```
#include <stdio.h>
#define MAXIMA_LONGITUD_CADENA 50

int main(){
    char arreglo[][MAXIMA_LONGITUD_CADENA] = {"Arsenal", "Bautizo", "Comadreja",
        "Consulta", "Dinosaurio", "Zapato"};
    int longitudDelArreglo = sizeof(arreglo) / sizeof(arreglo[0]);
    printf("El arreglo mide: %d\n", longitudDelArreglo);
    for(int x = 0; x < longitudDelArreglo; x++){
        printf("En %d el arreglo tiene el valor %s\n", x, arreglo[x]);
    }
}
```

# Pasar un arreglo de cadenas como argumento a una función en C

Veremos cómo podemos llamar a una función y pasarle nuestro arreglo de cadenas; dentro de la función simplemente lo vamos a recorrer. Si queremos recibir un arreglo, debemos definir sus límites al igual que cuando lo declaramos.

```
void funcion(char arreglo[ ][20]){  
}
```

No debemos especificar la longitud del mismo, pero sí de lo que contiene. Para llamar a una función, primero declaramos el arreglo y se lo pasamos sin más. Siguiendo el ejemplo anterior:

```
//Trabajando con arreglos de cadenas en C: Pasarlo como argumento a función
```

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

```
#define MAXIMA_LONGITUD_CADENA 50
```

```
void imprimirArregloDeCadenas(char arreglo[][MAXIMA_LONGITUD_CADENA], int longitudDelArreglo);
```

```
int main(){
```

```
    char arreglo[][MAXIMA_LONGITUD_CADENA] = {"Arsenal", "Bautizo", "Comadreja", "Consulta",  
    "Dinosaurio", "Zapato"};
```

```
    int longitudDelArreglo = sizeof(arreglo) / sizeof(arreglo[0]);  
    imprimirArregloDeCadenas(arreglo, longitudDelArreglo);
```

```
}
```

```
void imprimirArregloDeCadenas(char arreglo[][MAXIMA_LONGITUD_CADENA], int longitudDelArreglo){
```

```
    for(int x = 0; x < longitudDelArreglo; x++){  
        printf("En %d el arreglo tiene el valor %s\n", x, arreglo[x]);
```

```
    }
```

```
}
```

// Enseñar no es transferir conocimientos, sino crear las posibilidades de su construcción; quien enseña aprende al enseñar y quien aprende enseña al aprender”



**Paulo Freire**