

Cadenas en C

¿Qué es una cadena de caracteres en C?

En C, **no existe un tipo de dato string** como en lenguajes como Python o Java.

En su lugar, se usan **arreglos de caracteres** (`char[]`) terminados en `'\0'` (carácter nulo).

```
char saludo[] = "Hola";
```

```
// Equivalente a: {'H', 'o', 'l', 'a', '\0'}
```



¿Por qué no hay strings nativos?

C es un lenguaje de **bajo nivel y cercano al hardware**.

Está diseñado para ser **eficiente y simple**, lo que implica menos abstracciones.

La falta de un tipo string es intencional: el programador controla cómo se almacenan y manipulan los datos.



¿Cómo se representan las cadenas?

Una cadena en C es:

- Un **puntero** a un bloque de memoria con caracteres.
- Terminada por un **carácter nulo** ('\0').

```
char *nombre = "Ana"; // Constante de cadena
```



¿Cómo se pasan cadenas a funciones?

Se pasan como **punteros** (`char *` o `char[]`).

```
void saludar(char *nombre) {  
    printf("Hola, %s!\n", nombre);  
}
```



¿Qué pasa al modificar cadenas?

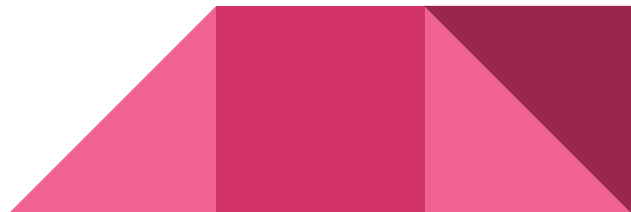
Cuidado con modificar **constantes de cadena** (como "Ana" visto en slide anterior).

- Son **inmutables** en muchas implementaciones.

Usa `char arreglo[]` si necesitas modificarla:

```
char nombre[] = "Ana";
```

```
nombre[0] = 'J'; // Ahora es "Jna"
```



Funciones comunes para trabajar con cadenas (<string.h>)

`strlen(cadena)` – longitud

`strcpy(dest, src)` – copia

`strcmp(cad1, cad2)` – comparar

`strcat(dest, src)` – concatenar



Buenas prácticas

Siempre reservar espacio extra para el '\0'.

Usar funciones seguras (strncpy, etc.).

Si trabajas mucho con cadenas, considera usar una **librería externa**

Recordar que '\0' ocupa 1 byte y es **parte de la cadena**.

Las cadenas se representan como arrays en memoria RAM, byte por byte.

Se pasan a funciones como punteros (char *).

Modifica solo cadenas que se hayan declarado como arreglos (char nombre[] = ...).

