

# Chuleta de Manejo de Cadenas en C

En el lenguaje C, las cadenas de caracteres son arreglos de caracteres terminados en un carácter nulo ('\0').

A continuación, se presentan algunas de las funciones más comunes para el manejo de cadenas en C y ejemplos de uso.

## 1. strlen()

La función strlen() se utiliza para obtener la longitud de una cadena. No cuenta el carácter nulo.

Sintaxis:

```
size_t strlen(const char *str);
```

Ejemplo:

```
char str[] = "Hola Mundo";  
  
int longitud = strlen(str);
```

## 2. strcpy()

La función strcpy() copia una cadena de origen en una cadena de destino.

Sintaxis:

```
char *strcpy(char *dest, const char *src);
```

Ejemplo:

```
char src[] = "Hola";  
  
char dest[10];
```

```
strcpy(dest, src);
```

### 3. strcat()

La función strcat() concatena (añade) una cadena al final de otra.

Sintaxis:

```
char *strcat(char *dest, const char *src);
```

Ejemplo:

```
char str1[20] = "Hola";  
char str2[] = " Mundo";  
strcat(str1, str2); // str1 ahora es "Hola Mundo"
```

### 4. strcmp()

La función strcmp() compara dos cadenas. Devuelve 0 si las cadenas son iguales, un valor negativo si la primera es menor, y un valor positivo si la primera es mayor.

Sintaxis:

```
int strcmp(const char *str1, const char *str2);
```

Ejemplo:

```
char str1[] = "Hola";  
char str2[] = "Mundo";  
int resultado = strcmp(str1, str2);
```

### 5. sprintf()

La función sprintf() permite formatear una cadena con variables y guardarla en otra cadena.

Sintaxis:

```
int sprintf(char *str, const char *format, ...);
```

Ejemplo:

```
int edad = 25;
```

```
char buffer[50];
```

```
sprintf(buffer, "Tengo %d años", edad);
```

*Esta chuleta incluye las funciones más comunes para el manejo de cadenas en C. ¡Úsala como referencia rápida!*