



ANEXO

Cadenas de caracteres al estilo de C

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores

Luis Hernández Yáñez/Pablo Moreno Ger
Facultad de Informática
Universidad Complutense



Índice

Cadenas al estilo de C	582
E/S con cadenas al estilo de C	583
La biblioteca <code>cstring</code>	584
Ejemplo	585



Cadenas de caracteres al estilo de C

Arrays de caracteres terminados en nulo

```
const Max = 15;  
typedef char tCadena[Max];  
tCadena cadena = "Adiós"; // Inicialización al declarar
```

Siempre hay al final un carácter nulo (código ASCII 0 - '\0')

Indica que en esa posición termina la cadena (exclusive)

cadena	A	d	i	ó	s	\0									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

En el array caben MAX-1 caracteres significativos

Longitud máxima de la variable cadena: 14

No se pueden asignar cadenas literales: ~~cadena = "Hola";~~

Ni copiar cadenas directamente: ~~cad2 = cad1;~~

Ni comparar con op. relacionales: ~~if (cad1 < cad2) ...~~



Entrada/salida con cadenas al estilo de C

```
tCadena cadena;
```

```
cin >> cadena; // Se añade un nulo al final
```

Extractor: la lectura termina en el primer espacio en blanco

¡No se comprueba si se leen más caracteres de los que caben!

setw(): máximo de caracteres a colocar (incluyendo el nulo)

```
cin >> setw(15) >> cadena;
```

```
cin.getline(cadena_estilo_C, máx):
```

Para leer también los espacios en blanco y no más de *máx-1*

```
cin.getline(cadena, 15); // Hasta 14 caracteres
```

```
cout << cadena << endl; // El nulo no se muestra
```



```
cin.getline(cad, máx)  
getline(cin, cad)
```

Cadenas al estilo de C
Cadenas de tipo string



La biblioteca cstring

- ✓ `strlen(cadena)`: longitud actual de la *cadena*
`cout << "Longitud: " << strlen(cadena);`
- ✓ `strcpy(destino, origen)`: copia *origen* en *destino*
`strcpy(cad2, cad1); strcpy(cad, "Me gusta C++");`
- ✓ `strcat(destino, origen)`: añade *origen* al final de *destino*
`tCadena cad1 = "Hola", cad2 = "Adiós";`
`strcat(cad1, cad2); // cad1 contiene "HolaAdiós"`
- ✓ `strcmp(cad1, cad2)`: compara lexicográficamente las cadenas
0 si son iguales, 1 si `cad1 > cad2` ó -1 si `cad1 < cad2`
`tCadena cad1 = "Hola", cad2 = "Adiós";`
`strcmp(cad1, cad2) // Devuelve 1 ("Hola" > "Adiós")`

...

<http://www.cplusplus.com/reference/cstring/>



Ejemplo de cadenas al estilo de C

cadena.cpp

```
#include <iostream>
using namespace std;
#include <cstring>

int main() {
    const int MAX = 20;
    typedef char tCad[MAX];
    tCad cadena = "Me gusta C++";
    cout << cadena << endl;
    cout << "Cadena: ";
    cin >> cadena; // Lee hasta el primer espacio en blanco
    cout << cadena << endl;
    cin.sync(); // Sincronizar la entrada
    cout << "Cadena: ";
    cin.getline(cadena, MAX);
    cout << cadena << endl;
    cout << "Longitud: " << strlen(cadena) << endl;
    strcpy(cadena, "Hola");
    ...
}
```



Ejemplo de cadenas al estilo de C

```
tCad cadena2 = " amigo";
strcat(cadena, cadena2);
cout << cadena << endl;
if (strcmp(cadena, cadena2) == 0) {
    cout << "Iguales";
}
else if (strcmp(cadena, cadena2) > 0) {
    cout << cadena << " es mayor que " << cadena2;
}
else {
    cout << cadena << " es menor que " << cadena2;
}
cout << endl;

return 0;
}
```



Acerca de *Creative Commons*



Licencia CC (*Creative Commons*)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

-  Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
-  No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.
-  Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Pulsa en la imagen de arriba a la derecha para saber más.

