

Cadenas de caracteres.

***Programación I – Laboratorio I.
Tecnicatura Superior en Programación.
UTN-FRA***

Autores: *Pablo Gil
Hector Farina*

Revisores: *Ing. Ernesto Gigliotti
Lic. Mauricio Dávila*

Versión : 1



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](http://creativecommons.org/licenses/by-sa/4.0/).

Índice de contenido

| | |
|--|---|
| 1Cadenas de caracteres..... | 3 |
| 1.1Inicialización de cadenas de caracteres..... | 3 |
| 1.2Funciones usadas para trabajar con cadenas de caracteres..... | 3 |
| 1.2.1SCANF..... | 4 |
| 1.2.2GETS (GET String)..... | 4 |
| 1.2.3STRCMP (STRing CoMParison)..... | 5 |
| 1.2.4STRCPY (STRing CoPY)..... | 6 |
| 1.2.5STRLEN (STRing LENght)..... | 6 |

1 Cadenas de caracteres

Hasta el momento no teníamos la posibilidad de ingresar texto, ya sea nombres, apellidos, direcciones, comentarios o cualquier otro dato que contenga más de una letra.

Una cadena de caracteres es una palabra o frase (un texto cualquiera). Todos los lenguajes de programación necesariamente deben poder trabajar con este tipo de datos. Puesto que en C no existe un tipo de dato "string" o "cadena" que nos permita almacenar más de una letra, lo que se hace es guardar la frase o palabra en un vector de caracteres.

De acuerdo a esto, cada vez que necesite trabajar con datos que sean cadenas de caracteres, se deberá definir un vector de caracteres.

En el lenguaje C las cadenas se representan como una secuencia de letras terminadas en un CERO (0 o ASCII '\0'), por lo tanto si ingresamos la palabra "hola" en el vector se cargará lo siguiente:

| | | | | |
|-----|-----|-----|-----|------|
| 'h' | 'o' | 'l' | 'a' | '\0' |
|-----|-----|-----|-----|------|

Es importante recordar que para trabajar con cadenas debe definir un vector y que el tamaño del vector debe ser uno más que la máxima cantidad de letras a guardar.

Todas las tareas que se realicen con cadenas serán hechas por medio de funciones realizadas a tal fin.

A continuación se detallarán las funciones básicas más comunes para trabajar con cadenas de caracteres.

1.1 Inicialización de cadenas de caracteres

Al igual que podemos inicializar en algún valor una variable del tipo entero, se puede inicializar una cadena de caracteres.

Algunas de las formas de hacerlo son:

```
char saludo[10]="hola";  
char saludo[10]={'h','o','l','a','\0'};  
char saludo[ ]="hola";  
char menu [ ][3]={"Archivo","Nuevo","Abrir"};
```

1.2 Funciones usadas para trabajar con cadenas de caracteres

Se describirán solo las funciones usadas durante el curso. Es de hacer notar que todas las funciones que se explicarán en este capítulo llevan como parámetros la dirección de comienzo de la cadena, que por el momento lo interpretaremos de la siguiente forma:

Como la cadena está guardada en un vector, se coloca solamente el nombre del vector (no se tiene en cuenta el índice).

1.2.1 SCANF

La función "scanf" permite realizar el ingreso de datos según el formato que se le indique en uno de sus parámetros.

Para el uso con cadenas de caracteres se deberá usar **%s**

```
scanf( "%s" , vec );
```

Los dos argumentos de la función son:

1. "%s" : Indica que lo que se ingrese sea interpretado como cadena de caracteres
2. "vec" : Dirección de comienzo del vector donde se va a guardar la palabra (nombre del vector sin usar "&")

"scanf" lee los caracteres que se ingresan por teclado hasta que se presiona un espacio o enter. Lo que se leyó lo guarda a partir de la dirección indicada por vec y coloca el terminador nulo en lugar del enter.

Tenga en cuenta que con esta función no se pueden ingresar palabras separadas por espacio ya que corta el ingreso de datos cuando detecta un espacio en blanco.

Ejemplo: Cargar 1 nombre y mostrarlo por pantalla

```
#include <stdio.h>

void main(void)
{
    char texto[20];
    printf("\nIngrese nombre: ");
    scanf("%s", texto);

    printf("\nNombre: %s ",texto);
}
```

1.2.2 GETS (GET String)

La función gets permite leer desde el teclado cadenas de caracteres.

```
gets (vec);
```

El argumento es la dirección de comienzo del vector donde se va a guardar la palabra. La función lee los caracteres que se ingresan por teclado hasta que se presiona un enter. Lo que se leyó lo guarda a partir de la dirección indicada por vec y coloca el terminador nulo en lugar del enter.

Ejemplo: Cargar 1 nombre y mostrarlo por pantalla

```
void main(void)
{
    char texto[20];
    printf("\nIngrese nombre: ");
    gets(texto);

    printf("\nNombre: %s ",texto);
}
```

1.2.3 STRCMP (STRing CoMParison)

La función strcmp sirve para comparar cadenas de caracteres.

```
int strcmp ( cad1 , cad2 )
```

- 1er argumento: Dirección de comienzo del vector donde esta la otra palabra a comparar
- 2do argumento: Dirección de comienzo del vector donde esta una de las palabras a comparar

Como resultado de la comparación la función devuelve un valor entero que se interpreta de la siguiente manera

- Si devuelve menor a cero significa que la palabra en cad1 es menor que que la palabra en cad2
- Si devuelve cero significa que que la palabra en cad1 es la misma que que la palabra en cad2
- Si devuelve mayor a cero significa que la palabra en cad1 es mayor que que la palabra en cad2

Se debe tener en cuenta que esta función diferencia las mayúsculas de las minúsculas, por lo tanto encontrará como distintas las palabras HOLA y hola.

Si se quiere comparar sin importar mayúscula o minúscula se debe usar **stricmp**.

Ejemplo:Cargar 1 password en una cadena y compararlo con "1234"

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

void main(void)
{
    char texto[20];
    printf("\nIngrese el password: ");
    gets(texto);

    if(strcmp(texto,"1234")==0)
        printf("Password correcto");
    else
        printf("Password incorrecto");
}
```

1.2.4 STRCPY (STRing CoPY)

La función strcpy sirve para copiar una cadena de caracteres desde un lugar a otro , es decir desde un origen a un destino.

`strcpy (destino , origen)`

- 1er argumento: Dirección de comienzo del vector donde se van a guardar los datos
- 2do argumento: Dirección de comienzo del vector que tiene los datos a copiar

La función copia la cadena de caracteres que se encuentra guardada en un vector (cuya dirección de comienzo es origen) hacia otro vector (cuya dirección de comienzo es destino).

Ejemplo: Cargar una palabra en una cadena, copiarla a otra cadena auxiliar, e imprimir ambas.

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char texto1[20];
    char texto2[20];
    printf("\nIngrese una palabra: ");
    gets(texto1);

    strcpy(texto2,texto1);

    printf("Se ingreso:%s la copia es:%s",texto1,texto2);
}
```

1.2.5 STRLEN (STRing LENght)

La función strlen sirve para obtener la cantidad de letras que tiene una cadena de caracteres. Esta función cuenta los caracteres hasta que llega al 0(terminador de cadena), es decir , si ingresamos la siguiente cadena "Hola como estas" , la función devolvera 15 que es la cantidad de caracteres que tiene la cadena (los espacios tambien son contados).

`int strlen (vec)`

El argumento de la función es la dirección de comienzo del vector que tiene los datos.

Ejemplo: Cargar un vector con una palabra y mostrar la cantidad de caracteres que tiene.

```
#include <stdio.h>
#include <string.h>
#define MAX 10

void main(void)
{
    char vec[MAX];
    int i,x;

    printf("\ningrese nombre: ");
    gets(vec);
    x=strlen (vec);
    printf("\nla cantidad de letras es %d: ",x);
}
```