

Universidad de Guanajuato División de Ingenierías  
Campus Irapuato Salamanca (DICIS)

Algoritmos y estructura de datos  
Carlos Hugo García Capulín

Tarea No. 5  
Cadenas de caracteres

Jair Chávez Islas  
09/Septiembre/2021

# Problema

En esta ocasión necesitamos hacer un programa que lea una cadena de caracteres y muestre ciertos datos a continuación de la cadena.

Datos como: Numero de caracteres por palabra, Número de caracteres en total y Número de palabras.

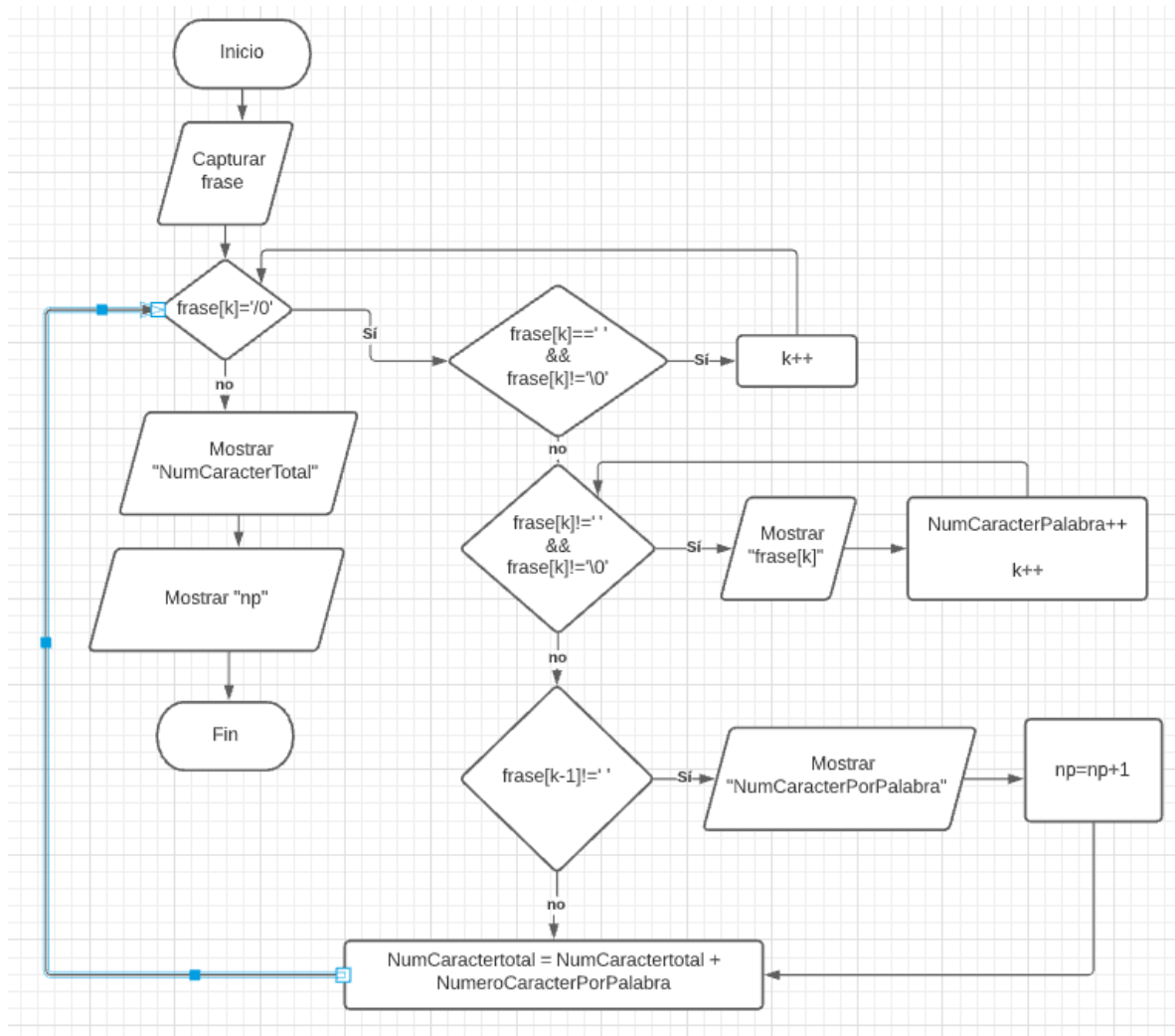
Como ejemplo tenemos lo siguiente.

“Saludos a todos los presentes” Analizando esta frase, podemos recolectar los siguientes datos:

Saludos	7
a	1
todos	5
los	3
presentes	9
Total de palabras:	5
Total de caracteres:	25

# Solución implementada

Diagrama del problema

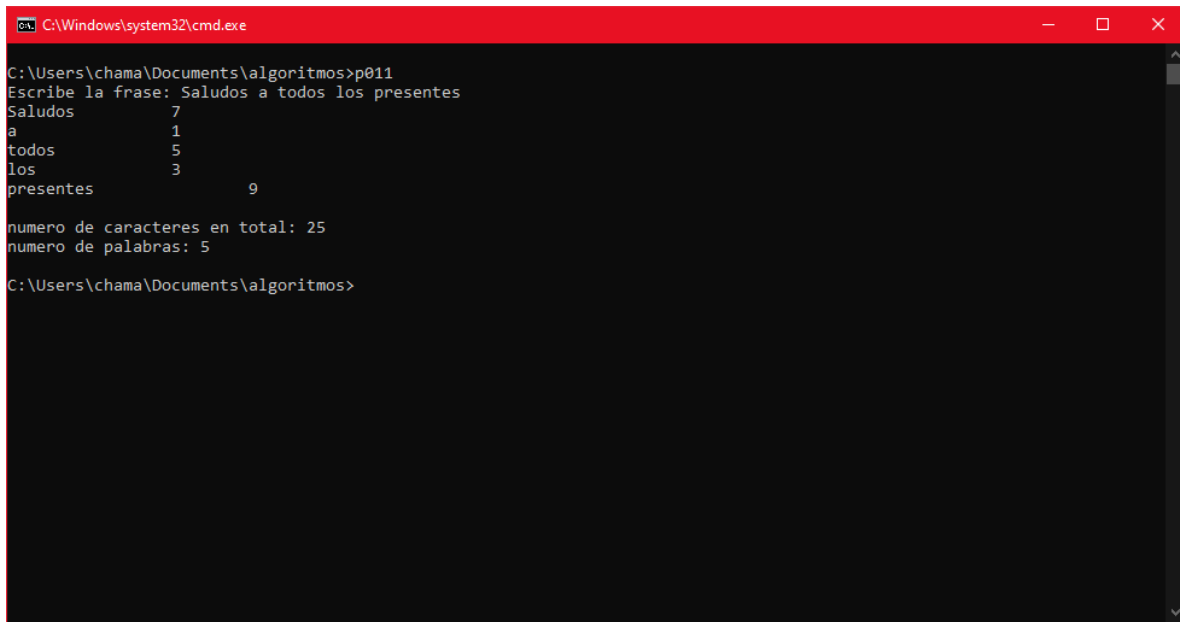


## Código comentado del problema

```
1  #include <stdio.h> //Agregamos la librería stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3  int main() //Inicia función Principal
4  {   char frase[200]; //Se declaran las variables.
5      unsigned int k=0, NumCaracterPorPalabra, NumCaracterTotal, np;
6
7
8      printf("Escribe la frase: "); //Se le pide al usuario que capture una cadena
9      scanf("%[^\n]s", frase); //La cadena se guarda en el array
10
11     NumCaracterTotal = 0;
12     np = 0; //igualamos estas dos variables en cero.
13
14     //En este ciclo recorre la cadena desde el primer caracter hasta el ultimo.
15     while(frase[k]!='\0')
16     {
17
18         //Ciclo que esquiva los espacios hasta encontrar el primer caracter.
19         while((frase[k]==' ') && (frase[k]!='\0'))
20             k++;
21
22         //Ciclo que contabiliza los caracteres hasta encontrar el siguiente espacio.
23         NumCaracterPorPalabra = 0;
24         while((frase[k]!=' ') && (frase[k]!='\0'))
25         {   putchar(frase[k]);
26             NumCaracterPorPalabra++;
27             k++;
28             //Mientras que no se encuentre ningún espacio, el numero de caracter por palabra va aumentando
29         }
30         if(frase[k-1]!=' ')
31         {
32             printf("\t\t %u\n", NumCaracterPorPalabra); // \t << tabulacion, genera "tabla"
33             //En este punto sumaremos np+1, que ira guardando el número de palabras de la cadena
34             np=np+1;
35         }
36
37         NumCaracterTotal = NumCaracterTotal + NumCaracterPorPalabra;
38         //En esta variable se irán sumando los caracteres de todas las palabras.
39     }
40
41     printf("\nnumero de caracteres en total: %u", NumCaracterTotal); //Imprimimos el numero de caracteres totales.
42     printf("\nnumero de palabras: %u", np); //Guardamos el número total de palabras
43
44     printf("\n");
45     return 0;
46 }
```

# Pruebas y resultados

## Evidencia del problema



```
C:\Windows\system32\cmd.exe
C:\Users\chama\Documents\algoritmos>p011
Escribe la frase: Saludos a todos los presentes
Saludos      7
a            1
todos        5
los          3
presentes    9

numero de caracteres en total: 25
numero de palabras: 5
C:\Users\chama\Documents\algoritmos>
```

Ejecutando el problema, tenemos que, de una manera efectiva, nos arroja los datos que ya habíamos calculado nosotros manualmente con la misma frase.

“Saludos a todos los presentes”

Saludos	7
a	1
todos	5
los	3
presentes	9
Total de palabras:	5
Total de caracteres:	25