



# PROGRAMACIÓN ESTRUCTURADA

PROF. PEDRO NÚÑEZ YEPIZ

REPORTE DE PRACTICA #7  
CADENAS

KEVIN ALEJANDRO GONZALEZ TORRES  
372354 – GPO 932

## **INTRODUCCIÓN**

En este reporte se mostrara el uso de cadenas, de sus distintos usos en diversas aplicaciones diarias y de como funcionan las funciones de la principal libreria que trabaja con cadenas <string.h>

## **COMPETENCIA**

Se practicará el uso de las cadenas.

## **FUNDAMENTOS**

Strings:

<https://www.geeksforgeeks.org/strings-in-c/>

<https://www.programiz.com/c-programming/c-strings>

## PROCEDIMIENTO

1.- Función que reciba como parámetro una cadena y la convierta a MAYUSCULAS

```
109 char *uppercaseString(char string[])
110 {
111     for (int i = 0; string[i] != '\0'; i++)
112     {
113         if (string[i] >= 'a' && string[i] <= 'z')
114         {
115             string[i] -= 32;
116         }
117     }
118
119     return string;
120 }
```

2.- Función Que reciba como parámetro una cadena y la convierta a MINUSCULAS

```
122 void lowercaseString(char string[])
123 {
124     for (int i = 0; string[i] != '\0'; i++)
125     {
126         if (string[i] >= 'A' && string[i] <= 'Z')
127         {
128             string[i] += 32;
129         }
130     }
131 }
```

3.- Función que reciba como parámetro una cadena y la convierta a CAPITAL

```
133 void capitalString(char string[])
134 {
135     uppercaseString(string);
136
137     for (int i = 1; string[i] != '\0'; i++)
138     {
139         if (string[i - 1] == ' ')
140         {
141             if (string[i] >= 'a' && string[i] <= 'z')
142             {
143                 string[i] -= 32;
144             }
145         }
146         else
147         {
148             if (string[i] >= 'A' && string[i] <= 'Z')
149             {
150                 string[i] += 32;
151             }
152         }
153     }
154 }
```

4.-Función que reciba como parámetro una cadena y retorne la cantidad de caracteres que tiene la cadena

```
156 int lengthString(char string[])
157 {
158     int length = 0;
159     while (string[length] != '\0')
160     {
161         length++;
162     }
163     return length;
164 }
```

- 5.- Función que reciba como parámetro una cadena y retorne una cadena con sus caracteres acomodados de forma inversa (al revés)

```
166 char *reversedString(char string[])
167 {
168     int length = strlen(string);
169     int reverse = 0;
170     char temp;
171
172     for (int i = 0; i < length / 2; i++)
173     {
174         temp = string[i];
175         string[i] = string[length - i - 1];
176         string[length - i - 1] = temp;
177     }
178
179     return string;
180 }
```

- 6.-Función que reciba como parámetro una cadena y genere una nueva cadena basada en la original pero sin espacios

```
182 void noSpacesString(char string[])
183 {
184     int i, no_space_count = 0;
185
186     for (i = 0; string[i] != '\0'; i++)
187     {
188         if (string[i] != ' ')
189         {
190             string[no_space_count++] = string[i];
191         }
192     }
193
194     string[no_space_count] = '\0';
195 }
```

7.-Función que sirva para leer una cadena y solo permita caracteres alfabéticos (A...Z) y el espacio, donde una cadena no puede comenzar o terminar con espacio, no debe tener dos espacios seguidos. retornar la cadena ya sea como parámetro o variable

```
197 void alphabeticalString(char string[])
198 {
199     int i, asd = 0;
200     for (i = 0; string[i] != '\0'; i++)
201     {
202         if ((string[i] >= 'a' && string[i] <= 'z') || string[i] == ' ')
203         {
204             string[asd++] = string[i];
205         }
206     }
207     string[asd] = '\0';
208 }
209
```

8.-Función que reciba como parámetro una cadena, y utilizando las funciones anteriores, imprima en MAYUSCULAS, MINUSCULAS , CAPITAL, SIN ESPACIOS, ALREVES la cadena original

```
210 void manyFunctions(char string[])
211 {
212     uppercaseString(string);
213     printf("Entered uppercase string is: %s\n", string);
214     lowercaseString(string);
215     printf("Entered lowercase string is: %s\n", string);
216     capitalString(string);
217     printf("Entered string capitalized is: %s\n", string);
218     noSpacesString(string);
219     printf("String with spaces removed: %s\n", string);
220     reversedString(string);
221     printf("Entered reversed string is: %s\n", string);
222 }
```

9.-Función que reciba como parámetro una cadena, y desplegar la leyenda si la cadena es un palíndromo SI o NO

```
224  ∨ int isPalindrome(char string[])
225      {
226          noSpacesString(string);
227          int length = lengthString(string);
228          int left = 0;
229          int right = length - 1;
230
231  ∨  while (left < right)
232      {
233  ∨      if (string[left] != string[right])
234          {
235              return 0;
236          }
237          left++;
238          right--;
239      }
240
241  ∨  return 1;
242  }
243
```