



## Seminario de Algoritmia

# REPORTE DE PRÁCTICA

## IDENTIFICACIÓN DE LA PRÁCTICA

Práctica	4	Nombre de la práctica	Búsqueda de cadena
Fecha	09/09/2021	Nombre del profesor	Alma Nayeli Rodríguez Vázquez
Nombre del estudiante	1. Cárdenas Pérez Calvin Cristopher		
	2. Farfán de León José Osvaldo		
	3. García Martínez Noe Aaron		

## OBJETIVO

El objetivo de esta práctica consiste en implementar el algoritmo de búsqueda de cadena.

## PROCEDIMIENTO

Realiza la implementación siguiendo estas instrucciones.

Implementa el algoritmo de búsqueda de cadena utilizando Matlab y C++ / Python. Para la implementación, utiliza una cadena de por lo menos 7 palabras. Apóyate en el siguiente algoritmo:

```
ALGORITHM BruteForceStringMatch( $T[0..n-1]$ ,  $P[0..m-1]$ )
//Implements brute-force string matching
//Input: An array  $T[0..n-1]$  of  $n$  characters representing a text and
//      an array  $P[0..m-1]$  of  $m$  characters representing a pattern
//Output: The index of the first character in the text that starts a
//      matching substring or  $-1$  if the search is unsuccessful
for  $i \leftarrow 0$  to  $n - m$  do
     $j \leftarrow 0$ 
    while  $j < m$  and  $P[j] = T[i + j]$  do
         $j \leftarrow j + 1$ 
    if  $j = m$  return  $i$ 
return  $-1$ 
```

## IMPLEMENTACIÓN

Agrega el código de tu implementación aquí.

```
T = ['e', 's', 't', 'e', ' ', 'e', 's', ' ', 'e', 'l', '...',
    ' ', 'a', 'l', 'g', 'o', 'r', 'i', 't', 'm', 'o', ' ', 'd', 'e', ' ', 'l', 'a', '...',
    ' ', 'c', 'a', 'd', 'e', 'n', 'a'];
P = ['c', 'a', 'd', 'e', 'n', 'a'];

indice = stringMatching(T, P)
```



## Seminario de Algoritmia

```
function i=stringMatching(T,P)
    n = numel(T);
    m = numel(P);
    for i=1: n-m+1
        j=1
        while j<m+1 && P(j)==T(i+j-1)
            j=j+1;
        end
        if j == m+1
            return;
        end
    end
    i=-1;
end
```

### Código de Matlab

```
1 #include <iostream>
2 #include<windows.h>
3
4 using namespace std;
5
6 int EncontrarCad(char* b, char* a){
7     int M = strlen(b);
8     int N = strlen(a);
9     for (int i=0;i<=N-M;i++){
10         int j;
11         for (j=0;j<M;j++){
12             if (a[i+j] != b[j])
13                 break;
14             if (j==M)
15                 return i;
16         }
17         return -1;
18     }
19
20 int main(){
21     char a[] = "Abecedario";
22     char b[] = "dar";
23     int c=EncontrarCad(b, a);
24     if(c!=-1){
25         int M = strlen(b);
26         int N = strlen(a);
27         cout<<"A: ";
28         for(int i=0;i<N;i++){
29             cout<<a[i];
30         }
31         cout<<"\n\n";
32         cout<<"B: ";
33         for(int i=0;i<M;i++){
34             cout<<b[i];
35         }
36         cout<<"\n\n";
37         cout<<"Coincidencia en el indice: "<<c<<endl;
38     }else{
39         cout<<"No hubo coincidencias :(\n";
40     }
41     return 0;
42 }
```

### Código en C++/Python



## Seminario de Algoritmia

### RESULTADOS

Agrega la imagen de la consola con el despliegue de los resultados obtenidos.

```
Command Window  
j =  
    1  
  
j =  
    1  
  
indice =  
    28  
fx >>
```

#### Resultados Matlab

```
A: Abecedario  
B: dar  
Coincidencia en el indice: 5  
Process returned 0 (0x0)   execution time : 0.122 s  
Press any key to continue.  
_
```

#### Resultados C++/Python

### CONCLUSIONES

Escribe tus observaciones y conclusiones.

La actividad fue sencilla de realizar, la idea de ella es ir recorriendo todo el arreglo asta lograr encontrar la primera letra de la palabra que estamos buscando, una vez que la encuentra ir verificando que las siguientes letras de la palabra coincidan con la que estamos buscando, si todas las letras coinciden retornamos la posición de la primer letra que estábamos buscando.