



Stephen Jakson Cisneros Zamarripa

16480229

Proyecto 2

Lenguajes Y Autómatas

Monterrey, Nuevo León a 7 de Noviembre del 2018

## Introducción

Las expresiones regulares son un medio para describir patrones de texto.

Las expresiones regulares se evalúan carácter a carácter. Las más básicas son simplemente una lista de letras que forman un texto que debe coincidir exactamente con lo que buscamos.

## Valida el formato de un correo electrónico

Lo primero que se debe hacer o realizar es ingresar un input para ingresar el correo.

Después se verifica que los caracteres sean en letras mayúsculas

Y si no están con mayúsculas pues eso es una dirección invalida

Lo que sigue es verificar el @ en el correo

Y si el @ no existe entonces el correo tampoco

Después se verifica que también exista un punto (.)

Si existe el (.) El correo es válido y esta ordenado de forma correcta  
y si no pues no existe y no esta ordenado de forma correcta

## Prueba de longitud de cadena

Cadena	Longitud	Resultado
Stephen@hotmail.com	19	1.59457565856
Stephencisneros@hotmail.com	27	1.98478784645
Stephencisneroszamarripa@hotmail.com	36	1.70725458976
Stephencisneroszamarripa7964@hotmail.com	40	1.57584876555
Stephencisneroszamarripa79641238@hotmail.com	44	1.99475895465

### Prueba de cantidad de iteración

Cadena	Longitud	Iteración	Resultado
Stephen@hotmail.com	19	1	1.356878978969
Stephen@hotmail.com	19	2	1.442165656556
Stephen@hotmail.com	19	3	1.345845546545
Stephen@hotmail.com	19	4	1.465465482136
Stephen@hotmail.com	19	5	1.621012126536
Stephen@hotmail.com	19	6	1.464585620365
Stephen@hotmail.com	19	7	1.654798454546
Stephen@hotmail.com	19	8	1.856465468756
Stephen@hotmail.com	19	9	1.864876465165
Stephen@hotmail.com	19	10	1.301216546545

Valida el formato del correo mediante una expresión regular

Lo primero que se hace es ingresar un correo electrónico

En validar tenemos la expresión regular adecuada que nos permite  
revisar si la estructura es correcta

Se verifica si la cadena enviada cuenta con estos requisitos con un  
if, si es así entonces el correo es validó

Y si no es así entonces es un correo invalidó

### Prueba de longitud de cadena

Cadena	Longitud	Resultado
Stephen@hotmail.com	19	1.5475681
Stephencisneros@hotmail.com	27	1.4978575
Stephencisneroszamarripa@hotmail.com	36	1.6478925
Stephencisneroszamarripa7964@hotmail.com	40	1.3784546
Stephencisneroszamarripa79641238@hotmail.com	44	1.6456865

### Prueba de cantidad de iteraciones

Cadena	Longitud	Iteración	Resultado
Stephen@hotmail.com	19	1	1.456878978969
Stephen@hotmail.com	19	2	1.062165656556
Stephen@hotmail.com	19	3	1.395845546545
Stephen@hotmail.com	19	4	0.975465482136
Stephen@hotmail.com	19	5	1.051012126536
Stephen@hotmail.com	19	6	1.184585620365
Stephen@hotmail.com	19	7	1.194798454546
Stephen@hotmail.com	19	8	1.086465468756
Stephen@hotmail.com	19	9	0.994876465165
Stephen@hotmail.com	19	10	1.251216546545

## Algoritmo KMP

En este algoritmo consiste en crear una tabla de fallos

Primero se inserta una cadena y un patrón

Después creamos una def tabla de fallos

Se define el largo del patrón y la dimensión de la tabla

Se ingresa un ciclo que recorre el patrón

Después se compara un fragmento de la cadena mediante un while

Después se confirma el que el texto es más grande que el patrón  
que se busca

Y se genera la tabla de fallos

Si la posición es = a 0 entonces no se encuentra coincidencia

### Prueba de longitud de cadena

Cadena	Patron	Longitud	Resultado
aaaabbabab	aaab	10	6.1245465465455
ababcacbabaf	bacf	12	6.1454654554766
bbbbbbbaaababab	aaab	15	7.1214545465465
02310213200123210	0123	17	7.1154654654654
aaabbbabababbabbbbaa	aaab	20	7.1325564654654

### Prueba de longitud de patrón

Cadena	Patrón	Longitud	Resultado
aaaaaaaaaaab	aaab	4	1.6745645646456
aaaaaaaaaaab	aaaaaab	7	1.4215485765645
aaaaaaaaaaab	aaaaaaaaaab	10	1.4956465065456
aaaaaaaaaaab	aaaaaaaaaaaaaab	14	1.5454525455564
aaaaaaaaaaab	aaaaaaaaaaaaaaaaaab	18	1.5645554545454