



TECNOLÓGICO NACIONAL DE MÉXICO  
TECNOLÓGICO DE NUEVO LEÓN

ISC

Lenguajes y Autómatas 1

Unidad 2

Reporte

Marlene Garza Guzmán

16480135

Guadalupe N.L. octubre 2018

**Buscar cómo funciona el algoritmo KMP, codificarlo y probarlo.**

**Realizar un análisis de complejidad de su algoritmo KMP y explicar cómo se ve afectado por los diferentes factores como el tamaño de la cadena o el tamaño del patrón a buscar**

Al algoritmo KMP es aquel que busca una su cadena simple, teniendo como objetivo el buscar la su cadena dentro de la cadena existente.

El algoritmo utilizado para hacer el programa KAM es de una complejidad de  $nm$  considerando que  $n$  es la longitud del texto y la  $m$  es la longitud del patrón.

Se considera que cuando el programa es realizado este se marca como la cadena de texto más larga y una sub cadena que generalmente es más cota, el programa consiste en que se ira colocando la sub cadena desde el principio verificándonos que no tenga un fallo que quiere decir que mientras los caracteres de las se la cadena y la sub cadena sean iguales el programa seguirá recorriendo, hasta que se encuentra con el fallo o la inconsistencia entre las dos y se detiene, después de eso hace que la sub cadena empiece nuevamente a recoger los espacios pero esta vez con un espacio más a la derecha, y así seguirá hasta que el programa detecte que las dos sean iguales.

**Text : A A B A A C A A D A A B A A B A**

**Pattern : A A B A**

A A B A															A A B A			
A	A	B	A	A	C	A	A	D	A	A	B	A	A	B	A			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
												A A B A						

**Pattern Found at 0, 9 and 12**

**Validar formato de un correo electrónico mediante método propio**

**Se deberá crear una función que realice una validación a una cadena de texto para saber si tiene el formato correcto para ser un correo electrónico.**

Primeramente, se realizaron diferentes investigaciones sobre los métodos que existían de verificación de caracteres después se encontró un método que ayuda a separar el código en dos verificando que en esa separación exista el carácter

indicado, se repite dos veces ya que el email tiene dos caracteres especiales que son el "@" y el "."

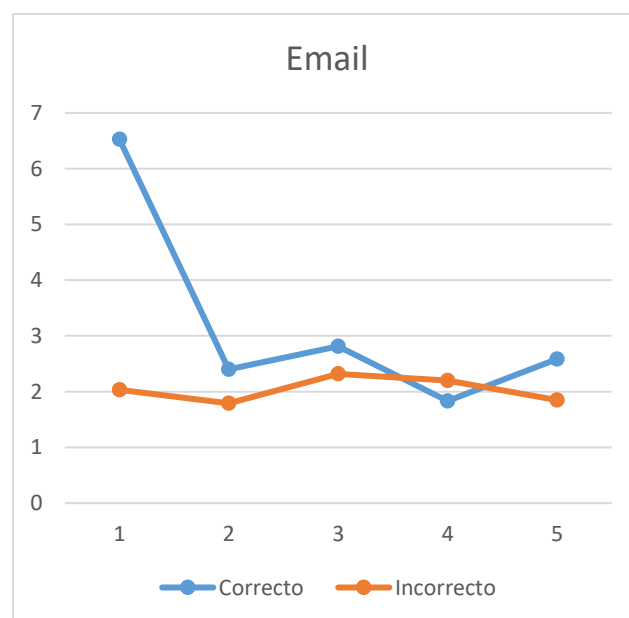
Método utilizado:

Split: es un método utilizado para poder dividir una cadena en partes más pequeñas.

Este método se utilizó para separar la parte que va antes y después del @ y si se llegaba a cumplir la condición ahora separaba la parte que iría después del @ que está separado a través de un '.' Haciendo que verifique nuevamente si se cumple la condición si se llegan a cumplir ambas condiciones se pune como que el e mail es correcto, sin embargo, con alguna de estas restricciones se mandara directamente a un else indicando que el correo no es válido.

N° Prueba ("marlene@hotmail.com")	Tiempo que tardo
1	6.5259997844696045
2	2.3983426094055176
3	2.8121702671051025
4	1.8324003219604492
5	2.5849905014038086

N° Prueba ("marlene.hotmail@com")	Tiempo
1	2.032242774963379
2	1.7904508113861084
3	2.317572593688965
4	2.1962180137634277
5	1.849306344985962



Los datos mostrados en la gráfica anterior muestran los valores que se obtuvieron correr el programa sobre un determinado número de veces tanto incorrecto como

válido. Al inicio con un tiempo más alto conforme se va procesado un número repetido de veces el tiempo se va acortando esto aplica para ambos casos

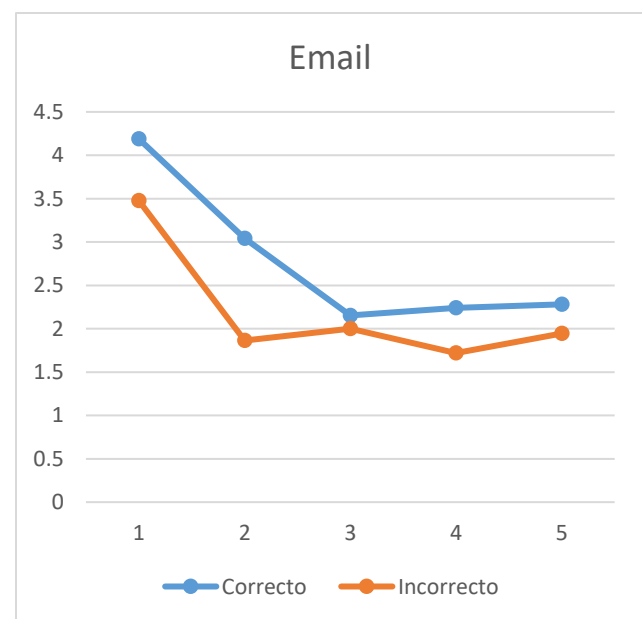
Este programa fue principalmente realizado para saber las diferentes formas de crear una condicional que aplique para un email. La función principal era saber si un programa contaba con las herramientas esenciales para saber si la sintaxis de un correo es correcta o incorrecta, fue realizado dentro de eclipse con el programa Python

### Buscar cual sería la expresión regular necesaria para validar un correo electrónico en Python y crear una función que lo haga.

Primeramente, al llamar el programa se realizó la entrada del email, una vez que se ingresó esta, se comprobó que caracteres son los que debería permitir cada parte del correo, después de que se compruebe que se pasó por todos los requisitos necesarios se indicara que el correo es válido, o de otra manera indicara que no está correcto o es invalido y terminara el programa.

N° Prueba ("marlene@hotmail.com")	Tiempo que tardo
1	4.1883978843688965
2	3.039281129837036
3	2.1529488563537598
4	2.2416675090789795
5	2.2817938327789307

N° Prueba ("marlene.hotmail@com")	Tiempo
1	3.4756927490234375
2	1.862776279449463
3	2.001913070678711
4	1.7191107273101807
5	1.9433197975158691



## **Comparación del programa que realizamos de email y el programa de expresiones regulares, explicando cual fue el mejor.**

Por su complejidad el de expresiones regulares abarca un mayor rango de acierto con una complejidad menor al que realice ya que permite identificar que caracteres son los que se pueden poner en cada área del correo, mientras que el mío tiene un mayor nivel de complejidad y además abarca mucho menos ya que solo verifica que se estén cumpliendo con tener las características principales de un email que son el “@” y “.”.

## **Las pruebas de los programas fueron realizadas en una computadora Lenovo ideapad 320**

- **Procesador:** Intel Core i5 de 7<sup>th</sup> generación.
- **RAM:** 12GB
- **Disco duro:** 1T
- **Sistema Operativo:** Windows 10 home

## **Fuentes**

- <http://cecilia-urbina.blogspot.com/2013/02/string-matching.html?m=1>
- <https://www.pythonforbeginners.com/dictionary/python-split>
- <https://note-code.blogspot.com/2017/01/algoritmo-kmp.html>
- <https://www.youtube.com/watch?v=-OMysIB8FfY>
- <https://www.youtube.com/watch?v=Kglp2Sy5dr0>
- <https://www.youtube.com/watch?v=tIEcDiP9Tql>
- <https://www.lawebdelprogramador.com/codigo/Python/2040-Validar-cuenta-de-correo.html>
- <https://www.youtube.com/watch?v=FYuIWn85RNk>