TECNOLÓGICO
NACIONAL DE MÉXICO

Proyecto 1. Lenguajes y Autómatas

UNIDAD 1

Nombre:

César Alejandro Medina Arredondo

Maestro:

Juan Pablo Rosas Baldazo

Un alfabeto es un conjunto de caracteres con los cuales se pueden formar palabras a las que se le llaman lenguajes. Se pueden crear una infinidad de palabras combinadas para cada alfabeto.

¿Que se hizo?

En este programa codificado en python del cual permite leer un alfabeto y genera una lista e palabras

¿En donde se hizo?

Fue realizado en lenguaje Python utilizando la maquina con las siguientes características:

AMD A8-6410 Radeon R5 Graphics 2.00 GHz

Memoria RAM 6.00 GB

Windows 10 Home 64 bits

1- Definir un numero n de palabras en un alfabeto

Primero que nada se declara alfabeto con un input donde esta pueda realizar una acción antes de ingresar el alfabeto que se estará utilizando en el programa, se noto un error en que el alfabeto debe estar pegado, sin espacios ni comillas ya que se tomara como un elemento al querer hacer combinaciones. Se declara luego las variables que se estaría utilizando en el código más adelante, indicando su longitud o dejarlos como una cadena vacía.

Con un ciclo a través de un while se estará indicando que se estará checando la cantidad de letras sea menor a la indicada, por el for estará avanzando en cada letra del alfabeto, después de leer cada letra que lo conforman se

elegirá una letra al azar y cada vez que vuelva a dar una vuelta se coloca una de las letras del alfabeto.

Con la función append guarda la cadena que se creo, y se elimina la cadena formada posteriormente, esto es para cuando se estén generando nuevas cadenas no tenga elementos sobre ella y se evita que las combinaciones sean muy largas ya que se combinan las cadenas creadas anteriormente en las nuevas si no están vacias.

```
improt random

def clase():
    alfabeto = input (“ingrese el alfabeto”)
    n = 4
    m = 0
    letra = “ ”
    lista= [ ]


while       m<=n:
for x in range (0, len(alfabeto)):
letras = random.randint(1,4)
letra = letra.join(random.choice(alfabeto)) for_in range(letras))
lista.append(letra)
letra = “ “
m = m+1
```

```
    print(lista)


from time import time

start = time()

clase()

end=time()-start

print(end)
```

En la tabla se mostro utilizando e mismo abecedario en distintas ocasiones para la generación de una palabra o combinaciones de estas mismas, se noto que al realizarlo tenia un mayor numero de tiempo que se tardaba realizando la operación después disminuyo

Numero de prueba (“sgadf32623g”)	**Tiempo**
1	**53.279603**
2	**1.283284**
3	**1.3042066**
4	**1.554936**

2- Dado un conjunto de cadenas de un lenguaje, definir cual es el alfabeto e ese lenguaje

En este siguiente programa se pide que se realice una identificación del alfabeto partiendo de una cadena de texto

que se ingresara con anterioridad. Al igual que el primer programa de ingresar lo que es el input donde el usuario pordra ingresar una cadena de texto, se irán declarando las variables que se irán utilizando. Se declara el “por i in” donde se identifica cada valor de las listas.

Se toma el primer valor y si el primer elemento de la lista no esta en la nueva cadena se guardara, se irán eliminando los caracteres que estén repetidos dejando así que solo se imprima una vez cada elemento.

Tanto como este como el programa anterior la cadena o alfabeto se tiene que escribir todo junto para que no se tomen ningún elemento vacio.

```
def clase 2 ():
    lista = input (“escribe la cadena de texto”)
    cadena = “ ”
    lista2 = [ ]
    for i in (lista):
        m = i
        for x in (m):
        if x not in lista2:
        lista2.append (x)
print (lista2)
```

```
from time import time

start = time()

clase2 ()

end = time()-start

print( end)
```

Como se vio en el primer programa, la vez que se ingreso la cadena de texto se tardo mas que las que fueron posteriores exactamente con la misma cadena, mas sin embargo no fue tan grande el rango en que tardo en ejecutarse como en el primer programa.

N prueba Programa 2	**Tiempo**
1	**13.11482143**
2	**1.6568455696**
3	**1.4249203205**
4	**3.4965324401**