

INSTITUTO TECNOLOGICO DE NUEVO LEON



Carrera:

Ing. Sistemas Computacionales

MATERIA:

Lenguajes autómatas 1

Trabajo:

Reporte de Proyecto 1

Profesor:

Ing. Juan Pablo Rosas Baldazo

Alumno:

Eduardo Martínez Martínez

15690231

Primeramente, describiré el concepto de un alfabeto el cual es un conjunto de caracteres con los cuales se pueden formar palabras, las cuales llamamos lenguaje.

Entre todo esto, es posible crear una cantidad enorme de combinaciones (palabras) para cada alfabeto. Así cómo es posible también deducir un alfabeto a partir de un conjunto de palabras.

¿Qué se hizo?

En este documento se detallará el desarrollo de un programa, codificado en Python, que permite leer un alfabeto y generar una lista de palabras (lenguaje) de tamaño variable.

Su segunda función es, determinar un alfabeto a partir de un lenguaje, leyendo el alfabeto ingresado y formando palabras.

Ambas funciones mostraran sus resultados impresos en pantalla

¿En dónde se hizo?

Se realizó en Python y ejecutado en el mismo

Así como la utilización de máquinas las cuales mostrare a continuación;

El equipo en que se harán las pruebas es un laptop HP Pavilion x360 de los siguientes programas;

- ✓ GENERAR UN LENGUAJE A PARTIR DE UN ALFABETO
- ✓ GENERAR UN ALFABETO A PARTIR DE UN LENGUAJE

tiene estas características

- Procesador i5
- Graficos Rendon ddr 5 Graphics
- Windows 10 Home de 32 bits
- 6 GB de memoria RAM

¿Cómo se realizó?

Se les mostrara paso a paso de acuerdo a cada programa ya que tiene similitudes entre sí;

Porque el primer proceso que se realizará en el primer programa, será el segundo que se realizará en el segundo programa

Primer programa:

Descripción

Generar un lenguaje a partir de un alfabeto

Comenzamos creando un generador de cadenas aleatorias, para esto cree una lista con las vocales y varios ciclos para que me diera todas las combinaciones posibles.

GeneradorLenguaje (n,alfabeto)

```
lenguaje = []
```

```
palabra=""
```

```
palabra=palabra.join(random.choice(alfabeto) for _ in range(longitud))
```

El siguiente paso consistió en agregar la palabra aleatoria a una lista que contendrá el lenguaje

```
if palabra not in lenguaje
```

```
    lenguaje.append(palabra)
```

```
    control = control +1
```

```
    palabra = ""
```

```
else
```

```
    palabra = ""
```

Se necesito crear una condición de ciclo “while”, y un contador para contar las iteraciones, de forma que sean las mismas que se pidieron.

```
while control != n
```

```
    longitud=random.randint(1, 6)
```

```
    palabra=palabra.join(random.choice(alfabeto) for _ in range(longitud))
```

```
    if palabra not in lenguaje:
```

```
        lenguaje.append(palabra)
```

```
        control = control +1
```

```
        palabra = ""
```

```
    else:
```

```
        palabra = ""
```

Con función “random.randint(n,m)” te permite devolver un numero al azar de entre un rango dado (1,6).

```
longitud=random.randint(1, 6)
```

por ultimo tuve que crear una nueva función que serviría para ejecutar ambas funciones requeridas para este proyecto, así como la captura de los datos necesarios, los cuales son enviados hacia sus respectivas funciones.

```
ingresarDatos()
```

```
prueba = input(" Ingresa (a) si deseas obtener un alfabeto a partir de un lenguaje  
\n Ingresa (l) si deseas obtener un lenguaje a partir de un alfabeto")
```

```
if prueba == "l":
```

```
    alfabeto = input("Ingresa las letras del alfabeto a utilizar.")
```

```
    n = int(input("Ingresa la cantidad de palabras que generara el lenguaje."))
```

```
    print(GeneradorLenguaje(n,alfabeto))
```

Segundo programa

Descripción

Generar un alfabeto a partir de un lenguaje

Si se da cuenta aquí es lo viceversa del programa anterior ya que primera la generación de un alfabeto a partir de un lenguaje

Lo primero se inicia crear la función “Generador Alfabeto”, que recibe una lista con las palabras que conforman el lenguaje.

GenerardorAlfabeto(lenguaje):

```
alfabeto = []
```

El siguiente paso consistió en juntar todas las palabras en una sola cadena para facilitar su procesamiento, usando la función {`"".join(lista)`}, que se encontró al hacer un programa previo, esta junta todos los elementos en una lista que están separados por comas.

```
frase = "".join(lenguaje)
```

Después en un ciclo “for” recorre la frase y dentro esta la condición, “Si el carácter actual no se encuentra en el alfabeto, esta se añade”, para terminar, se regresa el alfabeto.

```
for i in frase:
```

```
    if i not in alfabeto:
```

```
        alfabeto.append(i)
```

hast aquí el resultado presentado era en desorden, pero con la implementación de la función “sort” al alfabeto para que se acomodaran en orden ascendiente.

`alfabeto.sort()`

¿Para qué se realizó?

Para conocer la importancia de investigar métodos en un lenguaje de programación lo cual se tuvo que hacer para obtener las funciones “.join(lista)” y “.split()”, así como a partir de un programa pequeño y algo tedioso se puede usar para algo complejo.

También entender que un programa no es sencillo porque siempre se tendrá que buscar que el programa sea más eficiente y eficaz lo cual en su trayectoria será más compleja

Pruebas de longitud de cadena para la aplicación de generar un alfabeto

Las pruebas de longitud.

Cadena	longitud	Resultado
a e i o u	5	2.6942105
1 2 3 4 5	5	2.26825
12345 abcde	10	2.4146552
12cd5 ab34e aed 531 12345 abcde	26	2.4650972
123456789 abcdefg hijklmn opqrst uvwxyz	35	2.8030076

Pruebas de longitud de cadena para la aplicación generar un lenguaje

Primero probaremos cómo reacciona el sistema cuando se le ingresan cada vez más datos.

Cadena	# palabras	longitud	Resultado
12345	20	5	3.1948299407958984
abcdefghijkl	20	10	3.3629891872406006
0123456789abcde	20	15	3.593363046646118
abcdefghijkl!#\$%&/()=	20	20	3.6419031620025635
0123456789abcdefgh ijklmno	20	25	3.806324005126953