

Proyecto Unidad 1

12-9-2018



Nombre: Angel Hernandez

Materia: Lenguajes y autómatas 1

Maestro: Juan Pablo Rosas Baldazo

N° de Control: 16480017

Pseudocódigo

Programa: Proyecto1

Entorno: Abecedario

Algoritmo

- ingresar valor primario
- leer valor longitud
- crear lista
- crear cadena
- agregar ciclos con sus respectivas condiciones
 - hacer la combinación de las letras
 - revisar valor existente
 - añadir a la lista

imprimir

medir tiempo en segundos

Fin del programa

¿Qué se hizo?

En el primer paso declaramos un arreglo que en este caso nosotros lo llamamos “abecedario”, en la cual irán las letras que nosotros queramos, pero cada carácter debe ir dentro de comillas simples o comillas dobles y separadas entre sí por comas para que se puedan tomar como valores independientes y no como un solo valor.

Después declaramos dos variables de tipo entero que en este caso sería “a” y “b”, que a la variable “a” le asignamos el valor de 0 y a la variable “b” le asignamos el input para poder ingresar desde el teclado el valor que se desea que tenga b.

En el siguiente paso declaramos una lista vacía en la cual se almacenarán todas las combinaciones que se vayan realizando cada vez que el programa de una vuelta, después de eso, se declara un arreglo que en este caso lo llamamos “letra” el cual estará vacío para que cada vez que el programa de una vuelta, el programa no vuelva a repetir esa misma combinación, es decir, que el nuevo valor sea distinto al que ya se ha agregado a la lista.

Declaramos un ciclo while con la condición de que “a” sea menor que “b” con el fin de que se lean puros valores positivos desde el teclado.

Dentro de ese ciclo while ira otro ciclo, pero en este caso utilizamos el ciclo “For” con la variable “i” que tendrá un rango de (a, b) que en este caso “a” lo declaramos con un valor de “0” y “b” sería el número que nosotros ingresemos desde el teclado.

Luego, le ingresaremos la combinación al arreglo “letra” de la siguiente manera:

Letra = Letra+random.choice(abecedario)

La cual nos permitirá hacer la combinación de dos o más letras, pero siempre tomando los valores que están en el arreglo “abecedario”.

Luego se le pone una condición con un "if" y revisara que la combinación que se le esta asignando al arreglo Letra no esté ya en la lista y lo que nos ayuda en este paso es el "not in" que es el que hace que revise si la combinación que se quiere arreglar a la lista no exista, si no existe, lo agregara a la lista y lo que hará posible que se agregue es lo siguiente:

```
Lista.append(Letra)
```

Que el append es el que nos permite agregar a la lista los valores que están en "Letra" y para que no se inicie nuevamente con el valor de "0", lo que hicimos es incrementar la "a" en 1 cada vez que el programa de una nueva vuelta de la siguiente manera:

```
a = a+1
```

luego imprimimos la lista con un print(lista)

después haremos que nos calcule el tiempo de ejecución del programa, para lo cual se necesita el siguiente código:

```
from time import time  
start = time()
```

```
Proyecto1()
```

```
end=time()-start  
print(end)
```

¿En dónde se realizó?

El proyecto fue elaborado en un ordenador de la marca "DELL" con un procesador Intel Core i5 7th Gen con una memoria de 8GB de RAM y nuestras experiencias fueron las siguientes:

Al inicio del proyecto, hubo dificultades al ejecutar el programa debido a que es un lenguaje, con el cual no habíamos interactuado y los errores predominaban en nuestro código.

Después de investigar un poco fuimos entendiendo para que era cada cosa y nos familiarizamos con esto, luego introducíamos valores que saturaban la memoria del ordenador por lo que la maquina no trabajaba a su ritmo e incluso se trababa por la sobresaturación de la memoria.

Se tuvieron que tomar decisiones no favorables para el ordenador, debido a que era demasiada la saturación de memoria que dejaba de responder y se tenía que apagar directamente del botón de encendido.

Devuelta, encendiendo el ordenador, poníamos los rangos no tan mayores para que no volviera a pasar lo mismo y de esta manera comenzó a funcionar un poco mejor.

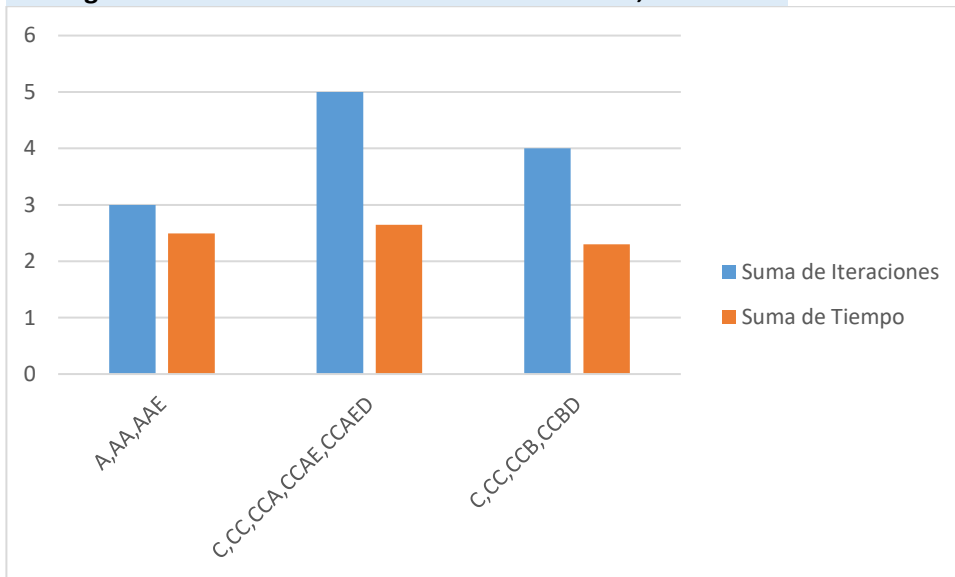
Existían muchas dudas con respecto al código debido a

Luego de entender más el código, detectamos cuales eran los errores por los que el ordenador tenia ciertas fallas, se corrigieron y el programa se ejecutó a la perfección y siempre cumpliendo con lo deseado.

¿Cómo se realizó?

Se realizaron pruebas para comprobar su funcionalidad, en la siguiente tabla se muestra el resultado y el tiempo que tardo en ejecutar:

Etiquetas de fila	Suma de Iteraciones	Suma de Tiempo
A,AA,AAE	3	2,494728
C,CC,CCA,CCAE,CCAED	5	2,645631075
C,CC,CCB,CCBD	4	2,30194521
Total general	12	7,442304284



¿Para qué se realizó?

Se realizó con la finalidad de que cada vez que el programa diera una vuelta, revisara si la combinación de caracteres ya está almacenada en la lista y también se realizó con el fin de entender un poco más los componentes de Python para poderse desarrollar y poder hacer pseudocodigos con mayor facilidad.

Pseudocodigo Proyecto 2

Programa: Proyecto2

Entorno: Ingresar palabras

Algoritmo

 crear una lista

 crear un ciclo for que tome los valores del entorno

 declarar variable e igualarla a la del ciclo

 crear un nuevo ciclo para revisar la existencia del caracter

 añadir caracter a la lista

 imprimir la lista

calcular el tiempo de cuanto tardea en ejecutarse por completo medido en segundos

Fin del programa

¿Qué se hizo?

Se declaró un arreglo en el cual se almacenará el alfabeto que nosotros ingresemos, para lograr que el alfabeto se lea desde el teclado, se le agrega el input que es el que hace dicha función y dentro de él se pide que ingresemos el alfabeto.

Después declaramos una lista en la cual se encargará de guardar las letras del alfabeto que nosotros ya hayamos ingresado, después declaramos un ciclo for el cual tendrá la variable "j" y tomará los valores de "b".

Se añadió una condición la cual tiene la función de revisar si los caracteres que se le están asignando a "j" ya existen en la lista que ya se ha creado, en caso de no existir, se añadirá a la lista, esto se lleva a cabo con un. append.

Se imprimió la lista para mostrar los elementos que se han guardado en ella y por ultimo se calcula el tiempo de ejecución en el programa medido en segundos.

¿Para qué se realizó?

El proyecto se realizó con el fin de imprimir combinaciones de letras sin perder su orden y que no se repitan, así mismo, darle al programa palabras y que las separe de forma que nos muestre el alfabeto con el que se conformaron las palabras sin que se repitan las letras.

Etiquetas de fila	Suma de Iteraciones	Suma de Tiempo
B, BF, BFA, BFAF, BFAFF	5	1,596203804
D, DC, DCC, DCCF	4	1,594943762
F, FE, FEC	3	3,507759571
Total general	12	6,698907137

