Introducción a la Programación con Python: Generador de Frases

Ever Favio Argollo Ticona
Universidad Católica Boliviana San Pablo
La Paz, Bolivia
e-mail: everfavioat@gmail.com

***Resumen*— Python es un lenguaje moderno, de alto nivel y con una flexibilidad que resulta muy amigable para quienes inicien con la programación. Este cuenta con las características necesarias para desarrollar programas bastante útiles. Un generador de frases es un programa que nos facilita textos con cierto sentido para poder usarlos en una etapa de desarrollo, Se hace uso del lenguaje de programación mencionado para implementar un generador de frases.**

***Índice de términos*— Python, ciencia de datos, algoritmo, random, frases**

I. INTRODUCCIÓN

En ocasiones necesitamos generar textos de prueba, por ejemplo, para rellenar un sitio web antes de que tengamos disponible contenido real, o proveer contenido de prueba cuando desarrollamos un escritor de reportes, para lo cual es necesario contar con un generador aleatorio de frases.

II. DESCRIPCIÓN DEL PROBLEMA

Crear unas listas de palabras agrupadas, por ejemplo, artículos (“el”, “la”, “los”... etc), sujetos(“gato”, “persona”, … etc), verbos(“saltar”, “comer”, … etc), y adverbios, luego realizar una serie de iteraciones y en cada una de ellas usar el módulo random.choice() para seleccionar las palabras agrupadas anteriormente descritas e imprimir las sentencias resultantes.

III. RESOLUCIÓN

En el mismo enunciado se establece el uso del módulo random.choice(), que nos proporciona Python, para la conjugación de la frase, el problema principal radica en los tiempos verbales del español, por lo que se declararán las palabras iniciales en inglés. se formará la frase e inmediatamente se hará uso de la api de google translate para mostrar los resultados en español.

A. Definiendo los datos de entrada y variables iniciales

El enunciado establece que se debe realizar una serie de iteraciones no definida por lo que este será nuestro parámetro de entrada, un número natural no mayor a 9 para no generar resultados repetitivos, aunque esto dependerá de la cantidad de información que se tenga en la lista de palabras preestablecidas.

0 < i < 10;

B. Definiendo los procesos principales

El proceso principal es, básicamente, una serie de concatenaciones sujetas al valor de la iteración definida por el usuario, por lo que podemos establecer:

```
for iterations=0, iterations < i, iterations ++:
  phrase =  random(articulo) +
            random(sujeto) +
            random(verbo) +
            random(adverbio);
end;
return phrase;
```

C. Resumen general del pseudocódigo

```
algorithm: phraseGenerator
INPUT: i , 0 < i < 10;
OUTPUT: (phrase) * i, where phrase:
phrase = random(artículo) +
         random(sujeto) +
         random(verbo) +
         random(adverbio);
```

D. Implementación del Código con Python

A continuación se describe la implementación principal del pseudocódigo descrito en lenguaje python, se han declarado librerías auxiliares para la mejor comprensión del código en el método principal.

```
inp = input('Introduce la cantidad de frases que deseas generar:
')
# Iterando las veces definidas en inp
try:
  for n in range(int(inp)) :
    if random.randint(0,1) == 1 :
      phrase = (f'{n + 1}
              {random.choice(articulos)}
              {random.choice(sujeto)}
              {random.choice(verbos)}
              {random.choice(infinitivos)}')
      result = (translator.translate(
             (phrase), src='en', dest='es'))
      print(result.text)
    else :
      phrase = (f'{n + 1}
              {random.choice(sujetos)}
              {random.choice(verbos)}
              {random.choice(infinitivos)}')
      result = (translator.translate(
             (phrase), src='en', dest='es'))
```

```
        print(result.text)
    pass
except ValueError as  error:
    print('Debes introducir un número válido!..')
```

IV.CONCLUSIONES

Haciendo uso del lenguaje Python y sus librerías integradas encontramos comodidad al momento de implementar pseudocódigo, el uso del módulo random puede sernos de gran utilidad a la hora de dejar al azar la selección de determinados elementos. La implementación completa del ejercicio puede ser revisada y descargada desde el siguiente repositorio: https://github.com/everfavio/datascience/blob/master/ejercicios/phrase-generator.py

REFERENCIAS

[1] M.. Summerfield, *Programming in Python 3 A complete introduction to the Python language, 2nd ed.* Boston, MA.: Pearson Education, pp. 2-3, 2010.

[2] Python.org "Python Documentation"[online] Available: https://docs.python.org/3/tutorial/inputoutput.html. 2020.

[3] Python.org "Python Documentation" [online] Available: https://docs.python.org/3/library/random.html 2020.