

Introducción a la Programación con Python: El Ahorcado

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. El Ahorcado es un juego de adivinanzas donde el objetivo principal es descubrir cuál es la palabra secreta sugiriendo una letra a la vez. Se hace uso del lenguaje de programación mencionado para implementar el juego del Ahorcado.

Índice de términos— Python, ciencia de datos, algoritmo, ahorcado

I. INTRODUCCIÓN

En ocasiones necesitamos que la computadora contraste y valide un set de datos insertados con información persistente, filtrando los datos útiles y descartando los que no aporten nada a la solución considerados también “datos basura”. Siendo Python un lenguaje de programación sencillo, legible y elegante que atiende a un conjunto de reglas que hacen muy cortas su curva de aprendizaje, se considera el adecuado para el procesamiento de datos.

II. DESCRIPCIÓN DEL PROBLEMA.

El juego del ahorcado consiste en tomar una palabra al azar y darle al usuario la posibilidad de que adivine letras hasta completar la palabra, tiene 7 oportunidades para adivinar letras. Por cada letra incorrecta, se resta un intento. Cuando se queda sin intento, pierde. Si adivina todas las letras, gana.

III. RESOLUCIÓN

A. Definiendo los datos de entrada y variables iniciales

El juego se basa en encontrar una palabra escogida al inicio, por lo que es necesario tener un stock de donde poder escoger aleatoriamente una, descartando todas aquellas que tengan una longitud menor a 6 letras para agregarle un cierto grado de complejidad.:

```
palabra.length >= 6
palabra ∈ palabras
palabras := [0:2999]
letrasCorrectas = []
```

El usuario tiene 7 oportunidades para descubrir la palabra, cada vez que se equivoca se resta un intento.

```
i = 0; i < 7
```

B. Definiendo los procesos principales

El primer proceso principal, previo a la inserción de las letras por parte del usuario, es la selección de una palabra dentro de un diccionario preestablecido.

```
palabra = random(0, len(diccionario) - 1).
```

El segundo proceso principal, es el de la inserción y validación de las letras del usuario, restando una oportunidad cada vez que la letra no pertenezca a la palabra seleccionada y agregando la letra a la posición correspondiente de la palabra dentro de un array en caso de que se haya introducido una letra válida.

```
l := user input
if l not in palabra then
    i := i + 1;
else
    letrasCorrectas.add(l)
fi
```

Finalmente, si el usuario ha utilizado todos sus intentos y no ha sido capaz de adivinar la palabra, se retorna el mensaje asignado para la derrota, en un caso contrario, se le declara vencedor retornando el mensaje de victoria.

```
if i := 0 then
    mensaje := 'derrota!';
else
    if palabra in letrasCorrectas then
        mensaje := 'victoria!';
fi
```

C. Resumen general del pseudocódigo

```
algorithm: hanged
PRESET: palabra, intentos, letrasCorrectas
INPUT: letra
OUTPUT: mensaje, where mensaje:
if intentos := 0 then:
    mensaje := 'derrota!';
else
    if palabra in letrasCorrectas then
        mensaje := 'victoria!';
fi
fi
end;
```

D. Implementación del Código con Python

A continuación se describe la implementación principal del pseudocódigo descrito con python, se han declarado métodos auxiliares para la mejor comprensión del código en

el método principal.

```
def hanged(palabras):
    letrasSeleccionadas = []
    palabraSecreta = establecerPalabraAleatoria(palabras.lista)
    palabraOfuscada = np.full(len(palabraSecreta), "_")
    intentos = 0
    mensaje = 'Jugemos'
    while intentos < 7:
        imprimirPantalla("".join(palabraOfuscada), intentos, mensaje,
palabraSecreta)
        letra = elegirLetra(letrasSeleccionadas)
        if letra not in palabraSecreta:
            intentos += 1
            mensaje = 'Intenta otra vez'
        else:
            for _ in range(len(palabraSecreta)):
                if palabraSecreta[_] == letra:
                    palabraOfuscada[_] = letra
            if ''.join(palabraOfuscada) == palabraSecreta:
                return victoria(palabraSecreta)
            mensaje = 'Bien, esa letra te ayudará'
    return derrota(palabraSecreta)
```

IV.CONCLUSIONES

Haciendo uso del lenguaje Python, encontramos que la simplificación establecida en el pseudocódigo puede ser fácilmente implementada en un script python, la estructura del código es bastante natural y promueve una forma de escribir que facilita su lectura, la implementación completa del ejercicio puede ser revisada y descargada desde el siguiente repositorio:

<https://github.com/everfavo/datascience/blob/master/ejercicios/hanged.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.