

Programa en Python

Proyecto final

Adivina el Número

Trabajo en equipo

Members:

Jared Emmanuel Aguirre Vera, Luis

Fernando Sotomayor Garcia, Karla Cecilia

Valeria Rios Lara, Esteban Villa Rosas

December 04, 2022.

Adivina el Número

Proyecto en equipo

December 04, 2022

1. Introduction

Al encontrar diferentes funciones en Python, en este caso nos llamó la atención el poder realizar un programa para un juego divertido, y que hasta cierto punto requiera de un poco de lógica, más adelante se darán cuenta del por qué mencionamos esto.

No obstante, creo que a todos en algún momento nos ha gustado algún juego, pero no siempre sabemos como es que funciona más a detalle, pero durante este proyecto, no solo msotraremos el programa, sino que todo el proceso y lo que se llevo a cabo para ralizarlo de una manera exitosa.

2. Hipótesis

Nuestro objetivo es realizar un programa cuyo propósito sea que una vez que hayamos dado un número cualquiera, tomando en cuenta que este número debe ser un natural mayor a 1, entonces el usuario pueda adivinar el número que la computadora elija al azar entre los parametros dados. Donde se le dan indicaciones, de si el número es mayor o menor al número que eligió el usuario y así hasta que pueda llegar al número elejido por la computadora.

3. Metodología

En primer lugar definimos una función la cual va a tener toda la lógica del programa del juego, y se llamará ¡adivina_elúmero!, dicha función va a tomar un parámetro el cual se denominara x, este parámetro va a representar el límite superior del intervalo válido de valores. Lo anterior podemos ilustrarlo de la siguiente manera:

Figura 1

```
def adivina_el_numero(x):
```

Por otra parte, vamos a utilizar “cont” por lo que lo decaramos. Esto nos permitirá contar los intentos fallidos que utilizó el usuario para descubrir el número.

Figura 2

```
cont = 0
```

Ahora bien, cuando comienza el juego, en principio necesitamos mostrar los mensajes de bienvenida al usuario y solicitarle su nombre, solamente por orden del juego o por que el usuario tenga una mejor experiencia durante su participación. Así que haremos esto con la función print, la cual nos permite mostrar un mensaje en la consola al ejecutar el programa. Entonces mostramos el mensaje ¡Bienvenido al juego! y preguntamos el nombre.

Figura 3

```
print("¡Bienvenido al juego!")  
print("¿Cual es tu nombre?: ")
```

Usaremos “username” para poder registrar el nombre del jugador.

Figura 4

```
username = input()
```

Luego tenemos que mostrar la meta del juego al usuario, mostrando el mensaje ¡el objetivo es adivinar el número generado!

Figura 5

```
print("el objetivo es adivinar el número generado")
```

Comenzamos a implementar la funcionalidad del juego generando el número aleatorio, utilizando una sentencia import, la cual nos permite importar un módulo, que es como un archivo que contiene funciones y elementos útiles que podemos usar en nuestro programa.

Figura 6

```
[ ] import random
```

En este caso vamos a importar el módulo “random”, que significa aleatorio, ya que este nos permite trabajar con procesos aleatorios y pues para

nuestro programa necesitamos generar un número aleatorio y que sea entero, para lo que necesitamos también a la función “randint”, la cual toma dos parametros a y b pues nos da un intervalo de (a,b), y retorna un entero aleatorio N tal que $a \leq N \leq b$, entonces estos parámetros van a determinar el rango de valores posibles que puede tomar ese valor aleatorio. En este caso definiremos el rango de valores entre 1 y x, esto sería de la siguiente manera.

Figura 7

```
número_aleatorio = random.randint(1, x)
```

Así, una vez teniendo nuestro número aleatorio, tenemos que preguntarle al usuario que número va a predecir, que es el número aleatorio de la computadora, es por eso que vamos a crear una variable que se va a llamar “predicción”, de la cual su valor inicial va a ser 0 para que no haya ninguna posibilidad de que coincida inicialmente con el número aleatorio, por ejemplo si nosotros comenzáramos nuestra predicción con el número uno, puede ser que el número aleatorio también sea uno y la predicción inicial sea uno y entonces no habría juego, el usuario no tendría ni una sola ronda para adivinar el valor.

Figura 8

```
predicción = 0
```

Luego de que definieramos nuestra variable predicción, tenemos que crear una parte repetitiva del proceso, y para ello vamos a utilizar un ciclo “while” ya que necesitamos repetir una secuencia de instrucciones, un número no específico de veces, pues nosotros no sabemos cuantas veces tendremos que pedirle al usuario que de una predicción del número, porque el usuario puede adivinar el número a la primera vez o quizá se necesitan 500 repeticiones para llegar al número aleatorio, dependiendo también del tamaño del intervalo. En este ciclo, le mostramos al usuario el mensaje de que “Adivina el número entre (1, x)”, lo cual quedaría de la siguiente manera.

Figura 9

```
while predicción != número_aleatorio: #el ciclo  
    print(f"Adivina el numero entre (1, {x}):")
```

La condición para que el proceso siga es que mientras la predicción no sea igual que el número aleatorio, pues si es igual entonces ya el usuario

predijo el número aleatorio correctamente y el juego termina. Por otro lado si la predicción es distinta del número aleatorio entonces necesitamos realizar un proceso específico.

1. Primero tenemos que pedirle al usuario que adivine un número entre 1 y x. Para obtener un valor del usuario necesitamos usar la función `input`, que nos permite interactuar con el usuario, es decir, mostrar un mensaje en el que solicitas algo y recibir algo a cambio. Cabe mencionar que `int` nos permite transformar el valor ingresado por el usuario a un entero, o mejor dicho, ese valor se asigna como un entero a la variable, permitiendo trabajar bien al programa. Esto nos quedaría de la siguiente manera:

Figura 10

```
predicción = int(input())
```

2. Una vez que tenemos la predicción, si la predicción es correcta el ciclo se va a detener, por que la condición va a ser falsa. Entonces el jugador gana el juego.
3. Ahora bien, si la predicción es menor que el número aleatorio, entonces mostramos el mensaje ¡fallaste, intenta con un número más grande!. Así mismo, se ira sumando un intento fallido por parte del usuario, con ayuda de “`cont`”, como lo mencionamos anteriormente.

Figura 11

```
if predicción < número_aleatorio: #establecemos condici  
    print("fallaste, intenta con un número más grande")  
    cont+= 1 #se suma 1 cada que alguien coloque un número
```

4. En cambio, si la predicción es mayor que el número aleatorio, entonces mostramos el mensaje ¡fallaste, intenta con un número más pequeño!. Y de igual manera, se sumará un intento fallido por parte del usuario, con ayuda de “`cont`”.

Figura 12

```
elif predicción >número_aleatorio:  
    print("fallaste, intenta con un número más pequeño")  
    cont+=1 #se suma 1 cada que alguien coloque un número
```

5. Finalmente, como se dijo en (2) si la predicción es igual que el número aleatorio, el ciclo se va a detener inmediatamente, cuando esto pase, mostraremos el siguiente mensaje “¡Muy bien! nombre del usuario haz hallado el numero número aleatorio. Felicidades”. Al igual que se mostrará el mensaje de intentos fallidos que se tuvieron.

Figura 13

```
print(f"¡Muy bien! {username} haz hallado el numero {número_aleatorio}. Felicidades")

print("Número de intentos fallidos", cont)
```

Nuestro programa completo quedaría de la siguiente manera.

Figura 14

```
✓ [8] import random
0s

[ ]

✓ [9]
0s
def adivina_el_numero(x):

    cont = 0 #Esto nos servirá para contar los intentos fallidos

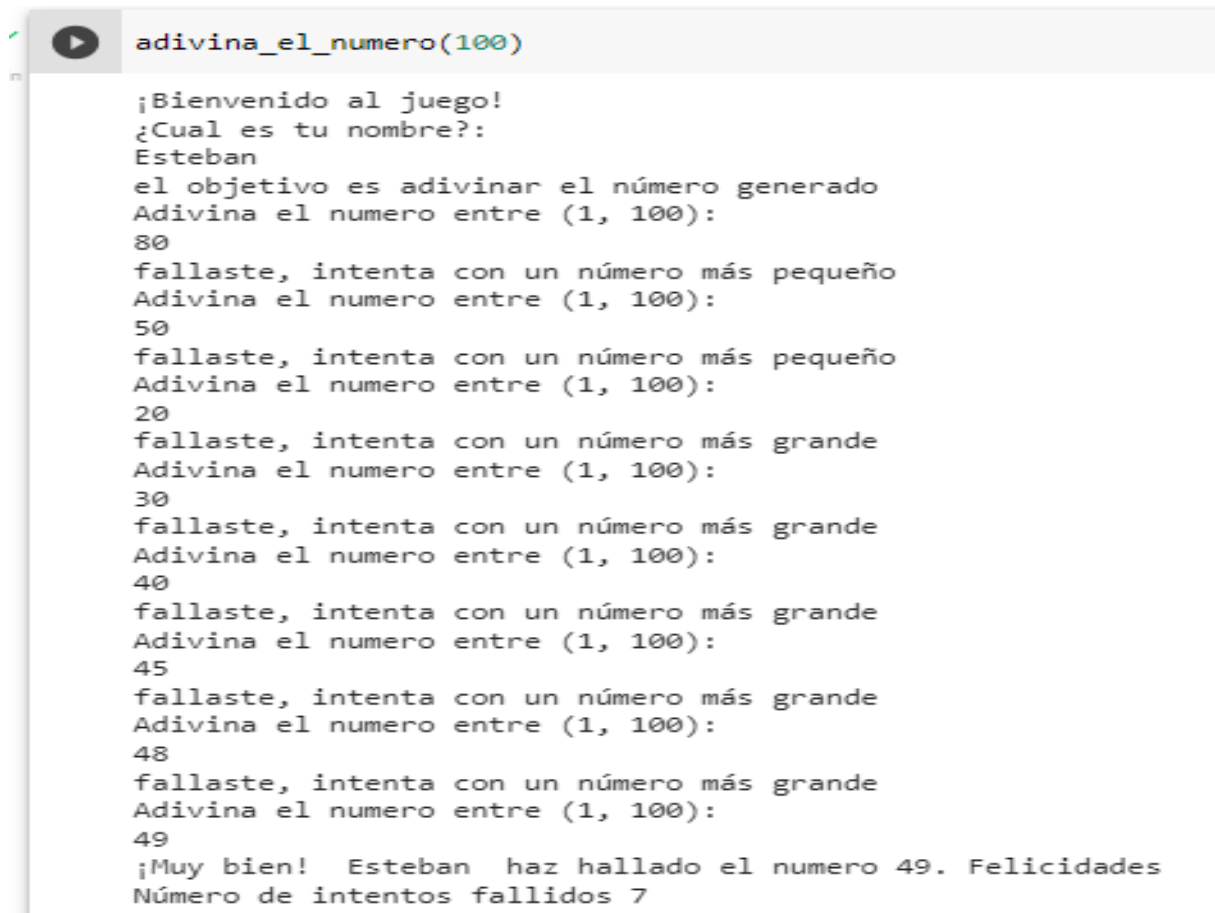
    print("¡Bienvenido al juego!")
    print("¿Cual es tu nombre?: ")
    username = input()
    print("el objetivo es adivinar el número generado")

    número_aleatorio = random.randint(1, x) #definimos un intervalo de número enteros

    predicción = 0 #Empezamos con un valor no definido en el intervalo para que no haya la posibilidad de que el juego acabe sin ninguna interacción
    while predicción != número_aleatorio: #el ciclo while lo usamos porque no sabemos que numero de veces vamos a requerir el mismo ciclo
        print(f"Adivina el numero entre (1, {x}):")
        predicción = int(input())
        if predicción < número_aleatorio: #establecemos condiciones al ciclo
            print("fallaste, intenta con un número más grande")
            cont+= 1 #se suma 1 cada que alguien coloque un número incorrecto
        elif predicción > número_aleatorio:
            print("fallaste, intenta con un número más pequeño")
            cont+=1 #se suma 1 cada que alguien coloque un número incorrecto
    print(f"¡Muy bien! {username} haz hallado el numero {número_aleatorio}. Felicidades")
```

Ahora, para ver funcionar el juego, llamamos a la función, escribiendo el nombre de la función y entre parentesis vamos a pasar el valor de x , en este caso escogimos $x = 100$, entonces el funcionamiento del juego sería algo así:

Figura 15



```
adivina_el_numero(100)

¡Bienvenido al juego!
¿Cual es tu nombre?:
Esteban
el objetivo es adivinar el número generado
Adivina el numero entre (1, 100):
80
fallaste, intenta con un número más pequeño
Adivina el numero entre (1, 100):
50
fallaste, intenta con un número más pequeño
Adivina el numero entre (1, 100):
20
fallaste, intenta con un número más grande
Adivina el numero entre (1, 100):
30
fallaste, intenta con un número más grande
Adivina el numero entre (1, 100):
40
fallaste, intenta con un número más grande
Adivina el numero entre (1, 100):
45
fallaste, intenta con un número más grande
Adivina el numero entre (1, 100):
48
fallaste, intenta con un número más grande
Adivina el numero entre (1, 100):
49
¡Muy bien! Esteban haz hallado el numero 49. Felicidades
Número de intentos fallidos 7
```

4. Resultados

Pudimos realizar un programa que mediante la definición de unas variables puede hacer un ciclo el cuál se repite hasta obtener un resultado determinado al azar, haciendo que este vaya cambiando cada vez que se ejecute, esto cumple con el hecho de poder servir como un juego, dónde de un conjunto se elije un número al azar y esté debe ser adivinado por el jugador. De esta forma podemos ver como python puede ser usado tanto con fines meramente académicos pero también lúdicos.

5. Conclusión

Pues bien, creo que finalmente, alcanzamos no solamente llegar a nuestro objetivo, sino que durante el proceso, de igual manera seguir aprendiendo de este lenguaje de programación “Python”, pues si bien en otro momento, por alguna razón ya habíamos visto un poco el como funcionaba estelenguaje, pero pues como bien sabemos, nunca dejamos de aprender, y esto

no fué la excepción, ya que, tuvimos que investigar más cosas y relacionarlas con lo que queríamos lograr. Entonces quedamos satisfechos con el proyecto, pues se logró el objetivo y aprendimos nuevas cosas.

Referencias

- [1] . Navone, E. C. (5 de julio de 2021). free code camp. Obtenido de 6 Proyectos de Python Básicos - Curso Completo Paso a Paso: <https://youtu.be/tWnyBD2src0>