



# APLICACIÓN DE CODIGOS BINARIOS EN JUEGOS

Trabajo realizado por:

-José Rueda Guerrero

-Marcos Sánchez Marín

# Indice:

1. Introducción
2. Código necesario
3. Explicacion del código
4. Conclusión

## 1 . Introducción

En la actualidad, la mayor parte de los juegos, por no decir todos, están compuestos de códigos binarios, los cuales son aplicados en los juegos para su funcionamiento.

Por nuestra parte, escogeremos un código binario de un juego sencillo, la adivinación de un número entre 0 y 20.

Durante el juego, se muestra un constante dialogo, el cual va programado y expresa pistas al jugador para adivinar el número escogido.

## 2. Código

```
In [ ]: 1. # Este es el juego de adivinar el número.

2. import random

3.

4. intentosRealizados = 0

5.

6. print('¡Hola! ¿Cómo te llamas?')

7. miNombre = input()

8.

9. número = random.randint(1, 20)

10. print('Bueno, ' + miNombre + ', estoy pensando en un número entre 1 y 20.')

11.

12. while intentosRealizados < 6:

13.     print('Intenta adivinar.') # Hay cuatro espacios delante de print.

14.     estimación = input()

15.     estimación = int(estimación)

16.

17.     intentosRealizados = intentosRealizados + 1

18.

19.     if estimación < número:

20.         print('Tu estimación es muy baja.') # Hay ocho espacios delante de print.

21.

22.     if estimación > número:

23.         print('Tu estimación es muy alta.')

24.

25.     if estimación == número:

26.         break

27.

28. if estimación == número:

29.     intentosRealizados = str(intentosRealizados)

30.     print('¡Buen trabajo, ' + miNombre + '! ¡Has adivinado mi número en ' + intentosRealizados + ' intentos!')

31.

32. if estimación != número:

33.     número = str(número)

34.     print('Pues no. El número que estaba pensando era ' + número)
```

### 3. Explicación del código

Esta función generará un número aleatorio para que el usuario adivine.

La línea 4 crea una nueva variable llamada intentos Realizados.

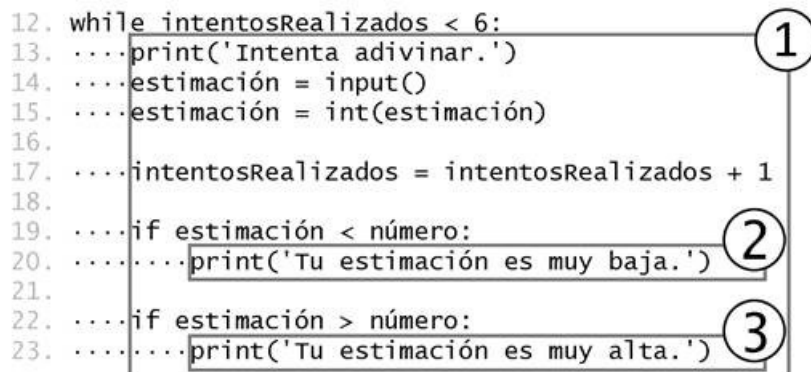
La línea 7 permite al usuario escribir su nombre y lo almacena en la variable mi Nombre

la línea 9 va a guardar el número secreto entre los valores estipulados (1, 20)

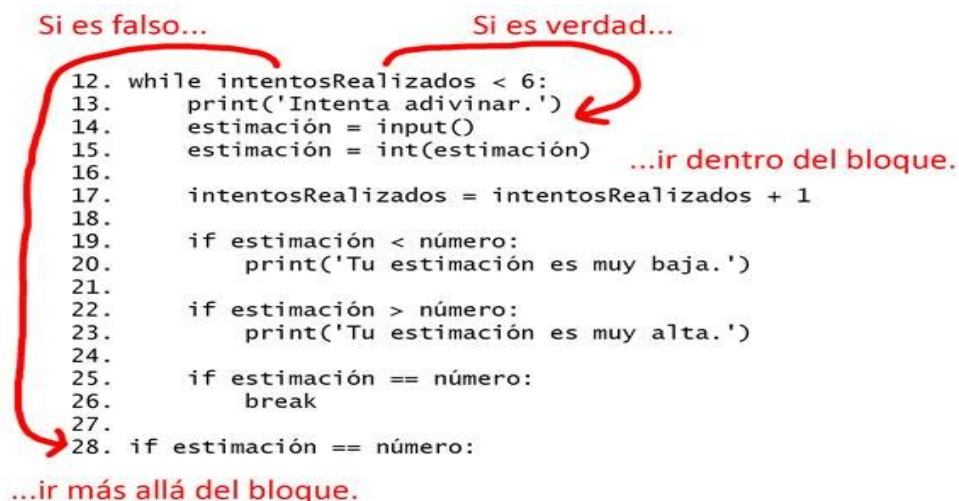
Línea 10 la función print()llama al jugador por su nombre, y le dice que la computadora está pensando un número aleatorio.

Línea 12 que indica el comienzo de un bucle que estipula que mientras los intentos sean menores de 6:

```
12. while intentosRealizados < 6:
13.     print('Intenta adivinar.')
14.     estimación = input()
15.     estimación = int(estimación)
16.
17.     intentosRealizados = intentosRealizados + 1
18.
19.     if estimación < número:
20.         print('Tu estimación es muy baja.')
21.
22.     if estimación > número:
23.         print('Tu estimación es muy alta.')
```



La sentencia while (mientras) indica el comienzo de un bucle. Los bucles pueden ejecutar el mismo código repetidas veces. Cuando la ejecución llega hasta una sentencia while, evalúa la condición junto a la palabra reservada while. Si la condición se evalúa a True, la ejecución se mueve dentro del bloque while. (En tu programa, el bloque while comienza en la línea 13.) Si la condición se evalúa a False, la ejecución se mueve hasta debajo del bloque while. (En “Adivina el Número”, la primera línea luego del bloque while es la línea 28.)



```
12. while intentosRealizados < 6:
13.     print('Intenta adivinar.')
14.     estimación = input()
15.     estimación = int(estimación)
16.
17.     intentosRealizados = intentosRealizados + 1
18.
19.     if estimación < número:
20.         print('Tu estimación es muy baja.')
21.
22.     if estimación > número:
23.         print('Tu estimación es muy alta.')
24.
25.     if estimación == número:
26.         break
27.
28. if estimación == número:
```

Si es falso... Si es verdad... ...ir dentro del bloque. ...ir más allá del bloque.

#### 4. **CONCLUSIÓN**

Al realizar este trabajo hemos aprendido nuevos usos que desconocíamos de Python y que podemos aplicarlos para crear distintos juegos sencillos, como en este caso la adivinación de un número, aplicando un código binario.