

Ejercicio 1 - Bagels

En Bagels, un juego de lógica deductiva, debes adivinar un número secreto de tres dígitos basándote en pistas. El juego ofrece una de las siguientes pistas en respuesta a tu intento: “*Pico*” cuando tu intento tiene un dígito correcto en el lugar incorrecto, “*Fermi*” cuando tu intento tiene un dígito correcto en el lugar correcto, y “*Bagels*” si tu intento no tiene ningún dígito correcto. Tienes 10 intentos para adivinar el número secreto.

Ejemplo de uso de Bagels

Las partes con el símbolo > son inputs del usuario

```
1 Estoy pensando en un número de 3 dígitos. Intenta adivinar cuál es.
2 Aquí tienes algunas pistas:
3 Cuando digo: Significa:
4 "Pico" Un dígito es correcto pero está en la posición incorrecta.
5 "Fermi" Un dígito es correcto y está en la posición correcta.
6 "Bagels" Ningún dígito es correcto.
7 He pensado en un número.
8 Tienes 10 intentos para adivinarlo.
9 Intento #1:
10 > 123
11 Pico
12 Intento #2:
13 > 456
14 Bagels
15 Intento #3:
16 > 178
17 Pico Pico
18 --snip--
19 Intento #7:
20 > 791
21 Fermi Fermi
22 Intento #8:
23 > 701
24 Lo has conseguido!
25 ¿Quieres jugar de nuevo? (sí o no)
26 > no
27 Gracias por jugar!
```

Cómo funciona

Ten en cuenta que este programa no utiliza valores enteros, sino valores de tipo cadena que contienen dígitos numéricos. Por ejemplo, ‘426’ es un valor diferente de 426. Necesitamos hacer esto porque

estamos realizando comparaciones de cadenas con el número secreto, no operaciones matemáticas. Recuerda que '0' puede ser un dígito inicial: la cadena '026' es diferente de '26', pero el número entero 026 es igual que 26.

```
1 import random
2
3 NUM_DIGITS = 3 # (!) Prueba a cambiar esto.
4 MAX_GUESSES = 10 # (!) Prueba a cambiar esto
5
6
7 def main():
8     """Función principal que maneja la lógica del juego."""
9     # TODO: Explicar las normas del juego
10
11     while True: # Main game loop.
12         secretNum = getSecretNum() # Esta función generará el número
13         # TODO: Imprimir que el número ha sido generado.
14         secreto.
15
16         numGuesses = 1
17         while numGuesses <= MAX_GUESSES:
18             guess = ''
19             # TODO: Solicitar al usuario que ingrese una suposición vá
20             lida.
21
22             clues = getClues(guess, secretNum) # Generar pistas con
23             base en la suposición.
24             print(clues)
25             numGuesses += 1
26
27             if guess == secretNum:
28                 break # Adivinó correctamente, salir del bucle.
29             if numGuesses > MAX_GUESSES:
30                 # TODO: Informar al jugador que se ha quedado sin
31                 intentos.
32                 pass
33
34             # Preguntar al jugador si desea jugar de nuevo.
35             # TODO: Solicitar si quiere jugar de nuevo y gestionar la
36             respuesta.
37             pass
38
39             # TODO: Despedida final cuando el jugador termine.
40             pass
41
42 def getSecretNum():
43     """Devuelve un número secreto de NUM_DIGITS dígitos únicos en
44     formato de cadena."""
45     numbers = list('0123456789') # Crea una lista con los numeros del
```

```
1 al 9.
41 random.shuffle(numbers) # Los pone en orden aleatorio.
42 # TODO: Implementar la lógica para generar un número aleatorio ú
    nico.
43 pass
44
45
46 def getClues(guess, secretNum):
47     """Devuelve una cadena con las pistas pico, fermi, bagels para una
        pareja de suposición y número secreto."""
48     # TODO: Implementar la lógica para generar las pistas basadas en la
        suposición.
49     pass
50
51
52 # Si el programa se ejecuta directamente (en lugar de importarse),
    iniciar el juego:
53 if __name__ == '__main__':
54     main()
```

Haz pruebas en el código

Después de ingresar el código fuente y ejecutarlo algunas veces, intenta hacer cambios experimentales en él. Los comentarios marcados con (!) tienen sugerencias de pequeños cambios que puedes realizar. Por tu cuenta, también puedes intentar averiguar cómo hacer lo siguiente:

- Cambiar la cantidad de dígitos del número secreto modificando NUM_DIGITS.
- Cambiar el número de intentos que el jugador tiene modificando MAX_GUESSES.
- Intentar crear una versión que incluya letras además de dígitos en el número secreto.

Contesta a las preguntas

- ¿Qué sucede cuando cambias la constante NUM_DIGITS?
- ¿Qué sucede cuando cambias la constante MAX_GUESSES?
- ¿Qué sucede si configuras NUM_DIGITS a un número mayor que 10?
- ¿Qué sucede si reemplazas getSecretNum() en la función main() por un valor fijo, como secretNum = '123'?
- ¿Qué sucede si eliminas o comentas random.shuffle(numbers) en la función getSecretNum()?