

ACTIVIDAD 6 Búsqueda de Números con Listas y Random

⌂ B I <> ↺ 🖼️ 🔍 ⌨️ — Ψ 😊 📄

COLIN REYES BRIAN GABRIEL\
MATRICULA: 379494 \
FECHA: 22/09/25 23/09/25

COLIN REYES BRIAN GABRIEL
MATRICULA: 379494
FECHA: 22/09/25 23/09/25

Ejercicio 1 Adivina el Número (3 Intentos)

Descripción

Juego donde el usuario intenta adivinar un número aleatorio entre 1 y 10 en un máximo de 3 intentos. Da pistas de si el número es mayor o menor, valida la entrada con try-except, y al final muestra cuántas partidas ganó o perdió, permitiendo repetir el juego si el usuario lo desea.

Solución

```
import random
```

```
def jugar():  
    numero = random.randint(1, 10)  
    intentos = 3  
  
    for i in range(1, intentos + 1):  
        try:  
            guess = int(input(f"Intento {i}: Ingresa un número entre 1 y 10: "))  
  
            if guess < 1 or guess > 10:  
                print("Debes ingresar un número dentro del rango (1-10).")  
                continue  
  
            if guess == numero:  
                print("¡Ganaste!")  
                return True  
            elif guess < numero:  
                print("El número es mayor.")  
            else:  
                print("El número es menor.")  
  
        except ValueError:  
            print("Entrada inválida, ingresa un número entero.")  
  
    print(f"Perdiste. El número era {numero}.")  
    return False
```

```
def main():  
    ganados = 0  
    perdidos = 0  
  
    while True:  
        if jugar():  
            ganados += 1  
        else:  
            perdidos += 1  
  
        print(f"\nJuegos ganados: {ganados} | Perdidos: {perdidos}")  
  
        otra = input("\n¿Quieres jugar de nuevo? (s/n): ").lower()  
        if otra != "s":  
            print("Gracias por jugar. ¡Hasta luego!")  
            break
```

```
if __name__ == "__main__":
    main()
```

```
Intento 1: Ingresa un número entre 1 y 10: 5
El número es menor.
Intento 2: Ingresa un número entre 1 y 10: 2
El número es mayor.
Intento 3: Ingresa un número entre 1 y 10: 4
El número es menor.
Perdiste. El número era 3.
```

```
Juegos ganados: 0 | Perdidos: 1
```

```
¿Quieres jugar de nuevo? (s/n): s
```

```
Intento 1: Ingresa un número entre 1 y 10: 4
El número es mayor.
Intento 2: Ingresa un número entre 1 y 10: 8
El número es menor.
Intento 3: Ingresa un número entre 1 y 10: 7
El número es menor.
Perdiste. El número era 5.
```

```
Juegos ganados: 0 | Perdidos: 2
```

```
¿Quieres jugar de nuevo? (s/n): n
Gracias por jugar. ¡Hasta luego!
```

Ejercicio 2 Busca Número en Lista (3 Versiones)

✓ Descripción

Este programa genera una lista de 10 números aleatorios únicos usando tres métodos diferentes: validación con ciclos (while), random.sample() y random.shuffle(). Después selecciona un número al azar de la lista y el usuario debe adivinar su índice en un máximo de 3 intentos, validando la entrada con try-except y mostrando mensajes de acierto o error.

✓ Solución

```
import random
random.seed(42)
```

```
def llena_lista_shuffle():
    lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    random.shuffle(lista)
    return lista
```

```
def llena_lista_while():
    lista = []
    i = 0
    while i < 10:
        num = random.randint(1, 10)
        if num in lista:
            continue
        lista.append(num)
        i += 1
    return lista
```

```
def llena_lista_sample():
    lista = random.sample(range(1, 11), 10)
    return lista
```

```
lista = llena_lista_shuffle()
print(lista)
```

```
[8, 4, 3, 9, 6, 7, 10, 5, 1, 2]
```

```
lista = llena_lista_while()
print(lista)
```

```
[10, 7, 1, 2, 4, 9, 8, 5, 3, 6]
```

```
listaa = llena_lista_sample()
print(listaa)
```

```
[10, 6, 4, 9, 1, 7, 2, 8, 5, 3]
```

```
def jugar(lista):
    print("Lista generada:", lista)

    numero = random.choice(lista)
    print("El número a buscar es:", numero)

    intentos = 3
    for i in range(1, intentos + 1):
        try:
            indice = int(input(f"¿En qué índice está? (Intento {i}): "))
            if indice < 0 or indice >= len(lista):
                print("Índice fuera de rango. Intenta de nuevo.")
                continue

            if lista[indice] == numero:
                print(f"¡Ganaste! El {numero} está en el índice {indice}.")
                return
            else:
                print("Incorrecto.")
        except ValueError:
            print("⚠ Entrada inválida, escribe un número.")

    print(f"Perdiste. El {numero} estaba en el índice {lista.index(numero)}.")
```

```
if __name__ == "__main__":
    print("\n--- Versión con shuffle ---")
    lista_shuffle = llena_lista_shufle()
    jugar(lista_shuffle)

    print("\n--- Versión con while ---")
    lista_while = llena_lista_while()
    jugar(lista_while)

    print("\n--- Versión con sample ---")
    lista_sample = llena_lista_sample()
    jugar(lista_sample)
```

```
--- Versión con shuffle ---
Lista generada: [1, 8, 9, 10, 3, 6, 4, 5, 7, 2]
El número a buscar es: 10
¿En qué índice está? (Intento 1): 4
Incorrecto.
¿En qué índice está? (Intento 2): 2
Incorrecto.
¿En qué índice está? (Intento 3): 6
Incorrecto.
Perdiste. El 10 estaba en el índice 3.
```

```
--- Versión con while ---
Lista generada: [5, 2, 10, 3, 9, 4, 8, 7, 6, 1]
El número a buscar es: 3
¿En qué índice está? (Intento 1): 2
Incorrecto.
¿En qué índice está? (Intento 2): 5
Incorrecto.
¿En qué índice está? (Intento 3): 7
Incorrecto.
Perdiste. El 3 estaba en el índice 3.
```

```
--- Versión con sample ---
Lista generada: [1, 6, 7, 3, 10, 2, 8, 9, 5, 4]
El número a buscar es: 9
¿En qué índice está? (Intento 1): 2
Incorrecto.
¿En qué índice está? (Intento 2): 5
Incorrecto.
¿En qué índice está? (Intento 3): 7
¡Ganaste! El 9 está en el índice 7.
```

✓ Conclusión

En ambos programas se aplicaron fundamentos de generación de números aleatorios, control de ciclos (while y for) y validación de entradas con try-except. Se desarrolló la lógica de interacción con el usuario mediante juegos, promoviendo la práctica de estructuras de control, condicionales y manejo de errores, logrando que el usuario pueda experimentar con intentos limitados y retroalimentación inmediata sobre sus acciones.

✓ GITHUB

<https://github.com/resu0044/Phyton>