



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en computación

### Ingeniero en Software y tecnologías emergentes

**Materia:** Programación en Python / Clave **COM432**

**Alumno:** Emiliano Nevarez Palma

**Matrícula:** 375295

**Maestro:** Pedro Núñez Yépiz

**Actividad No. 8**

**Tema - Unidad:** Listas y Funciones

**Ensenada Baja California a 01 de abril del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. Introducción

En esta actividad se presentan dos programas en donde tienes que adivinar un numero dentro de un rango. Aplicando funciones como random, range, función para crear una nueva lista utilizando ciclos, rangos y listas. Por último, tener 100% validado los dos programas

### 2. Competencia

Con estos tipos de programa aprenderemos a mejorar cada vez en el tema de las funciones, ciclos y sobre todo en las listas.

### 3. Fundamentos

En Python, una lista es una estructura de datos que permite almacenar una colección ordenada de elementos. Cada elemento de la lista puede ser de cualquier tipo de datos, como números, cadenas de texto, booleanos, etc.

Las listas en Python son mutables, lo que significa que pueden ser modificadas después de su creación. Puedes agregar elementos, eliminar elementos o cambiar elementos individuales de la lista.

Las listas se definen utilizando corchetes [] y los elementos se separan por comas. Aquí hay un ejemplo de cómo definir una lista en Python:

```
mi_lista = [1, 2, 3, "cuatro", True]
```

### 4. Procedimiento

Estos son los problemas a realizar:

1.- Programa en python que genere un número entre el 1 y 10 (no visible) preguntarle al usuario que número cree que generó la computadora, el usuario tendrá 3 oportunidades de adivinar. Decir si adivino o si falló en sus 3 intentos.

NOTA: 100% VALIDADO (usar función para validar números)

NOTA2: El usuario podrá jugar cuantas veces lo desee, al final del juego desplegar cantidad de ganados y perdidos.

2.- El juego Busca Número muestra una lista de 10 numeros, sin mostrar su contenido, Al usuario se le muestra un número que se generó aleatoriamente y el usuario tendrá 3 intentos de adivinar en qué índice del arreglo se encuentra.

El usuario recibirá un mensaje que diga GANASTE, PERDISTE, TIENES UN NUEVO INTENTO

El usuario podrá jugar cuantas veces lo desee.

NOTA: La lista se deberá llenar con números aleatorios del 1 al 10 no repetidos.

(Hacer una función que regrese la lista con los 10 números sin repetir)

# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

NOTA: REALIZA 3 VERSIONES DIFERENTES DEL LLENADO DE LA LISTA ALEATORIA

VERSIÓN A) Usar ciclos para validar los repetidos, y sólo random para generar los números dentro del rango

VERSIÓN B) Usa funciones de la librería Random para llenar la lista con los número sin repetir

### 5. Resultados y Conclusiones

Este fue el resultado del programa realizado, estuvo poco complicado entender lo de las listas pero aquí está dicho código de los 2 programas:

Programa 1:

```
[ ] import random

def validar_numero(numero):
    if numero.isdigit():
        numero = int(numero)
        if 1 <= numero <= 10:
            return numero
    return None

ganados = 0
perdidos = 0

while True:
    numero_secreto = random.randint(1, 10)
    intentos = 0
    print("Adivina el número secreto (entre 1 y 10) en 3 intentos:")
    while intentos < 3:
        intentos += 1
        numero_usuario = input(f"Intento #{intentos}: ")
        numero_usuario = validar_numero(numero_usuario)
        if numero_usuario is None:
            print("Ingresa un número válido (entre 1 y 10).")
        elif numero_usuario == numero_secreto:
            print("¡Felicitaciones! Adivinaste el número secreto.")
            ganados += 1
            break
        else:
            print("Ese no es el número secreto.")
    else:
        print(f"Lo siento, perdiste. El número secreto era {numero_secreto}.")
        perdidos += 1
    jugar_de_nuevo = input("¿Quieres jugar de nuevo? (s/n) ")
    if jugar_de_nuevo.lower() != "s":
        break

print(f"Ganados: {ganados}, Perdidos: {perdidos}")
```

# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Programa 2:

```
[ ] # VERSIÓN A) Usar ciclos para validar los repetidos, y sólo random para generar los números dentro del rango
def generar_lista_v1():
    numeros = []
    while len(numeros) < 10:
        num = random.randint(1, 10)
        if num not in numeros:
            numeros.append(num)
    return numeros
```

```
[ ] # VERSIÓN B) Usa funciones de la librería Random para llenar la lista con los número sin repetir
def generar_lista_v2():
    numeros = random.sample(range(1, 11), 10)
    return numeros
```

```
▶ # Juego con la Version A
def jugar():
    numeros = generar_lista_v1()
    indice = random.randint(0, 9)
    intentos = 3
    while intentos > 0:
        print(f"Tienes {intentos} intentos para adivinar en qué índice se encuentra el número.")
        adivinanza = int(input("Ingresa tu número: "))
        if adivinanza == indice:
            print("¡GANASTE!")
            return
        else:
            print("No es el índice correcto.")
            intentos -= 1
            if intentos > 0:
                print("Tienes un nuevo intento.")
    print(f"PERDISTE. El número estaba en el índice {indice}.")

# Jugar mientras el usuario lo desee
while True:
    print("¡Bienvenido al juego Busca Número!")
    jugar()
    continuar = input("¿Quieres jugar de nuevo? (s/n): ")
    if continuar.lower() != "s":
        break
print("¡Gracias por jugar!")
```

# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

```
[ ] # Juego con la version B
def jugar():
    numeros = generar_lista_v2()
    indice = random.randint(0, 9)
    intentos = 3
    while intentos > 0:
        print(f"Tienes {intentos} intentos para adivinar en qué índice se encuentra el número.")
        adivinanza = int(input("Ingresa tu número: "))
        if adivinanza == indice:
            print("¡GANASTE!")
            return
        else:
            print("No es el índice correcto.")
            intentos -= 1
            if intentos > 0:
                print("Tienes un nuevo intento.")
    print(f"PERDISTE. El número estaba en el índice {indice}.")

# Jugar mientras el usuario lo desee
while True:
    print("¡Bienvenido al juego Busca Número!")
    jugar()
    continuar = input("¿Quieres jugar de nuevo? (s/n): ")
    if continuar.lower() != "s":
        break
print("¡Gracias por jugar!")
```

## 6. Anexos

PDF con el código ya realizado

[https://drive.google.com/file/d/1\\_ffNoBzGwRnUmZ5ug9fGRy43dtqI-Pg7/view?usp=sharing](https://drive.google.com/file/d/1_ffNoBzGwRnUmZ5ug9fGRy43dtqI-Pg7/view?usp=sharing)

Resultado programa 1:

```
if jugar_de_nuevo.lower() != "s":
    break

print(f"Ganados: {ganados}, Perdidos: {perdidos}")
```

☞ Adivina el número secreto (entre 1 y 10) en 3 intentos:  
Intento #1: 1  
Ese no es el número secreto.  
Intento #2: 23  
Ingresa un número válido (entre 1 y 10).  
Intento #3: 5  
¡Felicitaciones! Adivinaste el número secreto.  
¿Quieres jugar de nuevo? (s/n) n  
Ganados: 1, Perdidos: 0

# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Resultado del programa 2 con la versión A:

```
if continuar.lower() != "s":  
    break  
print("¡Gracias por jugar!")
```

↳ ¡Bienvenido al juego Busca Número!  
Tienes 3 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 3  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 2 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 5  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 1 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 7  
No es el índice correcto.  
PERDISTE. El número estaba en el índice 8.  
¿Quieres jugar de nuevo? (s/n): s  
¡Bienvenido al juego Busca Número!  
Tienes 3 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 1  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 2 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 3  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 1 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 6  
No es el índice correcto.  
PERDISTE. El número estaba en el índice 7.  
¿Quieres jugar de nuevo? (s/n): n  
¡Gracias por jugar!

### Resultado del programa 2 con la versión B

```
break  
print("¡Gracias por jugar!")
```

↳ ¡Bienvenido al juego Busca Número!  
Tienes 3 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 1  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 2 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 4  
No es el índice correcto.  
Tienes un nuevo intento.  
Tienes 1 intentos para adivinar en qué índice se encuentra el número.  
Ingresa tu número: 6  
¡GANASTE!  
¿Quieres jugar de nuevo? (s/n): n  
¡Gracias por jugar!

## 7. Referencias

Libro Python por Guido van Rossum

<https://drive.google.com/file/d/1Rp4EUHzbGoej-Fw3edDM34vA0FQyU16q/view>