



LABORATORIO 1 PROGRAMACION

CARLOS MANUEL DIAZ, ALVARO ANDRES GUZMAN

Resumen— En esta práctica de laboratorio se presenta el desarrollo de un juego de preguntas y respuestas tipo "Preguntados", implementado en Python. El objetivo fue aplicar diversas estructuras de programación, como variables, condicionales, ciclos, así como estructuras de datos como listas, tuplas y diccionarios. Se propuso el desarrollo de dos categorías de preguntas, con diez preguntas en cada una, almacenadas en listas y presentadas de forma aleatoria mediante el uso de tuplas y diccionarios. El juego, diseñado para un único jugador, asegura que las preguntas respondidas correctamente no se repitan y permite al usuario finalizar el juego en cualquier momento. Esta práctica refuerza el uso de estructuras de programación en Python.

Palabras clave—JUEGO, PYTHON, LISTAS, DICCCIONARIOS, TUPLAS, FUNCION, ESTRUCTURAS.

I.INTRODUCCIÓN

Esta práctica de laboratorio se centra en la implementación de un juego de preguntas y respuestas tipo "Preguntados" utilizando Python. El objetivo es aplicar estructuras básicas de programación, como variables, condicionales, ciclos y el manejo de listas, tuplas y diccionarios. A través de este proyecto, se busca no solo entretenerte, sino también consolidar los conocimientos adquiridos en la materia, promoviendo el uso de la lógica de programación.

• OBJETIVOS GENERAL

Desarrollar un juego de preguntas y respuestas tipo "Preguntados" en Python que aplique estructuras básicas de programación, permitiendo al usuario interactuar y consolidar su conocimiento sobre lógica de programación y el uso de estructuras de datos.

ESPECÍFICOS

- Emplear diversas estructuras de datos para almacenar y gestionar las preguntas y respuestas del juego, asegurando la correcta aleatorización y selección de preguntas.
- Emplear condicionales y ciclos para desarrollar la mecánica del juego, permitiendo al usuario responder preguntas y recibir puntuación de su desempeño.
- Diseñar una interfaz de usuario en terminal que permita al jugador interactuar con el juego de manera fluida, facilitando el inicio, el progreso y la finalización de este en cualquier momento.

II. INSTRUMENTOS

- II. Computador
- II. Entornos de desarrollo como visual studio
- II. PYTHON
- II. Lenguaje C++
- III. Draw.io

III. MÉTODOS.

- Implementar una versión reducida para la terminal de juego preguntados (trivia crack) mediante la herramienta Visual studio con la extensión



PYTHON. El programa debe cumplir una serie de requisitos los cuales son:

- Jugador único al iniciar el programa el Usuario ingresará su nombre el cual será usado durante el proceso del juego.
- Dos categorías cada una con 10 preguntas las cuales se iniciarán de manera aleatoria
- El usuario podrá salir del programa en el momento que lo desee
- El programa debe ser legible, correcto uso de recursos.
- Implementar estructuras vistas en clase y elegir la que mejor se adapte al programa es decir poco uso de memoria y legibilidad

Código desarrollado para el juego de preguntas:

[JuegoDePreguntas.py](#)

Diagrama de flujo:

<https://drive.google.com/open?id=1gr7AZ4KDzidpNaGNDQK3-m0gB5tYCtqT>

Estructura y explicación del código

- Se importa el módulo random el cual tiene como función generar las categorías y preguntas aleatorias en nuestro programa.
- Se define un diccionario llamado cuestionario el cual contiene dos categorías.

```
cuestionario = {
    "Historia": [
        ("¿En qué año comenzó la Primera Guerra Mundial?", "A) 1914", "B) 1918", "C) 1920", "D) 1939"),
        ("a"),
        ("¿Quién fue el primer presidente de los Estados Unidos?", "A) Abraham Lincoln", "B) Thomas Jefferson", "C) George Washington", "D) John Adams"),
        ("c"),
        ("¿Qué civilización construyó las pirámides de Egipto?", "A) Romanos", "B) Griegos", "C) Egipcios", "D) Persas"),
        ("c"),
        ("¿Qué famosa revolución comenzó en 1789?", "A) Revolución Americana", "B) Revolución Francesa", "C) Revolución Industrial", "D) Revolución Rusa"),
        ("b"),
        ("¿Quién fue el líder del movimiento independentista de la India?", "A) Nelson Mandela", "B) Simón Bolívar", "C) Mahatma Gandhi", "D) Winston Churchill"),
        ("c"),
        ("¿En qué país ocurrió la Revolución Industrial?", "A) Alemania", "B) Francia", "C) Inglaterra", "D) Estados Unidos"),
        ("c"),
        ("¿Qué famoso barco se hundió en 1912 tras chocar con un iceberg?", "A) Queen Mary", "B) Lusitania", "C) Titanic", "D) Britannic"),
        ("c"),
        ("¿Quién fue el primer hombre en pisar la luna?", "A) Yuri Gagarin", "B) Neil Armstrong", "C) Buzz Aldrin", "D) John Glenn"),
        ("b"),
        ("¿En qué año llegó Cristóbal Colón a América?", "A) 1492", "B) 1500", "C) 1453", "D) 1519"),
        ("a"),
        ("¿Qué país construyó la Gran Muralla?", "A) Japón", "B) China", "C) India", "D) Corea"),
        ("b"),
    ]
}
```

Se define un diccionario llamado cuestionario, que almacenará dos categorías, cada una con 10 preguntas. Cada pregunta se representa como una tupla que incluye su respuesta, con el fin de lograr una mayor legibilidad y un menor consumo de memoria en el programa.

- Se definen funciones para optimizar y mejorar la legibilidad del juego las funciones están conformadas así:

Función para mostrar pregunta y opciones de respuesta:

```
def mostrar_pregunta(pregunta, opciones):
    print(f"\n{pregunta}")
    for opcion in opciones:
        print(opcion)
```

Se define un diccionario llamado cuestionario, que almacenará dos categorías, cada una con 10 preguntas. Cada pregunta se representa como una tupla que incluye su respuesta, con el fin de lograr una mayor legibilidad y un menor consumo de memoria en el programa

Función para obtener respuesta o salir del código:

```
def obtener_resposta():
    return input("Ingrese su respuesta (o ingresa '0' para salir): ").lower()
```

Solicita al usuario que ingrese su respuesta y la retorna de forma minúscula independientemente si el usuario ingreso una mayúscula o presionar 0 para salir del juego.

Función para validar respuesta correcta:

```
def validar_resposta(resposta, resposta_correcta):
    return resposta == resposta_correcta
```

Compara la respuesta ingresada por el usuario con la respuesta correcta definida en la tupla.



Función para eliminar pregunta respondida y categoría vacía:

```
def eliminar_pregunta(categoría, pregunta):  
    cuestionario[categoría].remove(pregunta)  
    if not cuestionario[categoría]: # Si la  
        del cuestionario[categoría]
```

Elimina la pregunta correspondiente a categoría y si la categoría queda vacía la borrara del diccionario todo esto gracias a la función remove para así evitar que se repitan las preguntas ya respondidas

Función para calcular puntuación del usuario:

```
def calcular_puntuacion(correctas, total_preguntas):  
    return (correctas / total_preguntas) * 100
```

Calcula la puntuación del usuario en porcentaje de 0 a 100% esto con base a las respuestas correctas contestadas por el usuario.

Lógica principal del juego:

Ingresar nombre:

```
nombre_usuario = input("Por favor, ingresa tu nombre: ")  
print(f"\nBienvenido/a {nombre_usuario} al juego de preguntas al azar\n")
```

Solicita al usuario que ingrese su nombre, el cual será almacenado en la variable nombre_usuario, e imprimirá un mensaje de bienvenida al juego.

Conteo de preguntas:

```
total_preguntas = 0  
  
for categoría in cuestionario:  
    total_preguntas += len(cuestionario[categoría])
```

Se cuenta el total de preguntas disponibles en todas las categorías del cuestionario.

Mientras haya preguntas disponibles se sigue seleccionando:

```
while cuestionario:  
    categoría = random.choice(list(cuestionario.keys()))  
    pregunta, opciones, respuesta_correcta = random.choice(cuestionario[categoría])  
  
    mostrar_pregunta(pregunta, opciones)
```

Este bucle se ejecuta mientras haya preguntas disponibles en el cuestionario. Mediante la función random.choice, que selecciona aleatoriamente un elemento de la tupla , se elige una categoría y una pregunta de esa categoría al azar. Luego, la pregunta seleccionada se muestra al usuario.

Desea salir o continuar jugando:

```
while True: # Bucle para validar la respuesta  
    respuesta = obtener_respuesta() # Obtener respuesta del usuario  
  
    if respuesta == "0": # Salir del juego  
        if input("¿Estás seguro(a) que deseas salir? (si/no): ").lower() == "si":  
            print("Saliendo del juego")  
            exit() # Salir del bucle  
        else:  
            print("¡Continuemos respondiendo las preguntas!")  
            continue
```

Implementamos una estructura if con el propósito de dar al usuario la opción de salir del programa en cualquier momento al presionar cero. En este caso, se le mostrará un mensaje preguntándole si desea salir. Si el usuario responde que sí, el programa finalizará gracias a la función exit(). Si elige continuar, se mostrará un mensaje indicando que seguirá respondiendo las preguntas.

Evaluar que respuesta ingresada sea válida:

```
if respuesta in ["a", "b", "c", "d"]:  
    break # Salir del bucle si la respuesta es válida  
else:  
    print("Opción no válida. Por favor, elige a, b, c o d.")
```

Mediante un if, se evalúa si la opción ingresada por el usuario es válida. Las respuestas deben ser 'a', 'b', 'c' o 'd'. Si se ingresa una de estas opciones, el programa sale del bucle usando break y continúa con el siguiente paso del código. Sin embargo, si se ingresa una opción no válida, el else pedirá al usuario que ingrese una opción correcta para poder continuar

Verificar si respuesta ingresada es correcta o incorrecta:

```
if validar_respuesta(respuesta, respuesta_correcta):
    print("Respuesta correcta")
    correctas += 1
else:
    print("Respuesta incorrecta")
    incorrectas += 1
```

Se utiliza un bucle interno para obtener y validar la respuesta del usuario, asegurándose de que sea válida. Se actualizan los contadores de respuestas correctas e incorrectas según sea necesario.

Elimina preguntas respondidas:

```
eliminar_pregunta(categoría, (pregunta, opciones, respuesta_correcta))
```

Llama a la función para eliminar la pregunta que el usuario acaba de responder.

Calculo puntuación:

```
puntuacion = calcular_puntuacion(correctas, total_preguntas)
print(f"\nHa terminado el juego. Respuestas correctas: {correctas}, Respuestas incorrectas: {incorrectas}
print(f"Tu puntuación de 0 a 100% es: {puntuacion:.2f}%")

if correctas == total_preguntas:
    print("\n¡Felicitaciones! Has contestado todas las preguntas correctamente.")
```

Una vez respondidas las preguntas, se calcula la puntuación y se presenta al usuario en el monitor en caso de responder todas las preguntas correctamente se presentará el mensaje de felicitaciones al usuario esto se lo puede ver reflejado en el if implementado.

Resultados al ejecutar el juego

Al correr el programa, se solicita ingresar el nombre del usuario. Luego, se elige una categoría y una pregunta al azar mediante el módulo random. Se muestra la pregunta seleccionada junto con sus opciones, y el usuario ingresa su respuesta. Posteriormente, se mostrará un mensaje indicando si la opción escogida fue correcta o incorrecta. El usuario puede decidir en cualquier momento si desea salir o continuar respondiendo. El juego terminará cuando se hayan respondido todas las preguntas de las dos categorías. Finalmente, se mostrará un mensaje reflejando

cuántas preguntas fueron contestadas de forma correcta e incorrecta, y en base a eso se calculará la puntuación del usuario de 0 a 100%.

Pruebas de escritorio realizadas:

Se realizaron varias pruebas de escritorio en las que se puede ver un correcto funcionamiento del código y que cada una de las cosas implementadas en el cumple con su función correctamente.

Prueba de escritorio solicitud del nombre del usuario y generar categoría y pregunta aleatoriamente:

```
ter.py"
Por favor, ingresa tu nombre: Andrés

Bienvenido/a Andrés al juego de preguntas al azar

¿En qué continente se encuentra el desierto del Sahara?
A) Asia
B) África
C) América
D) Oceanía
Ingrese su respuesta (o ingresa '0' para salir): b
Respuesta correcta

¿Quién fue el líder del movimiento independentista de la India?
A) Nelson Mandela
B) Simón Bolívar
C) Mahatma Gandhi
D) Winston Churchill
Ingrese su respuesta (o ingresa '0' para salir): C
```

Al iniciar el programa le pide el nombre al usuario y elige una categoría aleatoria gracias a la función random y una pregunta de esa categoría al responder indica si la opción ingresada ha sido correcta o incorrecta.



Prueba de escritorio si el usuario desea salir del programa:

Por favor, ingresa tu nombre: andres

Bienvenido/a andres al juego de preguntas al azar

¿Cuál es el metal más ligero?

- A) Aluminio
- B) Hierro
- C) Litio
- D) Cobre

Ingrrese su respuesta (o ingresa '0' para salir): 0

¿Estás seguro(a) que deseas salir? (si/no): si
Saliendo del juego

PS C:\Users\PERSONAL\Codigos laboratorio> █

Al ingresar 0 y confirmar con la palabra si el programa termina automáticamente esto debido a que en el if implementado al final si el usuario decide salir el programa finaliza mediante un exit ()

Finalización del código con informe de preguntas correctas e incorrectas y puntuación del jugador:

Ha terminado el juego. Respuestas correctas: 19, Respuestas incorrectas: 1

Tu puntuación de 0 a 100% es: 95.00%

PS C:\Users\PERSONAL\Codigos laboratorio> █

Al finalizar el juego, se muestra la puntuación obtenida, junto con la cantidad de preguntas respondidas correcta e incorrectamente. Esto refleja el correcto funcionamiento de la función para calcular la puntuación y el conteo de preguntas

En caso de ingresar una opción de respuesta no válida:

¿En qué año llegó el hombre a la Luna por primera vez?

- A) 1965
- B) 1969
- C) 1972
- D) 1962

Ingrrese su respuesta (o ingresa '0' para salir): y
Opción no válida. Por favor, elige a, b, c o d.

Ingrrese su respuesta (o ingresa '0' para salir): █

Al ingresar una opción no válida, es decir, diferente a las solicitadas por el programa, se mostrará un mensaje indicando que la opción ingresada es incorrecta y que debe elegir una de las opciones: a, b, c o d. De lo contrario, el juego no continuará mostrando más preguntas

Finalización del juego con todas las preguntas

Contestadas correctamente:

Ha terminado el juego. Respuestas correctas: 20, Respuestas incorrectas: 0
Tu puntuación de 0 a 100% es: 100.00%

¡Felicitaciones! Has contestado todas las preguntas correctamente.

PS C:\Users\PERSONAL\Codigos laboratorio> █

Al contestar todas las preguntas correctamente, al finalizar el juego se mostrará un mensaje de felicitaciones indicando que la puntuación obtenida es del 100%. Además, se mostrará un contador de preguntas, el cual reflejará que se han respondido correctamente las 20 preguntas correspondientes a las categorías definidas en el código.

VI. CONCLUSIONES

- El uso de diccionarios y tuplas permite una correcta organización y eficiencia en la gestión de las preguntas.
- **Control de flujo adecuado:** La implementación de condicionales y bucles garantiza que el juego se ejecute correctamente y se adapte a las decisiones del usuario.
- Mediante random.choice, se asegura que las preguntas y categorías se elijan al azar, brindando una experiencia única en cada partida.
- El manejo de entradas no válidas permite una interacción sin interrupciones, guiando al usuario hasta que ingrese una opción correcta.
- La función de puntuación refleja correctamente el desempeño del usuario, proporcionando retroalimentación clara al final del juego.

VII. REFERENCIAS

- «[Trivia Crack](#)», *Wikipedia* . [En línea]. Disponible: https://en.wikipedia.org/wiki/Trivia_Crack.
- [1] "Curso python desde cero" YouTube, 2024.[Online].Available: <https://www.youtube.com/watch?v=mENHDQ8SLsI&list=PLyvsggKtwbLW1j0d5yaCkRF9Axpdlhsxz>. [Accessed: 25-Sep-2024].
- *Programar, A.* a. [[@aprendeaprogramar6670](#)]. (s/f). *Curso Maestro de Python: Creemos un juego de Preguntas y Respuestas con Python* **100** #39. YouTube. Recuperado el 19 de



PROGRAMA DE
INGENIERÍA ELÉCTRICA
Acreditación de Alta Calidad.
Fes, MEX. 11/06 – 17/06/2010 (9 años)

septiembre de 2024, de
<https://www.youtube.com/watch?v=pFEs4dO63y4&t=462s>

- "Juego de preguntas en Python," YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=aBL2lh_uU2fl. [Accessed: 24-Sep-2024].

Enlace Github

<https://github.com/AG-Program/laboratorio-1-programacion.git>