

Laboratorio 1 Programación.

Integrantes:

David Gaviria Ruano

Julian David Ramirez Mayag



Profesores:

Alexander Parrado

Julián Darío Barrera

Universidad del Quindío

Facultad de ingeniería

Ingeniería Electrónica

Programación.

2024-02

Resumen

Se propuso un ejercicio en el cual se debía realizar un juego basado en "Preguntados", en el cual se debían adquirir conocimientos como crear listas, conjuntos, tuplas y diccionarios, todo realizado en tres laboratorios.

En el primer laboratorio, se planteó cómo deducir el problema y cómo implementar los conocimientos ya adquiridos al juego, sin necesidad de complicarse. Se aprendió a utilizar tuplas y conjuntos.

En el segundo laboratorio, ya con un código base, se empezaron a realizar pruebas de escritorio con solo dos preguntas de cada sección para asegurar que el código no fallará al momento de implementar las diez preguntas de cada sección. Durante este proceso, ocurrieron varios problemas al compilar el código.

En el tercer laboratorio, con ayuda de los profesores, se logró entender cómo optimizar y mejorar el código cuando se nos planteó utilizar diccionarios y cómo debía estar estructurado. Esto mejoró el código, ayudando a que no consumiera muchos recursos del computador.

Palabras claves: *Código, diccionarios, compilación, problemas, laboratorios, tuplas, lista.*

Introducción:

Esta práctica de laboratorio se desarrolla con el objetivo de implementar una versión simplificada del juego “Preguntados”, permitiendo a los usuarios interactuar y probar sus conocimientos de manera accesible y educativa.

El propósito de esta práctica es buscar que el programa sea funcional, que soporte un jugador y que contenga dos categorías de preguntas, cada una con 10 interrogantes. A lo largo de la práctica se deben utilizar listas, diccionarios, tuplas, funciones y las demás estructuras de programación vistas en clase. Se espera que el resultado sea un juego interactivo que no solo entretega, sino que también eduque al usuario.

Objetivo general

Aprender y aplicar el uso de ciclos, tuplas y conjuntos con el desarrollo de un código que simula un juego de preguntas.

Objetivos específicos:

- Implementar una versión funcional del juego de trivial, asegurando que cumpla con las especificaciones dadas, incluyendo la interacción con el usuario y la presentación de preguntas.
- Diseñar un sistema que permita almacenar y gestionar un conjunto de preguntas por categoría, asegurando que se mantenga un registro de las preguntas respondidas correctamente para evitar repeticiones.
- Hay que asegurar que las preguntas de cada categoría sean variadas y cubran diferentes aspectos del tema, evitando que el juego se vuelva monótono

Materiales

- Computador
- Laboratorios de robótica
- Software visual studio
- Asesorías dadas por los profesores
- Material de internet “videos, guías”
- GitHub

Procedimientos

Se comenzó agregando una parte del código en la cual se buscaba que, al ingresar un usuario, se le diera la bienvenida. Posteriormente, se utilizó un diccionario y tuplas para organizar el cuestionario, con ayuda de los profesores, quienes brindaron recomendaciones, ya que antes se utilizaban múltiples diccionarios separados. Al agregar las 20 preguntas (10 en cada sección), se declaró una variable para controlar el puntaje de cada usuario dependiendo de cuántas respuestas correctas obtuviera, y todos los datos se guardaron allí. Se declaró también el total de preguntas por cada sección para almacenarlas. Se inició con un ciclo while para asegurar que se cumplieran las condiciones dadas. Dentro de este ciclo, se seleccionaron preguntas aleatorias usando la función random.choice, lo que permitió que las preguntas dentro del cuestionario se seleccionaran de manera aleatoria. Si no había más preguntas, el ciclo pasaba a la siguiente categoría. Luego, se elegía una pregunta aleatoria, que se elimina del cuestionario utilizando remove. Después, se mostraban las preguntas para que el usuario las respondiera. Se declaró un ciclo for que revisaba las opciones buscando la respuesta correcta. Una vez que el usuario ingresaba su respuesta, si era correcta, se le informaba; de lo contrario, se indicaba que era incorrecta. En ese momento, se sumaba 1 a las preguntas hechas para que no se repitieran. Al final, se imprimía un resumen para que el usuario pudiera ver sus resultados, calculando su porcentaje y el número de preguntas correctas.

Código final:

```
import random

# Pedimos el nombre del usuario
usuario = input("Ingrese su nombre: ")
```

```
print(f"¡Bienvenido al juego de preguntas, {usuario}!"")  
  
# Cuestionario con categorías y preguntas (listas de tuplas)  
  
deportes = [  
  
    ("¿Cuándo anotó su primer gol Messi?",  
        ("a) 1 de mayo del 2005" "b) 10 de junio de 2004" "c) 15 de  
        enero de 2007" "d) 10 de junio de 2008"),  
  
    ("a",)  
  
    ("¿Cuántas copas mundiales tiene Pelé?",  
        ("a) 5 mundiales" "b) 2 mundiales" "c) 1 mundial" "d) 3  
        mundiales"),  
  
    ("d",)  
  
    ("¿Cuántos goles anotó Cristiano Ronaldo en toda su carrera?",  
        ("a) 900" "b) 850" "c) 1000" "d) 770"),  
  
    ("a",)  
  
    ("¿La invención del voleibol fue en?",  
        ("a) 9 de febrero de 1885" "b) 4 de enero de 1896" "c) 30  
        de febrero de 1987" "d) 15 de abril de 1882"),  
  
    ("b",)  
  
    ("Un cubo de Rubik tiene caras en rojo, naranja, verde, amarillo,  
    azul y... ¿Qué otro color?",  
        ("a) morado" "b) negro" "c) blanco" "d) gris"),  
  
    ("c",)  
  
    ("¿Quién ganó el mundial de fútbol de 2010?",  
        ("a) Alemania" "b) Francia" "c) Italia" "d) Inglaterra"))
```

("a) Francia" "b) España" "c) Brasil" "d) Colombia")

"b")

("¿En qué posición juega el cancerbero de un equipo de fútbol?")

("a) Lateral" "b) Extremo" "c) Portero" "d) Delantero")

"c")

("¿Quién se considera el mejor jugador de baloncesto de todos los tiempos?")

("a) Michael Ballack" "b) Julian Smith" "c) Kobe Bryant"
"d) Michael Jordan")

"d")

("¿Qué tipo de competición es el Giro de Italia?")

("a) Golf" "b) Ciclismo" "c) Fútbol" "d) Baloncesto")

"b")

("¿Cuánto dura un partido de balonmano?")

("a) 70 minutos" "b) 60 minutos" "c) 40 minutos" "d) 30 minutos")

"b")

]

historia = [

("¿Cuántos años duró la Segunda Guerra Mundial?")

("a) 6 años" "b) 3 años" "c) 4 años" "d) 10 años")

"a")

("¿Quién inventó la bombilla?")

("a) Thomas Edison" "b) Nikola Tesla" "c) Alexander Graham Bell" "d) Ever Kirchhoff")

"a")

("¿Cuándo acabó la II Guerra Mundial?")

("a) 1956" "b) 1930" "c) 1945" "d) 1940")

"c")

("¿En qué año se produjo la Revolución Francesa?")

("a) 1740" "b) 1789" "c) 1840" "d) 1850")

"b")

("¿En qué año llegó Cristóbal Colón a América?")

("a) 1500" "b) 1420" "c) 1492" "d) 1450")

"c")

("Según la Biblia, ¿cuántos años vivió Matusalén?")

("a) 800 años" "b) 80 años" "c) 100 años" "d) 969 años")

"d")

("¿En qué país nació Adolf Hitler?")

("a) Colombia" "b) Austria" "c) Rusia" "d) Inglaterra")

"b")

("¿Cuánto duró 'La Guerra de los Cien Años'?")

("a) 100 años" "b) 116 años" "c) 50 años" "d) 120 años")

"b")

("¿Hace cuánto se extinguieron los dinosaurios?")

("a) Hace 66 millones de años" "b) Hace 100 millones de
años" "c) Hace 40 años" "d) Todas las anteriores")

"a")

("¿Quién fue el último faraón de Egipto?"

("a) Gaviria II" "b) Amosis I" "c) Akenatón" "d) Ramsés
III")

"d")

]

Diccionario para las categorías y sus preguntas

```
cuestionario = [("Deportes" deportes) ("Historia" historia)]
```

Variables para el juego

```
puntuacion = 0
```

```
total_preguntas = 20
```

```
preguntas_hechas = 0
```

Bucle principal de preguntas

```
while preguntas_hechas < total_preguntas
```

Elegimos una categoría aleatoria

```
categoria preguntas = random choice(cuestionario)
```

Se verifica si hay preguntas disponibles

```
if preguntas == []  
  
    continue # Si no hay preguntas, pasamos a la siguiente  
    categoría  
  
  
    # Escogemos una pregunta aleatoria  
  
    pregunta opciones respuesta_correcta = random choice(preguntas)  
  
    preguntas remove((pregunta opciones respuesta_correcta)) #  
  
    Removemos la pregunta ya hecha  
  
  
    # Mostramos la categoría, la pregunta y las opciones  
  
    print(f"\nCategoría: {categoría}")  
  
    print("-----")  
  
    print(pregunta)  
  
    for opción in opciones  
  
        print(opción)  
  
  
  
  
    # Pedimos la respuesta del usuario  
  
    respuesta = input("Ingrese su respuesta (a, b, c, d): ")  
  
  
  
  
    # Verificamos si la respuesta es correcta  
  
    if respuesta == respuesta_correcta  
  
        puntuacion += 1  
  
        print("¡CORRECTO!")
```

```
else

    print(f"INCORRECTO. La respuesta correcta era:

{respuesta_correcta}")

preguntas_hechas += 1

# Resumen del juego al finalizar

print("\n-----")

print("-----RESUMEN-----")

print("-----")

print(f"Respuestas correctas: {puntuacion}")

print(f"Su puntuación es: {int((puntuacion / total_preguntas) *
100)}%")
```

Resultados del código:

```
Ingrese su nombre: David
¡Bienvenido al juego de preguntas, David!

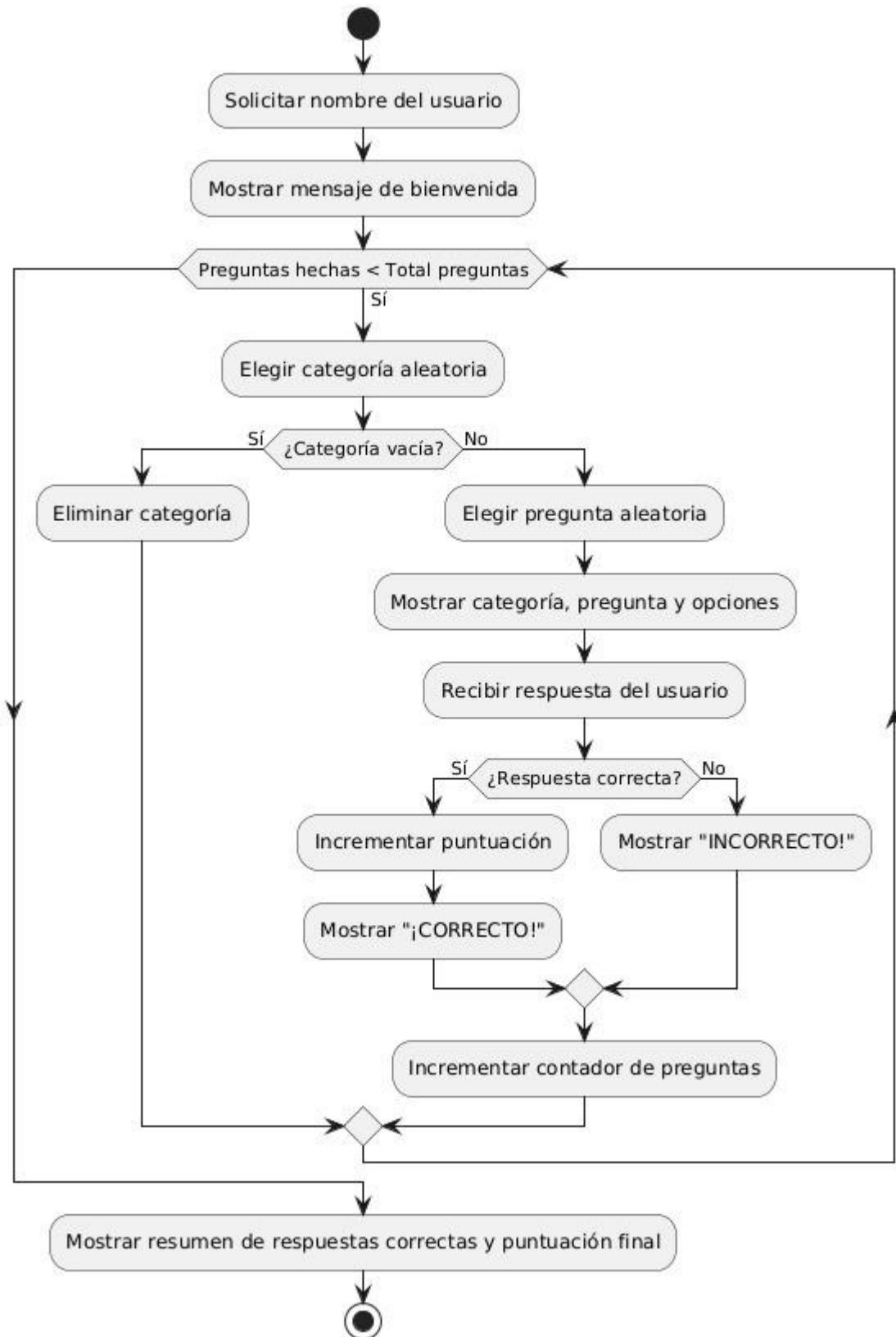
Categoría: Deportes
-----
¿La invención del voleibol fue en?
a) 9 de febrero de 1885
b) 4 de enero de 1896
c) 30 de febrero de 1987
d) 15 de abril de 1882
Ingrese su respuesta (a, b, c, d): a
INCORRECTO. La respuesta correcta era: b
```

```
Categoría: Historia
-----
¿Quién inventó la bombilla?
a) Thomas Edison
b) Nikola Tesla
c) Alexander Graham Bell
d) Ever Kirchhoff
Ingrese su respuesta (a, b, c, d): a
¡CORRECTO!

Categoría: Deportes
-----
¿En qué posición juega el cancerbero de un equipo de fútbol?
a) Lateral
b) Extremo
c) Portero
d) Delantero
Ingrese su respuesta (a, b, c, d):
```

```
-----
-----RESUMEN-----
-----  
Respuestas correctas: 5
Su puntuación es: 25%
PS C:\Users\gavir\Documents\Códigos Python> a
```

Diagrama de flujo:



Pruebas de escritorio:

Como se puede observar en la primera prueba de escritorio teníamos un código deficiente, que consumía mucho espacio, recursos e incompleto.

```
import random

secciones = ["deportes" "historia"]

# Juego de preguntas

preguntas_deportes = [
    respuestas_deporte = ["a" "d"]

preguntas_historia = [" a) 1945" "b) 1939" "c) 1965" "d) 1970 "
    a) Apolo 11" "b) Apolo 12" "c) Apolo 13" "d) Apolo 10"]

respuestas_historia = ["a" "a"]

puntuacion=0

respuestas_usuario = []
respondidas = []
```

En la segunda prueba de escritorio se obtuvo un código ya funcional con dos preguntas de cada sección, pero este código no era eficiente, no daba de forma aleatoria las preguntas, ni las secciones.

```

import random

usuario=input("ingrese su usuario")

print(f"Bienvenido al juego{usuario}")

# Definimos las preguntas, opciones y respuestas correctas en un solo diccionario para facilitar su acceso

cuestionario = {

    "Deportes"  [

        {

            "pregunta"  "¿Cuándo anotó su primer gol Messi?"

            "opciones"  ["a) 1 de mayo del 2005"  "b) 10 de junio de 2004"  "c) 15 de enero de 2007"  "d) 10 de junio de 2008"]

            "respuesta"  "a"

        }

        {

            "pregunta"  "¿Cuántas copas mundiales tiene Pelé?"

            "opciones"  ["a) 5 mundiales"  "b) 2 mundiales"  "c) 1 mundial"  "d) 3 mundiales"]

            "respuesta"  "d"

        }

    ]

    "Historia"  [

```



```
for i  pregunta in enumerate(preguntas)

    print("-----")

    print(pregunta["pregunta"])

    for opcion in pregunta["opciones"]

        print(opcion)

    guess = input("Enter (A, B, C, D): ") .lower()

    respondidas.append(guess)

    if guess == pregunta["respuesta"]

        puntuacion += 1

        print("¡CORRECTO!")

    else

        print(f"INCORRECTO! La respuesta correcta es:

{pregunta['respuesta']}")

    print()

print("-----")

print("-----RESUMEN-----")

print("-----")
```

```
print("Respuestas correctas: " puntuacion)

total_preguntas = sum(len(p) for p in cuestionario.values())

print(f"Su puntuación es: {int((puntuacion / total_preguntas) * 100)}%")
```

A este nivel ya se tiene la base para crear el código, con los fallos ya obtenidos se logró modificar hasta obtener el mejor resultado.

Conclusiones:

- Este proyecto no solo ha cumplido con los objetivos establecidos, sino que también ha proporcionado un entorno enriquecedor para el aprendizaje y el desarrollo personal de cada miembro del equipo.
- Se aprende a manejar la estructura de Python mejorando la comprensión y calidad de los códigos.
- Los diccionarios terminan siendo más eficientes para tipos de trabajos más largos, a diferencia de las listas.
- A las duplas no se le pueden agregar datos después de ya puestos.
- Utilizar funciones como, por ejemplo. choice ayudan a optimizar el código además de hacerlo más práctico.
- La implementación de este juego de trivia ha permitido implementar los conocimientos y conceptos teóricos aprendidos en clase, como el uso de listas, diccionarios, tuplas, funciones y estructuras, facilitando un aprendizaje más profundo y práctico

¿'

Referencias bibliográficas

[Preguntados] *La plataforma de trivia No1 del mundo* . (s/f).

Preguntados.com. Recuperado el 23 de septiembre de 2024, de

<https://preguntados.com/>

Code, B. [@BroCodez]. (s/f). *Create a QUIZ GAME with python* . Youtube.

Recuperado el 23 de noviembre de 2022, de

<https://www.youtube.com/watch?v=zehwgTBoV8&t=183s>

Cañedo, C. (2022, 6 de agosto). *630 preguntas de cultura general y su respuesta*. Cosmopolita. <https://www.cosmopolitan.com/es/consejos-planes/familia-amigos/a40706432/200-preguntas-de-cultura-general-y-su-respuesta/>

Dantí, C. M. (2020, julio 14). *120 Preguntas de deportes*. Mundo Deportivo.

<https://www.mundodeportivo.com/uncomo/ocio/articulo/120-preguntas-de-deportes-50646.html>