

Solución Desafío - Quiz App

Requerimiento 1

- **Paso 1:** En el módulo `validador.py` se debe definir una condición sólo con un ciclo `while` que no termine si es que la elección dada no es parte de las opciones.
- **Paso 2:** En caso que el ciclo comience se debe solicitar mediante `input` una elección nueva que sí esté entre las opciones dadas.

```
def validate(opciones, eleccion):  
    # Paso 1  
    while eleccion not in opciones:  
        # Paso 2  
        eleccion = input('Opción no válida, ingrese una de las opciones  
válidas: ')  
    return eleccion  
  
if __name__ == '__main__':  
  
    eleccion = input('Ingresa una Opción: ').lower()  
    # letras = ['a','b','c','d']  
    # pueden probar con letras también para verificar su funcionamiento.  
    numeros = ['0','1']  
    # Si se ingresan valores no válidos a eleccion debe seguir preguntando  
    validate(numeros, eleccion)
```

Requerimiento 2

- **Paso 3:** En el módulo `level.py` se debe generar lógica de nivel tomando el número de pregunta y las preguntas por nivel mediante un `if/elif/else`, si está entre el primer grupo debe ser básicas, en el 2do grupo intermedias y sino avanzadas.

```
def choose_level(n_pregunta, p_level):  
  
    # Paso 3  
    if n_pregunta <= p_level:  
        level = 'básicas'
```

```
elif n_pregunta <= 2*p_level:
    level = 'intermedias'
else:
    level = 'avanzadas'

return level

if __name__ == '__main__':
    # verificar resultados
    print(choose_level(2, 2)) # básicas
    print(choose_level(3, 2)) # intermedias
    print(choose_level(7, 2)) # avanzadas
    print(choose_level(4, 3)) # intermedias
```

Requerimiento 3

- **Paso 4:** En el módulo `shuffle.py` se debe tomar de las preguntas sólo la clave correspondiente a alternativas y estas deben ser mezcladas.

```
import preguntas as p
import random

def shuffle_alt(pregunta):
    # Paso 4
    random.shuffle(pregunta['alternativas'])
    return pregunta['alternativas']

if __name__ == '__main__':
    # si se ejecuta el programa varias veces las alternativas
    # debieran aparecer en distinto orden
    print(shuffle_alt(p.pool_preguntas['basicas']['pregunta_1']))
    # a modo de ejemplo
    # [['alt_1', 0], ['alt_3', 0], ['alt_2', 1], ['alt_4', 0]]
```

Requerimiento 4

- **Paso 5:** En el módulo `question.py` utilizando la dificultad se debe llamar al pool de preguntas de dicha dificultad.
- **Paso 6:** Se debe escoger alguna de las opciones disponibles. Además luego de escoger la pregunta se debe eliminar la opción del diccionario de opciones global para que en un siguiente llamado no se pueda escoger esa opción nuevamente.
- **Paso 7:** Se debe desacoplar la pregunta en enunciado según la opción escogida y su respectivas alternativas mezcladas con la función `shuffle_alt()`.

```
import preguntas as p
import random
from shuffle import shuffle_alt

# Opciones dadas para escoger.
#####
opciones = {'basicas': [1,2,3],
            'intermedias': [1,2,3],
            'avanzadas': [1,2,3]}
#####

def choose_q(dificultad):
    # Paso 5
    preguntas = p.pool_preguntas[dificultad]

    # Paso 6
    global opciones
    # escoger una pregunta
    n_elegido = random.choice(opciones[dificultad])
    # eliminarla del ambiente global para no escogerla de nuevo
    opciones[dificultad].remove(n_elegido)

    # Paso 7
    pregunta = preguntas[f'pregunta_{n_elegido}']
    alternativas = shuffle_alt(pregunta)
    return pregunta['enunciado'], alternativas

if __name__ == '__main__':
    # si ejecuto el programa, las preguntas cambian de orden,
    # pero nunca debieran repetirse
    pregunta, alternativas = choose_q('basicas')
```

```
print(f'El enunciado es: {pregunta}')
print(f'Las alternativas son: {alternativas}')
pregunta, alternativas = choose_q('basicas')
print(f'El enunciado es: {pregunta}')
print(f'Las alternativas son: {alternativas}')

pregunta, alternativas = choose_q('basicas')
print(f'El enunciado es: {pregunta}')
print(f'Las alternativas son: {alternativas}')
```

Requerimiento 5

- **Paso 8:** En el módulo `print_preguntas.py` se debe imprimir las alternativas anteponiendo las letras de la A a la D según corresponda en cada alternativa.

```
import preguntas as p

def print_pregunta(enunciado, alternativas):

    # Paso 8
    print(f'{enunciado[0]}\n')
    letras = ['A', 'B', 'C', 'D']
    for l,a in zip(letras, alternativas):
        print(f'{l}. {a[0]}')

if __name__ == '__main__':
    # Las preguntas y alternativas deben mostrarse según lo indicado
    pregunta = p.pool_preguntas['basicas']['pregunta_1']
    print_pregunta(pregunta['enunciado'],pregunta['alternativas'])

    # Enunciado básico 1

    # A. alt_1
    # B. alt_2
    # C. alt_3
    # D. alt_4
```

Requerimiento 6

- **Paso 9:** En el módulo `verify.py` se debe generar las condiciones para identificar la alternativa correcta. Para ello hay que acceder al segundo elemento de cada Lista de Alternativas. En caso de que este sea 1, se imprime el mensaje correspondiente y se retorna un `True`, en caso de ser 0 se imprime el mensaje correspondiente y se retorna `False`.

```
import preguntas as p

def verificar(alternativas, eleccion):
    # devuelve el índice de elección dada
    eleccion = ['a', 'b', 'c', 'd'].index(eleccion)

    # Paso 9
    if alternativas[eleccion][1] == 1:
        print('Respuesta Correcta')
        correcto = True
    else:
        print('Respuesta Incorrecta')
        correcto = False

    return correcto

if __name__ == '__main__':
    from print_preguntas import print_pregunta

    # Siempre que se escoja la alternativa con alt_2 estará correcta,
    # e incorrecta en cualquier otro caso
    pregunta = p.pool_preguntas['basicas']['pregunta_2']
    print_pregunta(pregunta['enunciado'], pregunta['alternativas'])
    respuesta = input('Escoja la alternativa correcta:\n> ').lower()
    verificar(pregunta['alternativas'], respuesta)
```

Requerimiento 7

- **Paso 10:** Se implementa el validador para `opcion`.
- **Paso 11:** Se agrega el mensaje correspondiente para la opción 0 del menú.
- **Paso 12:** Se implementa un validador para el número de preguntas por nivel. Notar que es necesario convertir la salida en un `int`.
- **Paso 13:** Escoger el nivel de la pregunta dependiendo de qué pregunta se quiere hacer.
- **Paso 14:** Se implementa la elección de las preguntas dependiendo del nivel.
- **Paso 15:** Se imprime en pantalla con el formato correspondiente el enunciado y las alternativas elegidas.
- **Paso 16:** Se debe validar que las alternativas que se ingresan son entre la A y la D.
- **Paso 17:** Se implementa la verificación de respuesta correcta.
- **Paso 18:** Validar si los valores dados para continuar o no son `'y'` o `'n'`.

```
# No modificar
from verify import verificar
import preguntas as p
from question import choose_q
from print_preguntas import print_pregunta
from level import choose_level
from validador import validate
import time
import os
import sys

# valores iniciales - no modificar
n_pregunta = 0
continuar = 'y'
correcto = True
p_level = 10
op_sys = 'cls' if sys.platform == 'win32' else 'clear'

# -----

os.system(op_sys) # limpiar pantalla

print('Bienvenido a la Trivia')
opcion = input('Ingrese una opción para Jugar!
1. Jugar
0. Salir
```

```
> ''')
# Paso 10
opcion = validate(['0','1'], opcion)

# 2. Definir el comportamiento de Salir
if opcion == '0':
    print('Nos vemos pronto!') # Paso 11
    time.sleep(2)
    os.system(op_sys)
    exit()

# Funcionamiento de preguntas
while correcto and n_pregunta < 3*p_level:

    if n_pregunta == 0:
        p_level = input('¿Cuántas preguntas por nivel? (Máximo 3): ')
        # Paso 12
        p_level = int(validate(['1','2','3'],p_level))

    if continuar == 'y':
        #contador de preguntas
        n_pregunta += 1
        # Paso 13
        nivel = choose_level(n_pregunta, p_level)
        print(f'Pregunta {n_pregunta}:')
        # Paso 14
        enunciado, alternativas = choose_q(nivel)
        # Paso 15
        print_pregunta(enunciado, alternativas)

        respuesta = input('Escoja la alternativa correcta:\n> ').lower()
        # Paso 16
        respuesta = validate(['a','b','c','d'], respuesta)
        # Paso 17
        correcto = verificar(alternativas, respuesta)

    if correcto and n_pregunta < 3*p_level:
        print('Muy bien sigue así!')
        continuar = input('Desea continuar? [y/n]: ').lower()
        # Paso 18
        continuar = validate(['y','n'],continuar)
        os.system(op_sys)
    elif correcto and n_pregunta == 3*p_level:
```

```
        print(f'Felicitaciones, Has respondido {3*p_level} preguntas  
correctas. \n Has ganado la Trivia \n Gracias por Jugar, hasta luego!!!')  
        time.sleep(3)  
        os.system(op_sys)  
    else:  
        print(f'Lo siento, conseguiste {n_pregunta - 1} respuestas  
correctas,\n Sigue participando!!!')  
        time.sleep(3)  
        exit()  
else:  
    print('Nos vemos la proxima vez, sigue practicando')  
    time.sleep(3)  
    exit()
```