# Solución Prueba - Fundamentos de programación en Python

**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) |
| --- |

**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 = 'basicas'   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** |
| --- |

**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]] |
| --- |

**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}') |
| --- |

**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** |
| --- |

**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)** |
| --- |

**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() |
| --- |