**Reporte: Quiz Interactivo de Números en Python**

**Fecha:** 27 de agosto de 2025  
**Estudiante:** Cecilia Silva Heredia

**1. Propósito del proyecto**

El objetivo de este proyecto es desarrollar un **quiz interactivo de números** en Python que permita al usuario ingresar un número y responder preguntas sobre sus propiedades matemáticas.  
El propósito principal es **aplicar los conocimientos de programación adquiridos en el bootcamp**, incluyendo funciones, estructuras de control, listas, diccionarios, ciclos y manejo de entrada/salida.

**2. Lo aprendido**

Durante la realización del proyecto se fortalecieron las siguientes habilidades:

* Creación de **funciones** con retorno de datos (trivia\_fetch).
* Uso de **diccionarios y listas** para organizar información y preguntas.
* Implementación de **condicionales (if) y bucles (for)**.
* Interacción con el usuario mediante **input y print**.
* Aplicación de **aleatoriedad** con random.shuffle para mezclar opciones.
* Separación de la **lógica de cálculo** de la **lógica de presentación** para un código más limpio y modular.

**4. Pasos para el desarrollo del quiz**

**Para construir este proyecto, se siguió un proceso paso a paso, combinando planificación y desarrollo en Python:**

**Paso 1: Definir el objetivo y las propiedades del número**

* **Se identificaron las propiedades matemáticas del número que se usarían en las preguntas: paridad, signo, primalidad, divisibilidad, cuadrado, múltiplos y suma de dígitos.**

**Paso 2: Crear la función principal trivia\_fetch(num)**

* **Esta función recibe un número y calcula todas las propiedades.**
* **Se almacenan en un diccionario para acceder fácilmente a cada propiedad durante el quiz.**

**Paso 3: Diseñar la estructura de preguntas y opciones**

* **Se creó una lista de preguntas que usara los datos del diccionario.**
* **Cada pregunta contiene 4 opciones, incluyendo la correcta.**

**Paso 4: Implementar aleatoriedad con random.shuffle()**

* **Las opciones se mezclan aleatoriamente para que el quiz no sea predecible.**

**Paso 5: Ciclo de preguntas con validación de respuestas**

* **Se recorre cada pregunta usando un bucle for.**
* **Se pide al usuario que ingrese la letra de su respuesta.**
* **Se valida la respuesta y se actualiza el puntaje si es correcta.**

**Paso 6: Mostrar retroalimentación inmediata**

* **Después de cada respuesta, el programa indica si es correcta o incorrecta.**

**Paso 7: Calcular y mostrar puntaje final**

* **Al finalizar todas las preguntas, se muestra el puntaje total al usuario para evaluar su desempeño.**

**Paso 8: Separar lógica de cálculo y presentación**

* **Toda la parte de cálculo se mantiene en trivia\_fetch.**
* **La interacción con el usuario se realiza en el ciclo de preguntas, garantizando un código modular y limpio.**

**3. Funcionamiento del quiz**

**3.1 Motor del quiz: trivia\_fetch(num)**

Esta función recibe un número entero y devuelve un diccionario con toda la información necesaria para el quiz. Entre las propiedades calculadas se incluyen:

* **Paridad:** par o impar.
* **Signo:** positivo, negativo o cero.
* **Primalidad:** si el número es primo.
* **Cuadrado:** resultado de elevarlo al cuadrado.
* **Divisibilidad:** entre 3, 5 y 10.
* **Mayor que 100:** booleano.
* **Múltiplo de 7:** booleano.
* **Suma de dígitos y su paridad:** par o impar.

**Ejemplo de salida de la función para el número 42:**

{

"number": 42,

"paridad": "par",

"signo": "positivo",

"es\_primo": False,

"cuadrado": 1764,

"divisible\_por\_3": True,

"divisible\_por\_5": False,

"divisible\_por\_10": False,

"mayor\_que\_100": False,

"multiplo\_de\_7": True,

"suma\_digitos\_paridad": "par"

}

**3.2 Ciclo de preguntas**

1. El programa recorre cada pregunta en un ciclo for.
2. Cada pregunta tiene **4 opciones** (a, b, c, d), que se mezclan aleatoriamente usando random.shuffle para que el orden cambie cada vez que se ejecuta el quiz.
3. El usuario ingresa la letra de su respuesta.
4. El programa compara la letra ingresada con la opción correcta y muestra si es correcta o incorrecta.
5. Se suma un punto al puntaje total si la respuesta es correcta.

**Ejemplo de interacción:**

Pregunta 1: ¿Tu número pertenece al equipo de los pares o al de los impares?

a) impar

b) par

c) cero

d) ninguno

Tu respuesta (a/b/c/d): b

✅ ¡Correcto!

**3.3 Puntaje final**

Al finalizar todas las preguntas, se muestra el puntaje obtenido por el usuario, por ejemplo:

🎉 Has terminado el quiz. Tu puntaje final es 8/10

Esto permite evaluar de manera inmediata el desempeño del usuario y la comprensión de las propiedades del número ingresado.

**4. Tecnologías y conceptos aplicados**

* **Python** como lenguaje de programación.
* **Funciones** y modularidad (trivia\_fetch).
* **Listas y diccionarios** para organizar preguntas y respuestas.
* **Condicionales y bucles** para iterar y validar respuestas.
* **Entrada y salida** con input y print.
* **Aleatoriedad** con random.shuffle para mezclar opciones.

**5. Cierre**

Con este proyecto, se consolidó la capacidad de crear un **programa interactivo, modular y dinámico** en Python.  
Se logró combinar la lógica matemática con la interacción con el usuario y la validación automática de respuestas, reforzando habilidades de programación adquiridas durante el bootcamp.

Este proyecto también evidencia la importancia de **separar la lógica de cálculo de la presentación al usuario**, así como la utilidad de **listas, diccionarios y funciones** para desarrollar aplicaciones escalables y mantenibles.

EVIDENCIA :

  




