

Preguntas de Análisis Responda a las siguientes preguntas y crea un archivo de PDF con un análisis detallado, justificando tus respuestas con ejemplos de tu código en un Pdf

1. ¿Qué funciones de cadena de texto de Python utilizaste para limpiar los datos del JSON? Describe para cada una por qué fue la elección correcta.
En la clase Mision utilicé varias funciones para limpiar la frase:
 - `isalnum()`: Permite verificar si un carácter es alfanumérico, útil para eliminar símbolos y mantener letras y números.
 - `isspace()`: Verifica si un carácter es espacio, para mantener los espacios necesarios entre palabras.
 - `join()`: Une los caracteres filtrados en una nueva cadena limpia.
 - `split()` / `join()`: Para normalizar los espacios entre palabras, evitando dobles espacios o saltos innecesarios.
 - `title()`: Capitaliza la primera letra de cada palabra, mejorando la legibilidad del nombre y frase.
2. ¿Por qué es importante el uso de un bloque try-except con `ValueError` en este examen? Proporciona un escenario donde tu código podría fallar si no se implementa el manejo de errores.
Porque ayuda a que se hagan limpiezas de datos o consultas en la base de datos evitando que el programa se detenga
3. ¿Qué es una llave primaria (pk) en una base de datos? ¿Por qué fue crucial en este examen para encontrar tu registro?
Digamos que es como una etiqueta de identifica cada registro de la tabla en el examen nos ayudo a identificar cada estudiante
4. ¿Qué ventajas ofrece el uso de clases y objetos para organizar la solución de este acertijo, en comparación con el uso de funciones simples y variables globales?
 - Organiza mejor el código y evita variables globales.
 - Permite reutilizar métodos en diferentes partes del programa.
 - Facilita la ampliación del proyecto y el mantenimiento.
 - Encapsula datos privados y protege la información de modificaciones accidentales.
5. Describe el flujo de información que seguiste, desde que recibiste el archivo JSON hasta que obtuviste el mensaje final corregido.
 - Se recibe el archivo JSON con los estudiantes.

- `LectorJSON` lee y filtra el estudiante solicitado.
 - Se calcula la llave primaria con $f(n) = (\text{consecutivo} \% 42) + 1$.
 - Se consulta la base de datos `examen2` con la PK obtenida.
 - La frase recuperada se pasa a la clase `Mision` para limpiar y normalizar.
 - Finalmente, se obtiene la frase corregida y se muestran las funciones de string utilizadas
6. ¿Por qué la separación del código en diferentes clases (por ejemplo, `LectorJSON` y `Mision`) mejora la legibilidad y el mantenimiento del programa?
Cada clase tiene su responsabilidad
Permite detectar errores mas fácilmente y cambiar partes del programa sin afectar otras
7. Si la lista de estudiantes tuviera 1000 registros, ¿crees que tu método de filtrado seguiría siendo eficiente? ¿Qué podrías hacer para mejorarlo?
- Para mejorar, se podría usar un diccionario con el nombre del estudiante como clave para acceso directo ($O(1)$)
8. ¿Qué estrategia o herramientas utilizaste para depurar tu código y asegurarte de que cada paso funcionara correctamente?
- Uso de `print()` para verificar resultados intermedios.
 - Pruebas unitarias por pasos: primero la lectura JSON, luego filtrado, luego limpieza de frase.
 - Manejo de excepciones con `try-except` para ver mensajes de error claros.
9. ¿Qué aprendiste sobre la importancia de la validación de datos al trabajar con información que contiene errores intencionales?
Es muy importante trabajar ordenado y no tratar de desordenar todo el programa ya que eso lleva a tener muchos errores
10. Que comando utilizaste en SQL para filtrar los dato
- Se utiliza un parámetro (?) con la PK calculada para obtener el registro correcto.
- Esto protege contra inyecciones SQL y asegura que solo se obtenga el registro deseado.

