

UNIVERSIDAD AMERICANA



Algoritmos y Estructuras de Datos

Evaluación Práctica Estructuras de Datos

Autores:

Fernando Reyes Avilés
Rubén Valle Montenegro
Mario Grillo

Docente: Lic. Silvia Gigdalia Ticay Lopez

1. Introducción

La presente actividad busca aplicar los conceptos de estructuras de datos lineales en Python, específicamente el uso de **colas (FIFO)** y **pilas (LIFO)**, mediante un ejercicio práctico de separación de elementos de una lista en función de sus posiciones.

2. Objetivo

Desarrollar un programa en Python que, dado una cola inicial con elementos y una pila vacía, procese los elementos de la cola manteniendo los que se encuentran en posiciones pares y transfiriendo a la pila aquellos que están en posiciones impares, cumpliendo con la restricción de recorrer la cola una sola vez.

3. Descripción del problema

Se parte de una cola con varios elementos. La tarea consiste en analizar cada elemento según su posición:

- Si está en una **posición par** (0, 2, 4...), se mantiene en la cola.
- Si está en una **posición impar** (1, 3, 5...), se transfiere a una pila.

Esto debe realizarse recorriendo la cola una única vez, respetando el orden FIFO de la cola y el orden LIFO de la pila.

4. Estructura del proyecto

```
estructura_datos/  
├── main.py  
├── cola.py  
├── pila.py  
└── procesador.py
```

5. Explicación del código

cola.py

Contiene la clase Cola, que implementa una estructura FIFO:

- `encolar(item)`: agrega elementos al final.
- `desencolar()`: retira el primer elemento.
- `esta_vacia()`: verifica si está vacía.

pila.py

Contiene la clase Pila, que implementa una estructura LIFO:

- `apilar(item)`: agrega elementos al tope.
- `desapilar()`: remueve el último elemento agregado.

procesador.py

Función `procesarCola` que recorre una sola vez la cola, utiliza el índice para decidir si el elemento se mantiene o se apila, y devuelve una nueva cola con los pares.

main.py

Es el punto de entrada del programa. Crea la cola, la pila y llama a la función procesadora. Luego imprime los resultados.

6. Ejecución del programa

Entrada:

Cola original: [A, B, C, D, E]

Salida esperada:

Cola final: ['A', 'C', 'E']

Pila final: ['B', 'D']

7. Conclusiones

Este ejercicio permitió aplicar la lógica de estructuras de datos lineales mediante la programación orientada a objetos en Python. Se reforzaron los conceptos de manipulación de colas y pilas, manteniendo eficiencia y claridad en el código, y se respetaron las restricciones planteadas en el problema.