Actividad evaluativa - Eje 1 - Pilas y Colas

Melqui Alexander Romero Veru

Fundación Universitaria del Área Andina

Ingeniería en Sistemas, MODELOS DE PROGRAMACIÓN II

Deivys Morales

Soacha Cundinamarca

16 de febrero del 2025

Actividad evaluativa - Eje 1 - Pilas y Colas

Cuando trabajamos con datos, es clave organizarlos bien para que sean fáciles de manejar y acceder. Para esto, las estructuras de datos como Pilas y Colas nos ayudan a ordenar la información de distintas maneras según nuestras necesidades. Aunque ambas son listas enlazadas, la diferencia en ordenamiento entre Pila y Cola es fundamental: mientras que en una Pila el último dato en entrar es el primero en salir, en una Cola los datos se procesan en el mismo orden en que llegaron.

En este documento, exploraremos estas estructuras a través de un ejemplo práctico, donde crearemos un programa en Python para almacenar y gestionar datos en Pilas y Colas. Primero, trabajaremos con números insertados desde la terminal luego mostraremos y analizamos la información clave como la cantidad de pares, el promedio y el último dato agregado. Luego, aplicaremos lo aprendido en una lista más estructurada, donde almacenaremos información de personas, la mostraremos, eliminaremos el primer elemento y contaremos los datos restantes.

Objetivo General

Explicar la construcción e implementación de estructuras de datos Pila y Cola en el lenguaje de programación Python, con el propósito de comprender su funcionamiento y aplicabilidad en el almacenamiento y gestión de datos a través de un ejemplo práctico que muestra su funcionamiento y aplicación. Además, se busca proporcionar una guía clara sobre el desarrollo de la actividad solicitada, así como incluir conclusiones y referencias confiables.

Objetivos Específicos

1. Implementar una lista enlazada tipo Pila que permita la inserción de N elementos y realice operaciones como mostrar los datos, contar los números pares, calcular el promedio y obtener el último elemento.
2. Implementar una lista enlazada tipo Cola para almacenar información de personas, permitiendo mostrar los datos, eliminar el primer elemento y contar los elementos restantes.
3. Analizar el comportamiento de las estructuras de datos tipo Pila y Cola en la organización de información.

Tabla de Contenidos

Capítulo 1

Definición de Pilas y Colas

Pilas

Una pila es una estructura de datos que sigue el principio LIFO (Last In, First Out), es decir, el último elemento en entrar es el primero en salir. Las operaciones básicas de una pila son:

Agregar: Añade un elemento al inicio de la pila.

Eliminar: Elimina el primer elemento de la pila.

Mostrar lista: Muestra todos los elementos de la pila.

Contar elementos: Devuelve el número total de elementos en la pila.

En el proyecto, la clase `pila` implementa estas operaciones básicas. Aquí hay un ejemplo de cómo se define una pila en el código:

```python
class pila:
    def __init__(self):
        self._pila=[]

    def agregar(self):
        while True:
            elemento = int(input("Numero a agregar a la pila: "))
            self._pila.insert(0, elemento)

            continuar = input("¿Quieres agregar otro número? (s / N o otra letra): ")
            if continuar.lower() != 's':
                break

    def mostrarlista(self):
        return (f"Lista: {self._pila}")

    def promedioLista(self):
        return (f"Promedio: {round(sum(self._pila) / len(self._pila), 2)}")

    def ultimoDato(self):
        return (f"Ultimo elemento: {(self._pila[-1])}")

    def canditadpares(self):
        canPares = 0
        for n in self._pila:
            if n % 2 == 0:
                canPares += 1
        return (f"Total pares: {canPares}")

```

Figura 1. Declaración y funciones - Pila

Colas

Una cola es una estructura de datos que sigue el principio FIFO (First In, First Out), es decir, el primer elemento en entrar es el primero en salir. Las operaciones básicas de una cola son:

Agregar: Añade un elemento al final de la cola.

Eliminar: Elimina el primer elemento de la cola.

Mostrar lista: Muestra todos los elementos de la cola.

Contar elementos: Devuelve el número total de elementos en la cola.

En el proyecto, la clase `cola` implementa estas operaciones básicas. Aquí hay un ejemplo de cómo se define una cola en el código:

```python
class cola:
    def __init__(self):
        self._cola = []

    def agregar(self):
        while True:
            elemento = int(input("Numero a agregar a la cola: "))
            self._cola.append(elemento)

            continuar = input("¿Quieres agregar otro número? (s / N o otra letra): ")
            if continuar.lower() != 's':
                break

    def mostrarlista(self):
        return (f"Lista: {self._cola}")

    def promedioLista(self):
        return (f"Promedio: {round(sum(self._cola) / len(self._cola), 2)}")

    def ultimoDato(self):
        return (f"Ultimo elemento: {self._cola[-1]}")

    def canditadpares(self):
        canPares = 0
        for n in self._cola:
            if n % 2 == 0:
                canPares += 1
        return (f"Total pares: {canPares}")

```

Figura 2. Declaración y funciones – Cola

Capítulo 2

Estructura general del código

Descripción

El proyecto se divide en dos ejercicios principales, más una aplicación principal.

1. Ejercicio 1:
   - Clases: pila y cola
   - Funciones principales:
     - Agregar números
     - Mostrar lista de elementos
     - Calcular el promedio de los elementos
     - Contar la cantidad de números pares
     - Obtener el último dato de la lista
   - Clase: APP_1 (menú interactivo para las operaciones anteriores)

2. Ejercicio 2:
   - Clases: pila y cola
   - Funciones principales:
     - Agregar información de personas (código, nombre, teléfono y edad)
     - Mostrar la lista de personas
     - Eliminar el primer dato de la lista

- Contar la cantidad de elementos

- Clase: APP_2 (menú interactivo para las operaciones anteriores)

3. Aplicación principal (APP):

- Permite al usuario seleccionar:

  - Menú de Ejercicio 1

  - Menú de Ejercicio 2

  - Cerrar la aplicación

- Según la opción elegida, se instancia el menú (APP_1 o APP_2).

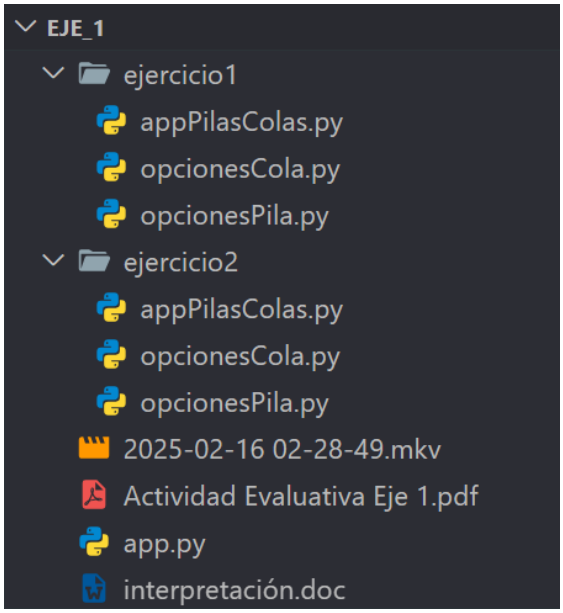

Figura 3. Estructura del proyecto

Capítulo 3

Ejecución del programa

Como ejecutarlo

Para le ejecución del programa debemos ejecutar el código en el archivo *app.py* este una vez ejecutado iniciará la App.

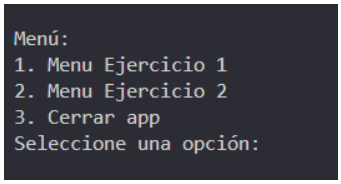

Figura 4. Menú inicial

Posibles salidas

Las salidas que puede generarse al ejecutar el flujo son muy diversas ya que la dinámica es algo amplia, a continuación, se mostrara un resultado de tipo pila.

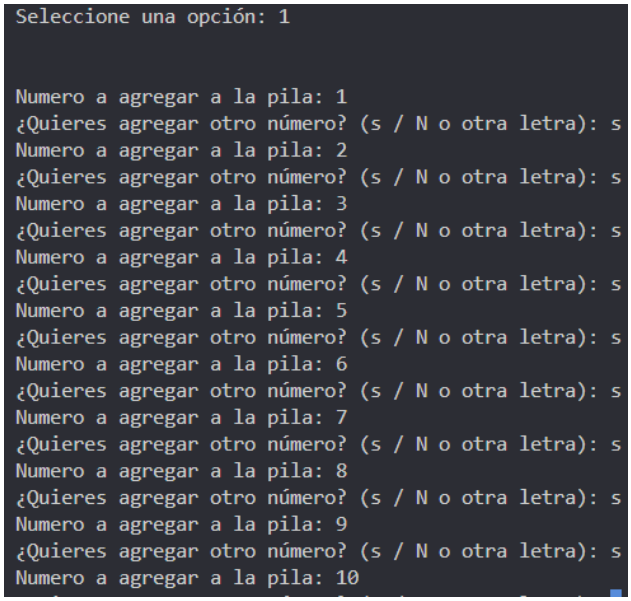

Figura 5. Dados insertados

Se muestra el resultado sobre los datos insertados:

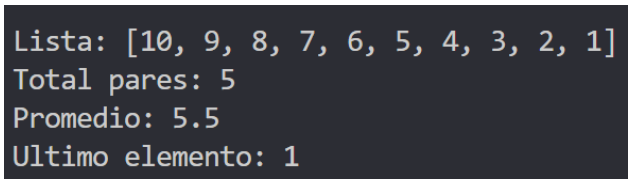

Figura 6. Salida de datos en una lista de tipo Pila

Se muestra el resultado sobre los datos insertados:

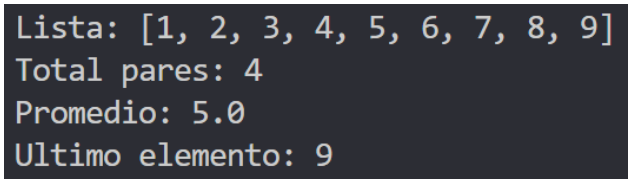

Figura 7. Salida de datos en una lista de tipo Cola

Video explicativo

Nota: Para la comprensión del programa se ha realizado un video explicativo, donde se muestra desde la estructura del programa hasta la ejecución del mismo.
url (https://youtu.be/8OKKSQkh1aI)

Conclusiones

Comprensión y Aplicabilidad de Pilas y Colas

La implementación de Pilas y Colas en Python permite entender la diferencia en ordenamiento entre ambas estructuras y su impacto en la organización de datos. Gracias a esta práctica, se evidencia cómo cada estructura es útil dependiendo del tipo de problema a resolver.

Eficiencia en la Gestión de Datos

Utilizar listas enlazadas para manejar Pilas y Colas demuestra ser un método eficiente y flexible. Al definir correctamente las funciones para insertar, eliminar y procesar datos, se facilita el acceso y manipulación de la información sin complicaciones.

Interacción a través de una Interfaz de Línea de Comandos

La integración de un menú interactivo mejora la experiencia de usuario al organizar las opciones disponibles de manera clara y estructurada. Este enfoque refuerza la importancia de la usabilidad en aplicaciones de programación.

Aplicación en el Mundo Real

Más allá del ejercicio académico, estas estructuras se utilizan en el desarrollo de software para la optimización de recursos y el procesamiento de datos en tiempo real. Dominar su uso abre la puerta a resolver problemas más complejos de forma estructurada y eficiente.

Lista de referencias

Python Software Foundation. (s.f.). Data Structures. Recuperado de: https://docs.python.org/3/tutorial/datastructures.html

Runestone Academy. (s.f.). Implementación de una pila en Python. Recuperado de: https://runestone.academy/ns/books/published/pythoned/BasicDS/ImplementacionDeUnaPilaEnPython.html

Runestone Academy. (s.f.). Implementación de una cola en Python. Recuperado de: https://runestone.academy/ns/books/published/pythoned/BasicDS/ImplementacionDeUnaColaEnPython.html

GeeksforGeeks. (s.f.). Stack Data Structure. Recuperado de: https://www.geeksforgeeks.org/stack-data-structure/

GeeksforGeeks. (s.f.). Queue Data Structure. Recuperado de: https://www.geeksforgeeks.org/queue-data-structure/

UC3M OpenCourseWare. (s.f.). Tema 2 Tipos Abstractos de Datos Lineales: 2.1. Pilas y Colas [PDF]. Recuperado de: https://ocw.uc3m.es/pluginfile.php/5742/mod_page/content/46/tema2-1.pdf

Normas APA. (s.f.). Normas APA. Recuperado de: https://normasapa.in/

Video explicativo de la Actividad url: https://youtu.be/8OKKSQkh1aI