REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DEL PODER POPULAR
PARA LA EDUCACIÓN UNIVERSITARIA
UPTT “MBI” NÚCLEO “BARBARITA DE LA TORRE”
PNF EN INFORMATICA

3ra Acitvidad de Programación
“Manejos de Archivos, Pilas, Colas en Python y Ejemplos Práctico”

Realizado por:

Jonathan Rodríguez	C.I. 31176588
José Rangel	C.I. 31520449
Roger Márquez	C.I. 31007828

Profesor: Carlos Bravo

Unidad Curricular: Programación

Trujillo, Nov de 2024

Manejo de Archivos en Python

Se refiere a la forma en que puedes abrir, leer, escribir y cerrar archivos en tu sistema. Los conceptos básicos son los siguientes:

1. **Abrir un archivo**: Usas la función `open()`, que acepta el nombre del archivo y el modo de apertura (por ejemplo, `'r'` para leer, `'w'` para escribir, `'a'` para agregar).

2. Leer un archivo: Puedes usar métodos como `.read()`, `.readline()` o `.readlines()` para obtener el contenido del archivo.

3. Escribir en un archivo: Al abrir un archivo en modo de escritura (`'w'`), puedes usar el método `.write()` para agregar contenido.

4. Cerrar un archivo: Es importante cerrar el archivo después de usarlo con el método `.close()`, así liberas recursos del sistema.

5. Manejo de excepciones: Es recomendable usar bloques `try-except` para manejar posibles errores, como que el archivo no exista.

6. Context Manager: Usar `with open(...) as ...:` es una buena práctica, ya que asegura que el archivo se cierre automáticamente al salir del bloque.

Ejemplo Práctico

```
# Escribir en un archivo
with open('mi_archivo.txt', 'w') as archivo:
    archivo.write('Hola, mundo!\n')

# Leer el archivo
with open('mi_archivo.txt', 'r') as archivo:
    contenido = archivo.read()
    print(contenido)
```

Pilas

En Python, una pila es una estructura de datos que sigue el principio LIFO (Last In, First Out), lo que significa que el último elemento agregado es el primero en ser removido. Las pilas son útiles en diversas aplicaciones, como la gestión de llamadas de funciones, la evaluación de expresiones y el retroceso en algoritmos.

Ejemplo Práctico

```
class Pila:
    def __init__(self):
        self.items = []

    def is_empty(self):
        return len(self.items) == 0

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        return None

    def peek(self):
        if not self.is_empty():
            return self.items[-1]
        return None
```

Colas en Python

En Python, una cola es una estructura de datos que sigue el principio FIFO (First In, First Out), lo que significa que el primer elemento agregado es el primero en ser removido. Las colas son muy útiles en situaciones donde se necesita procesar

elementos en el orden en que llegaron, como en la gestión de tareas, servicios de impresión o en algoritmos de búsqueda.

Ejemplo Práctico

```
# Crear una cola
cola = deque()

# Añadir elementos
cola.append('a')
cola.append('b')
cola.append('c')

# Retirar elementos
primer_elemento = cola.popleft()  # 'a'
segundo_elemento = cola.popleft()  # 'b'

print(primer_elemento)  # Salida: a
print(segundo_elemento)  # Salida: b
print(cola)  # Salida: deque(['c'])
```