

Python



Archivos de Texto

Algoritmos y Programación I
Lic. Gustavo Bianchi

1

Archivos de Texto

DEFINICION

Un archivo es una colección de información (datos relacionados entre sí), almacenados como una unidad en un dispositivo.

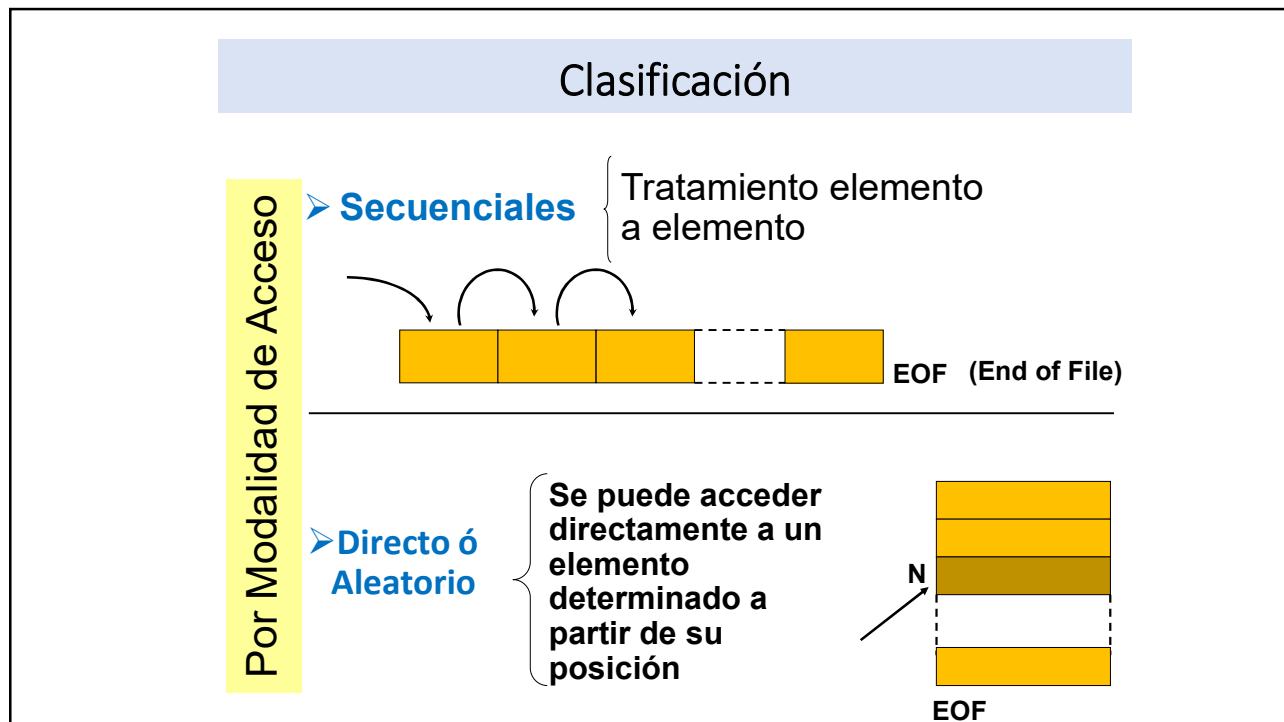
TAMAÑO

- No es Fijo
 - Limitado por el medio de almacenamiento
- } SON DINÁMICOS

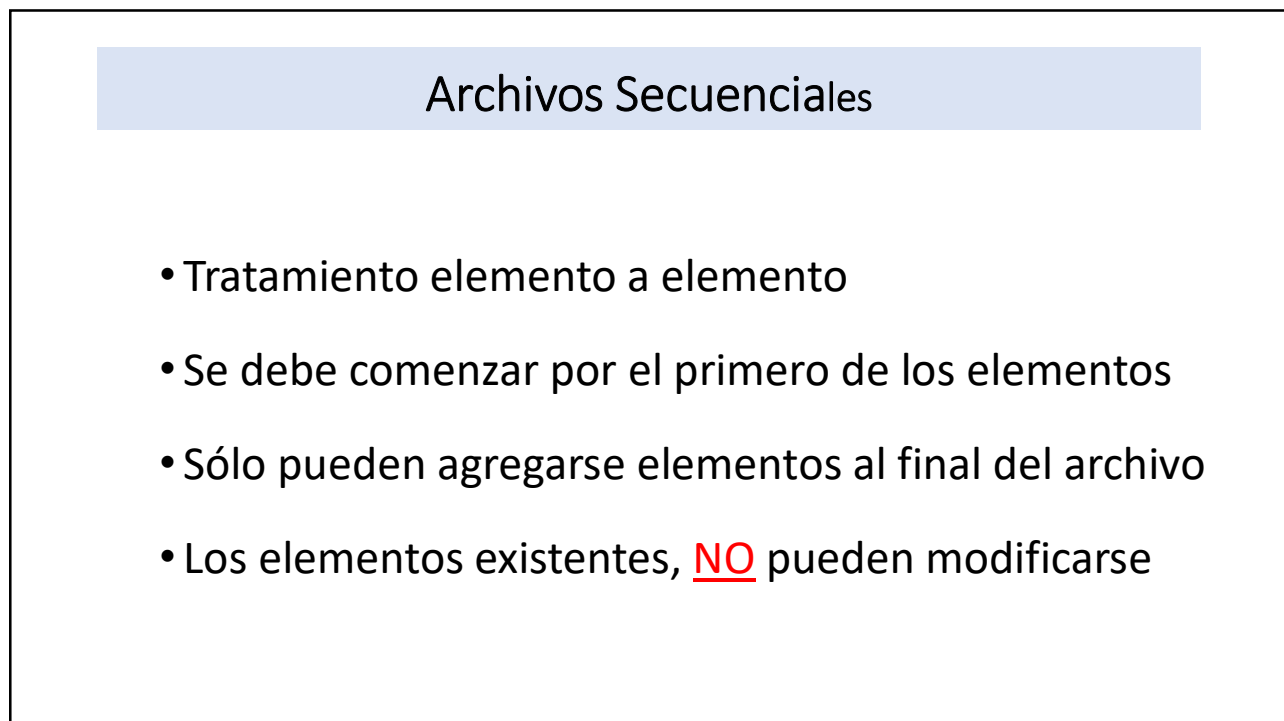
TIPOS

- Delimitados por coma → .csv (comma separated values)
- Sin delimitadores → .txt
- Con otro tipo de delimitadores, tabulaciones, ó longitud fija

2



3



4

Archivos Aleatorios

- Acceso directo a un elemento a través de su posición
- Se pueden agregar elementos al final del archivo
- Los elementos existentes, pueden ser modificados

5

Archivos de Texto

LEER

1. Abrir (open)
2. Leer (read, readline, readlines)
3. Cerrar (close)

ESCRIBIR

1. Abrir (open)
2. Guardar (write)
3. Cerrar (close)

6

Archivos de Texto

```
open( filename [, mode][, ..])
```

↓
ruta + nombre de archivo

↓
r sólo lectura (por omisión)
r+ lectura y escritura
w sobreescritura
a agrega datos
b indica archivo binario

```
>> fh = open("c:\\datos\\clientes.csv", "r+")
```

NOTA: En Windows debemos usar \\ en lugar de una \ para indicar el camino al archivo

7

Archivos de Texto

LEER

```
.read( [tamaño])
```

- Si se omite la cantidad, lee todo.
- EOF: devuelve ""

```
.readline()
```

Lee la línea completa

```
.readlines()
```

Devuelve una lista de N elementos.
Genera tantos elementos como líneas en el archivo.

8

Archivos de Texto

ESCRIBIR

```
.write( "Cadena a escribir" )
```

Si quiero que al final de la línea se grave un fin de línea, tengo que agregar `\n`, de lo contrario, la cadena siguiente, se grabará a continuación de esta.

9

Archivos de Texto

```
.tell()
```

Devuelve un número entero, que indica la posición a la que se está apuntando en el archivo.

En el caso de archivos binarios, representa la cantidad de bytes desde el comienzo del archivo.

```
.seek(posicion [,desde_donde])
```

Permite cambiar la posición a la que se apunta en el archivo.

Sumará "posición", al parámetro "desde_donde". Este parámetro, puede tomar valor:

- 0 (por omisión, e indica desde el principio);
- 1 (desde donde se encuentra);
- 2 (desde el final del archivo)

Nota: para archivos de texto, el uso de seek tiene limitaciones.

10

Archivos de Texto

```
.close()
```

Produce el cierre del archivo,
liberando los recursos pertinentes.

ALTERNATIVA

```
with open(filename [, mode]) as variable:
```

```
.....  
.....
```

Al usar la sentencia with, nos evitamos el tener que cerrar el archivo con close, porque al finalizar el bloque que encierra el with, el archivo es cerrado