

# ENTRADA Y SALIDA

1. Entrada y Salida estándar
  - a. Datos No Temporales
  - b. Flujos de Datos
2. Estructura de Datos: Archivo, Registro y Campo
3. Lectura secuencial de Registros
4. Lectura aleatoria de Registros
5. Operaciones sobre Archivos
6. Archivos Indexados
7. Archivos en JAVA

## Definición

- En computación, entrada/salida, también abreviado E/S o I/O (del original en inglés input/output), es la colección de interfaces que usan los distintos subsistemas de un sistema de información para comunicarse unos con otros.
- Las entradas son las señales recibidas por la unidad, mientras que las salidas son las señales enviadas por ésta.
- Todo algoritmo debe obedecer a la estructura básica de un sistema, es decir: entrada, proceso y salida.
- Donde **ENTRADA** corresponde al insumo, a los datos necesarios que requiere el proceso para ofrecer los resultados esperados.
- **PROCESO** son los pasos necesarios para obtener la solución del problema o la situación planteada.
- **SALIDA** son los resultados arrojados por el proceso como solución.

## Definición

- Hasta el momento se presentaron diferentes *estructuras de datos*, las cuales tienen una particularidad en común, son internas y temporales, es decir, son definidas en un algoritmo y ocupan memoria **RAM**, se utilizan en ejecución y todos los valores contenidos en ellas se pierden, a lo sumo, cuando el algoritmo termina de ejecutarse.
- La memoria interna o **RAM** de una computadora es denominada **memoria principal** y sus principales características son:
  - Tiene capacidad limitada
  - Es memoria volátil ya que al apagar la computadora o al cortarse la luz, se pierde la información que en ella reside.
  - Es memoria electrónica, por lo cual su velocidad de acceso es muy elevada (memoria de acceso aleatorio)

## Definición

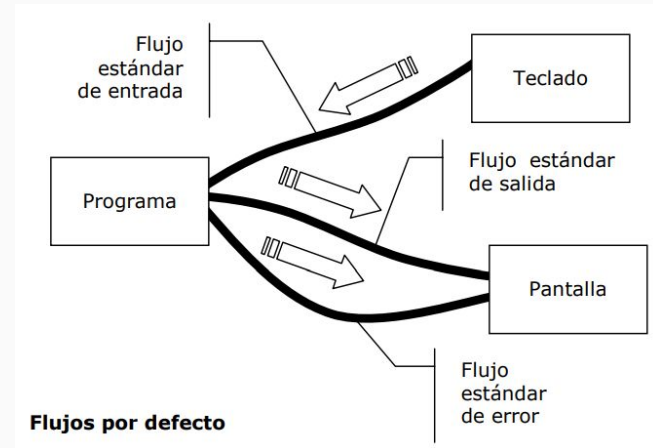
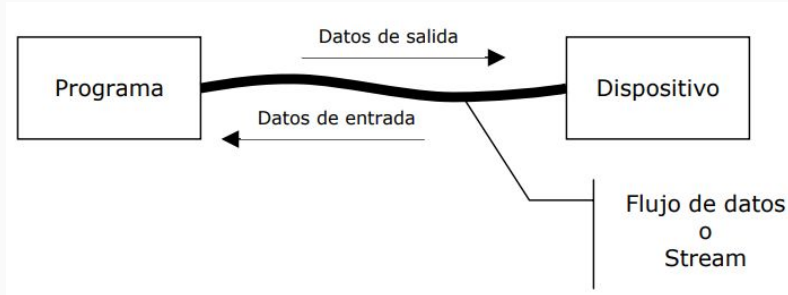
- Para determinados problemas es necesario disponer de un tipo de estructura de datos que permita guardar su información en soporte **no volátil** o temporal y de esa forma preservarla aunque el programa finalice.
- La **memoria secundaria**, auxiliar o externa se refiere a medios de almacenamiento que están ubicados fuera de la memoria RAM, y que son capaces de retener información luego de que un programa finaliza.
- Dichos dispositivos secundarios almacenan información en medios magnéticos (disco rígido, cintas, diskettes, etc.) u otros medios, aunque esto hace mucho más lenta su operación.
- La velocidad de acceso a la memoria afecta notablemente la eficiencia de un algoritmo.

## Definición

- Las estructuras de datos que se asocian con un dispositivo de memoria auxiliar permanente donde almacenan la información se denominan **archivos**.
- Se utilizan:
  - cuando la cantidad de datos que manipula el programa es muy grande y no hay suficiente espacio en la RAM.
  - cuando el resultado o la salida de un programa sirva de entrada para otro programa, por lo cual es necesario guardarlo en forma permanente.
  - cuando la carga sucesiva de datos se torna engorrosa, pesada y lleva demasiado tiempo, por lo cual conviene guardarla en forma permanente.

## Definición

- Los programas son capaces de obtener y mandar datos de y hacia cualquiera de estos dispositivos e incluso de ficheros. Para ello es necesario que se establezca una vía de comunicación entre el programa y el fichero o dispositivo. Esa vía, o canal de comunicación, se suele llamar flujo de datos o stream (la mayor parte de la literatura emplea el término en inglés).



## Tipos de Flujos

- Según la dirección de en la que viaja la información pueden ser:
  - **De entrada:** permiten leer información del fichero al que están conectados
  - **De salida:** permiten enviarle información
  - **De entrada/salida:** permiten hacer ambas operaciones
- Según el tipo de datos que circulen por ellos pueden ser:
  - **De texto:** Un flujo de texto es una secuencia de líneas; cada línea tiene cero o más caracteres y está terminada por una marca especial denominada “nueva línea”, que se representa con el carácter '\n'.
  - **Binarios:** Un flujo binario es una secuencia de bytes no procesados que representan datos internos. Los datos que se graben en formato binario deben ser interpretados en la lectura de forma adecuada, cae de parte del programa el saber interpretar la secuencia de bytes.

## Archivo

- Un archivo es una colección de datos del mismo tipo relacionados entre sí, almacenados en una unidad de memoria secundaria, con un determinado diseño o estructura.
- Los archivos también denominados ficheros (file); es una colección de información (datos relacionados entre sí), localizada o almacenada como una unidad en alguna parte de la computadora.

## Objetivos

1. Almacenamiento permanente.
2. Manipulación de un gran número de datos
3. Independencia de los programas.
4. Residentes en soportes externos.



# ESTRUCTURA DE DATOS: Archivos, Registros y Campos

## Estrutura

- Archivos => Colección de Registros
- Registros => Colección de Campos

## Partes

- **Registros:** Los registros están conformados por un conjunto de Campos, los cuales pueden ser comparados con Arreglos.
- **Campos:** Son los valores que se almacenan dentro de cada Registro y deben ser leído uno a uno.

DNI	Apellidos	Nombre	Nota Teoría	Nota Práctica
99.999.999	Fernández Alba	Manuel	8.5	5.7
88.888.888	Cid Rodríguez	Fernando	7.6	5.9
77.777.777	Álvarez Gutiérrez	Martín	6.8	4.2

## Operaciones sobre Archivos

El mecanismo de inserción y recuperación de la información debe seguir una "lógica" que el programa debe conocer y que estará en cierta medida condicionada por las características de los dispositivos.

- **Creación:** Definición del archivo.
- **Apertura:** Comunicación del archivo lógico con el físico.
- **Cierre:** Cerrar la conexión.
- **Lectura / Consulta:** Acceder al archivo para ver su contenido.
- **Fin de Archivo (EOF):** Detecta el final del archivo.
- **Destrucción:** Borra el Archivo.
- **Escritura:**
  - Alta: Adición de registros.
  - Baja: Eliminación o borrado lógico de registros.
  - Modificación: Altera la información del contenido.

## **Acceso a Archivos**

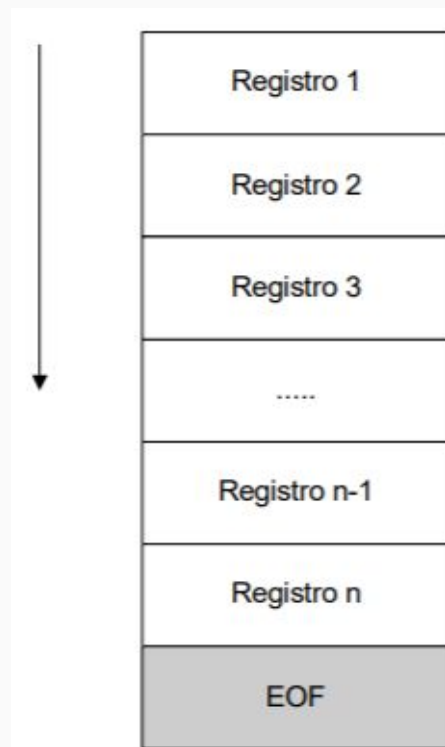
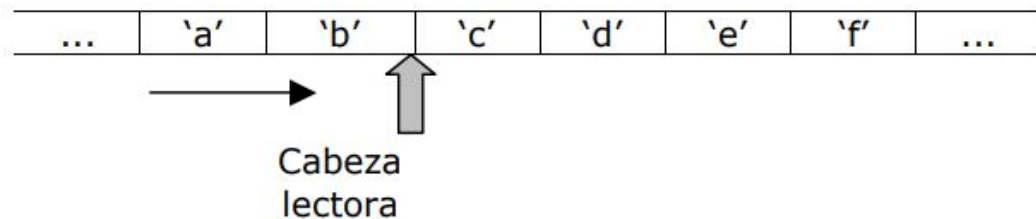
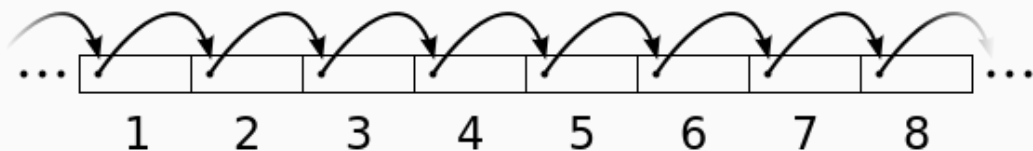
Según la forma de acceso a la información dentro del fichero estos pueden ser:

- Secuenciales
- Directos
- Indexados

## Organización o Acceso Secuencial

- Un archivo de Acceso Secuencial es aquel donde los registros están ubicados consecutivamente en un soporte de almacenamiento. De tal forma, que para acceder a un registro determinado, debemos pasar obligatoriamente por todos los registros que le preceden.
- Los registros están ubicados físicamente en una secuencia usualmente fijada por uno o más campos de control contenidos dentro de cada registro, en forma ascendente o descendente.
- Esta organización tiene el último registro en particular, contiene una marca (en general un asterisco) de fin de archivo, la cual se detecta con funciones tipo EOF (end of file) o FDA (Fin de Archivo).
- En las estructuras de datos, una estructura tiene acceso secuencial si sólo se pueden acceder a los valores contenidos en un determinado orden.

## Acceso secuencial



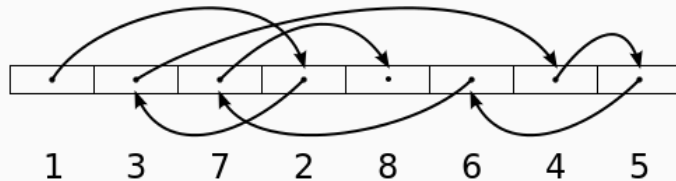
## Organización o Acceso Aleatorio o Directo

- Un archivo de acceso Directo es aquel que se encuentra almacenado en un dispositivo direccionable; y donde sus registros poseen una numeración que identifica inequívocamente a cada registro.
- Los datos se colocan y se acceden aleatoriamente mediante su posición, es decir, indicando el lugar relativo que ocupan dentro del conjunto de posiciones posibles.
- En esta organización se pueden leer y escribir registros, en cualquier orden y en cualquier lugar.
- Poseen ciertos inconvenientes como: Establecer la relación entre la posición que ocupa un registro y su contenido o puede desaprovechar parte del espacio destinado al archivo.
- Cuentan con la ventaja de la rapidez de acceso a un registro cualquiera.

# ESTRUCTURA DE DATOS: Organización o Acceso Aleatorio o Directo

- De esta forma en un archivo de acceso Directo nos referimos a un registro por medio de su posición en este y se puede acceder directamente a él sin tener que tratar todos los anteriores.
- Este tipo de ficheros permite el procesamiento de datos que necesiten ser tratados en orden no secuencial y solo es eficiente si el archivo se almacena en dispositivos que por naturaleza permitan el acceso aleatorio.

## Acceso aleatorio



Reg	Nombre	Apellidos	Dirección
1	María	...	...
2	Pedro	...	...
3	Luis	...	...
4	Carmen	...	...
5	Ana	...	...
6	Manuel	...	...
7	José	...	...
...	...	...	...

## Organización o Acceso Indexado

- Un archivo de acceso Indexado es aquel que se encuentra almacenado en un dispositivo direccionable.
- Sus registros poseen un campo que denominaremos **campo clave principal** y que identifica inequívocamente a cada registro.
- La **clave principal** es un **campo del registro** que tiene siempre un valor diferente a los ya introducidos, es decir, dentro del fichero Indexado no puede haber dos registros con los campos clave principal iguales.
- La gran ventaja de este tipo de ficheros es que usando el campo clave se puede localizar de manera inmediata el registro correspondiente al valor buscado.



## ESTRUCTURA DE DATOS: Organización o Archivos Indexados

- Además del campo clave principal pueden existir otros campos claves secundarios que permitirán la recuperación inmediata por otros criterios de búsqueda.

	Clave principal	Clave secundaria	Clave secundaria	
<b>Reg</b>	<b>DNI</b>	<b>Nombre</b>	<b>Provincia</b>	<b>Edad</b>
1	55.366.546	Luis	Sevilla	20
2	42.386.225	Teresa	Palencia	22
3	32.387.203	Carlos	Oviedo	35
4	46.399.554	Inmaculada	Gijón	12
5	60.643.434	Joaquín	Salamanca	46
6	22.543.986	José	Las Palmas	25

# ESTRUCTURA DE DATOS: Archivos Indexados

- **Ventaja:**
  - Rápido acceso, y, además, el sistema se encarga de relacionar la posición de cada registro con su contenido por medio del área de índices.
  - Gestiona las áreas de índices y excedentes.
- **Desventajas:**
  - Necesidad de espacio adicional para el área de índices.
  - El desaprovechamiento de espacio que resulta al quedar huecos intermedios libres después de sucesivas actualizaciones.

### Definición:

- La clase **File** es el único objeto del paquete java.io que representa el archivo de disco en sí.
- La clase **File** define algunos métodos independientes de la plataforma para manipular archivos.
- Puede crear, eliminar y renombrar archivos llamando a los métodos en la clase File.
- Las clases de datos pueden escribir datos en archivos, y los archivos también son los medios de datos más utilizados para flujos de datos.

## Clase File

método	valor de retorno	Descripción
getName()	String	Obtener el nombre del archivo
canRead()	boolean	Determinar si el archivo es legible
canWrite()	boolean	Determine si el archivo se puede escribir
exists()	boolean	Determine si el archivo existe
length()	long	Obtenga la longitud del archivo (en bytes)
getAbsolutePath()	String	Obtenga la ruta absoluta del archivo
getParent()	String	Obtenga su camino para padres
isFile()	boolean	Determine si el archivo existe
isDirectory()	boolean	Determine si el archivo es un directorio
isHidden()	boolean	Determine si el archivo es un archivo oculto
lastModified()	long	Obtener la última hora de modificación del archivo

- **FileWriter:** Se utiliza para escribir en el archivo siempre que el mismo exista o se le indique que lo cree antes de comenzar a escribir en el mismo.

```
public void FileWriter(){  
    File file = new File("F:\\io\\io_1.txt");  
  
    try {  
        FileWriter fw = new FileWriter(file,true);  
        String s = "¡Un lugar donde la producción es feliz!";  
  
        fw.write(s, 0, s.length()); // Escribe la cadena en el archivo  
        fw.close();  
  
    } catch (IOException e) {  
        System.out.println("Excepción de escritura de archivos" + e);  
    }  
}
```

- **FileReader:** lee el archivo de forma secuencial. Siempre que la secuencia no esté cerrada, el resto del contenido de la fuente se lee secuencialmente cada vez que se llama al método **read()** hasta que se cierra la fuente o se cierra la secuencia.

```
public void FileReader(){
    File file = new File("F:\\io\\io.txt");
    // La secuencia de caracteres lee la información del archivo
    try {
        char[] charArray = new char[1024];
        // Crea una matriz de caracteres para leer e imprimir información de texto

        FileReader fr = new FileReader(file);
        int length = fr.read(charArray);

        for(int i = 0 ; i < length ; i++){
            System.out.print(charArray[i]);
        }
        fr.close();// Cerrar la corriente

    } catch (FileNotFoundException e) {
        System.out.println("Archivo no encontrado" + e);
    } catch (IOException e) {
        System.out.println("Error al leer el archivo" + e);
    }
}
```