

**Tecnológico de Estudios  
Superiores de Jilotepec**

# **Programación Orientada a Objetos**

Investigación

# **FLUJOS Y ARCHIVOS**

Alumno(a): Vanesa Hernández Martínez

Matricula: 202223176

Grupo: 3203

Carrera: Ingeniería en Sistemas

Semestre 2

Docente: Yadira Esther Jiménez Pérez

Fecha de envió: 17 de junio del 2023

Ciclo Escolar: 2023-01

# Unidad VI. FLUJOS Y ARCHIVOS

## Contenido

6.1_Definición .....	3
6.2_Clasificación: Archivos de texto y binarios.....	9
6.3_Operaciones básicas y tipos de acceso .....	13
6.4 Manejo de objetos persistentes .....	20
Bibliografía.....	25

# Unidad VI. FLUJOS Y ARCHIVOS

## 6.1\_Definición

---

Un **archivo o fichero** de datos es una colección de registros relacionados entre si que tienen aspectos en común y están organizados para un propósito específico.

Ejemplos de archivos:

- Ficheros de clase: contienen un conjunto de registros de los estudiantes.
- Ficheros de nóminas de una empresa.
- Inventarios
- Stocks
- Suscriptores de una página web

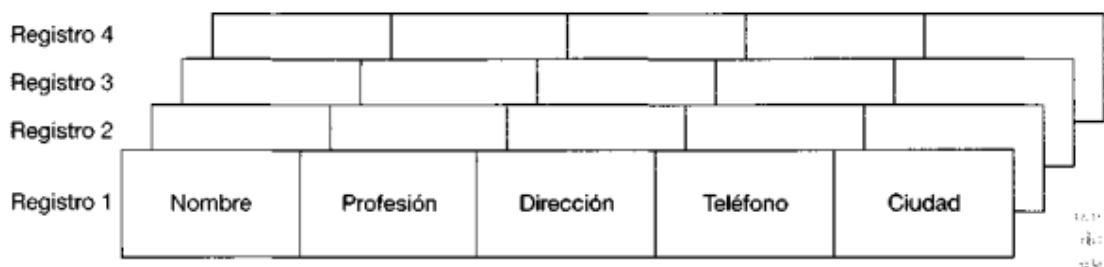


Imagen 1 Estructuras de un archivo "suscriptores"

Un archivo es un grupo de datos relacionados que usualmente se almacenan en un bloque contiguo o en un dispositivo de almacenamiento no volátil (como, por ejemplo, un disco duro). Los archivos de datos son fundamentalmente los mismos que los archivos de programa .java y .class que se utilizan para guardar programas de Java. Pero en lugar de contener programas, los archivos contienen datos que los programas leen como entrada o escriben como salida. Para evitar que el lector o la

computadora se confundan, se deben nombrar los archivos de datos con una extensión que los identifique como tal, por ejemplo: **.txt** o **.data**.

Un archivo también puede representarse como una estructura diseñada para contener datos organizados lo cual nos permite, borrar, actualizar, recuperar o incrementar los datos fácilmente.

La información de un archivo se organiza en registros, los registros en campos, los campos en bytes, y los bytes en bits. Así para que un programa pueda manejar un archivo en un dispositivo de almacenamiento permanente, como lo es un disco, primero debe crearse un flujo.

Un flujo es como un conducto a través del cual se transportarán los datos hacia o desde el dispositivo de almacenamiento. Los datos fluirán entre la memoria RAM de la computadora y el dispositivo de almacenamiento.

Dependiendo de donde salgan los datos se pueden dar dos situaciones en concreto:

1. **Output:** Si los datos van a enviarse desde la memoria hacia el disco, se trata de un flujo de SALIDA. A las operaciones de SALIDA se les conoce como de ESCRITURA (Write).
2. **Input:** si los datos van a enviarse desde el disco hacia la memoria, el flujo es de ENTRADA. A las operaciones de entrada se les conoce como de LECTURA (Read).

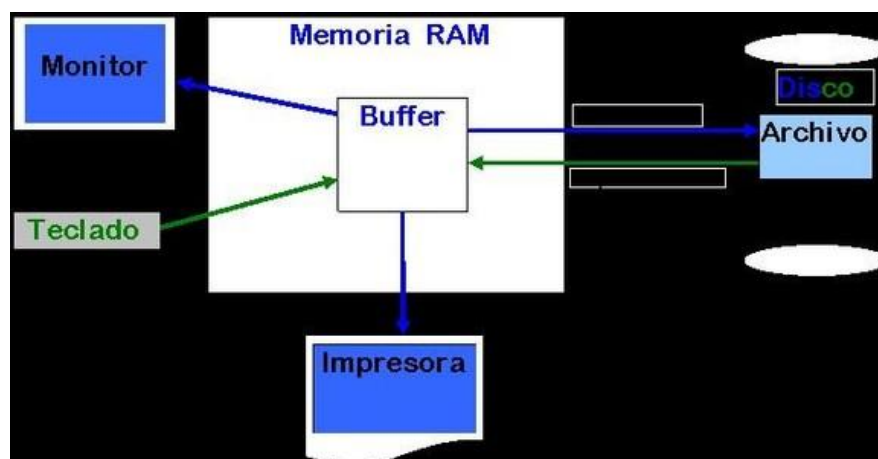


Imagen 2 Entrada y salida de flujos

# Terminología

Durante la gestión y diseño de archivos se implementen los siguientes conceptos y definiciones:

## ✓ Clave (indicativo)

Es un campo de datos que identifica el registro y lo diferencia de otros registros. Este apartado es diferente para cada espacio, algunas de las claves más utilizadas son nombres o números de identificación.

## ✓ Registro físico o bloque

Es la cantidad mínima de datos que pueden transferirse en una operación de entrada/salida entre la memoria central y los dispositivos periféricos o viceversa.

Un bloque puede contener uno o más registros lógicos. Un registro lógico puede ocupar menos de un registro físico, un registro físico o más de un registro físico.

### Ejemplos:

- Tarjeta perforada
- Línea de impresión.
- Sector de un disco magnético.

## ✓ Factor de bloqueo

El número de registros lógicos que puede contener un registro físico se denomina factor de bloqueo. Se pueden presentar los siguientes casos:

- Registro lógico > Registro físico.

En un bloque se contienen varios registros físicos por bloque; se denominan registros expandidos.

- Registro lógico = Registro físico.

El factor de bloqueo es 1 y se dice que los registros no están bloqueados.

- Registro lógico < Registro físico.

El factor de bloqueo es mayor que 1 y los registros están bloqueados.

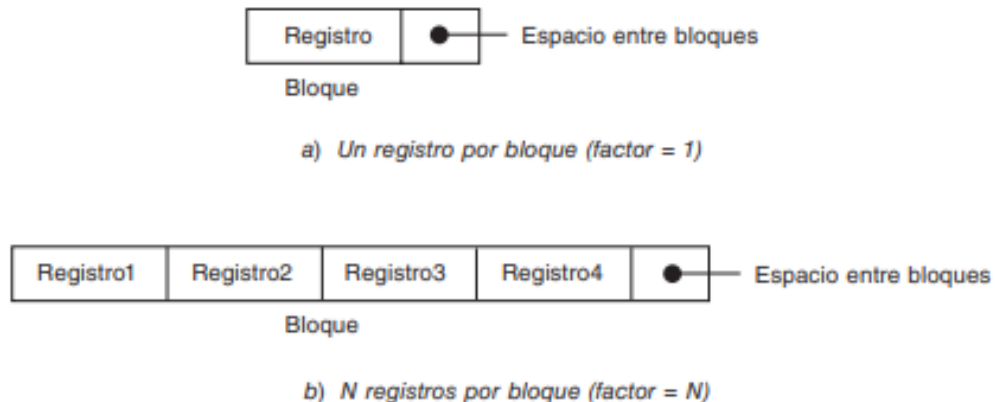


Imagen 3 Factor de bloqueo

Un archivo está siempre almacenado en un soporte externo a la memoria central. Todo programa de tratamiento intercambia información con el archivo y la unidad básica de entrada/salida es el registro. La información que almacena es permanente. En un momento dado, los datos extraídos por el archivo son los de un registro y no los del archivo completo. Los archivos en memoria auxiliar permiten una gran capacidad de almacenamiento.

### Conceptos lógicos y físicos de un registro:

- ❖ **Carácter:** Unidad de información, puede ser algún número, letra o símbolo.
- ❖ **Campo:** Espacio de almacenamiento para un dato en particular.

- ❖ **Registro:** Tipo de dato estructurado formado por la unión de varios elementos bajo una misma estructura.
- ❖ **Base de datos:** Recopilación u organizada de información o datos estructurados.

Tabla 1 Unidades de datos lógicos y físicos

Organización lógica	Organización física	Descripción
	Bit	Un dígito binario.
Carácter	Byte (octeto, 8 bits)	En la mayoría de los códigos un carácter se representa aproximadamente por un byte.
Campo	Palabra	Un campo es un conjunto relacionado de caracteres. Una palabra de computadora es un número fijo de bytes.
Registro	Bloque (1 página = bloques de longitud)	Los registros pueden estar bloqueados.
Archivo	Área	Varios archivos se pueden almacenar en un área de almacenamiento.
Base de datos	Áreas	Colección de archivos de datos relacionados que se pueden organizar en una base de datos.

## Java

En Java, un **archivo** es una secuencia de bytes que representan los datos almacenados. Java dispone de clases para trabajar las secuencias de bytes como datos de tipos básicos (int, double, String, etc.), incluso para escribir o leer objetos del archivo. El diseño del archivo es el que establece la forma de manejar las secuencias de bytes, ya sea con una organización secuencial, o como acceso directo.

Un **flujo** es una abstracción referente a una corriente de datos que van desde un origen o fuente (productor) hasta un destino o sumidero (consumidor); entre uno y otro debe existir una conexión o canal por donde circulan los datos con un flujo.

Al comenzar la ejecución de un programa en Java se crean automáticamente tres objetos flujo, canales por los que pueden fluir datos de entrada o salida, estos son:

- **System.in;** de entrada, estándar, permite el ingreso al programa de flujos de bytes desde el teclado.

- **System.out;** de salida estándar, permite al programa imprimir datos por pantalla.
- **System.err;** de salida estándar de errores, permite al programa imprimir errores por pantalla

En Java, un archivo es simplemente un flujo externo, una secuencia de bytes almacenados en un dispositivo externo, normalmente en disco; si el archivo se abre para salida, es un flujo de archivo de salida; si el archivo se abre para entrada, es un flujo de archivo de entrada. Los programas leen o escriben en el flujo, que puede conectarse a un dispositivo o a otro; el flujo, por lo tanto, es una abstracción porque las operaciones que realizan los programas son sobre él, independientemente del dispositivo al que se asocie.



## 6.2\_Clasificación: Archivos de texto y binarios——

Basado en el tipo de datos que manejan en sus registros, los archivos se clasifican en archivos en dos, archivos de texto y archivos binarios.

### ✓ Archivos de texto

Los archivos de texto están compuestos por texto sin formato, solo caracteres, los cuales se pueden codificarse de distintas maneras dependiendo de la lengua usada. También se les conoce como archivos de texto llano o texto simple ya que carece de información destinada a generar formatos y tipos de letra.

Los archivos de texto son un caso particular de archivos de organización secuencial. Es por eso que un archivo de texto es una serie continua de caracteres que se pueden leer uno tras otro.

También se puede ver a un archivo de texto como un archivo en el que cada registro pertenece a una cadena de caracteres. El tratamiento de archivos de texto es elemental y en lenguajes como Pascal es posible detectar lecturas de caracteres especiales con un fin de línea o un fin de archivo.

Los archivos de texto tienen más restricciones que los archivos binarios, ya que estos primeros solo pueden contener datos de texto, pero a diferencia de los archivos binarios, los archivos de texto son menos propensos a corromperse; si se presenta un error dentro de estos archivos se muestra una vez que se haya abierto el archivo, pero si se comete un error dentro de un archivo binario por mas pequeño que sea, pueden hacer que el archivo ya no sea legible.

Los archivos de texto se pueden guardar como:

- **Texto sin formato (formato .TXT)**

Un archivo de texto sin formato típico contiene varias líneas de texto, cada una seguida de un carácter de fin de línea (EOL). Un carácter de fin de archivo (EOF) se coloca después del último carácter, señalando el final del archivo.

- **Texto enriquecido (formato.RTF)**

Los archivos de texto enriquecido utilizan una estructura de archivos similar, pero también pueden incluir estilos de texto (como negrita y cursiva), así como la página de información de formateo.

Ambos tipos de texto, el enriquecido y sin formato, incluyen un grupo de codificación de caracteres que determina cómo los caracteres son interpretados y qué caracteres nos pueden mostrar.

Los datos en los archivos de texto se graban como secuencias de bytes.

**Ejemplo:**

123456

El dato a pesar de contener números se va a guardar como una secuencia de 6 bytes y no como un entero, por lo cual no es posible realizar operaciones matemáticas con él.

El manejo de archivos de texto se puede llevar a cabo por medio de dos tipos de flujos: de bytes y de caracteres.

## ✓ **Archivos Binarios**

El utilizar archivos binarios es una manera muy eficiente de almacenar números en el disco, En este tipo de archivos a diferencia de los de texto, los números ya no se almacenan como caracteres, sino que se almacena de la misma manera que lo hacen en memoria.

Un archivo binario es un archivo informático que contiene información de cualquier tipo, codificada en forma binaria para el propósito de almacenamiento y procesamiento de ordenadores.

Estos archivos contienen una secuencia de bytes, o agrupaciones ordenadas de ocho bits. Cuando se crea un formato de archivo personalizado para un programa, un desarrollador organiza estos bytes en un formato que almacena la información necesaria para la aplicación. Estos formatos pueden incluir múltiples tipos de datos en el mismo archivo, tales como datos de audio, imágenes y video. Estos datos pueden ser interpretados por los programas de apoyo, pero se mostrarán como texto ilegible en un editor de texto.

### Ejemplo:



Imagen 4 Ejemplo de un archivo binario

El visor de imágenes reconoce los datos binarios y muestra la imagen, pero cuando la imagen se abre en un editor de texto, los datos binarios se convierten en texto irreconocible y solo hay pequeñas partes de texto legible, algunos otros tipos de archivos binarios incluyen secciones de texto legible de esta manera. Por lo que es posible encontrar algo de información acerca de un tipo de archivo binario desconocido abriéndolo en un editor de texto.

Los archivos binarios pueden contener encabezados, que son bytes de datos al comienzo de un archivo que identifica el contenido del archivo, estos a menudo incluyen el tipo de archivo y otra información descriptiva.

Cuando se hacen operaciones con datos de alguno de los tipos primitivos (bool, byte, double, float, int, long, short, etc.), estos se escriben y leen en formato binario.

## **¿Qué sucede si nos encontramos con un archivo desconocido, o no sabemos identificar el archivo?**

Lo que se puede hacer es abrir el archivo dentro de un editor de texto, si se abre sin problema y muestra un código en texto que es totalmente legible se podría asegurar que el archivo es de texto, en cambio, si el texto no es legible se trata de un archivo binario. Por otro lado, si el archivo no abre en un editor de texto es un archivo binario ya que requerirá ser abierto por un programa apropiado.

También existe la posibilidad de entrar a [FileInfo.com](http://FileInfo.com) y buscar la extensión del archivo.

Los dos tipos de archivos pueden tener el mismo aspecto en la superficie, sin embargo, codifican datos de manera diferente. Ambos tipos de archivos contienen datos almacenados como una serie de bits (valores binarios de 1s y 0s), la diferencia es que los bits de los archivos de texto representan caracteres, mientras que los bits en los archivos binarios representan datos personalizados, sin olvidar que los archivos de texto contienen solamente datos de texto, mientras que los archivos binarios pueden contener tanto datos binarios como de texto.

	Flujos con Bytes	Flujos con caracteres
E N T R A D A S	InputStream	Reader
	ByteArrayInputStream	BufferedReader
	FileInputStream	LineNumberReader
	FilterInputStream	CharArrayReader
	BufferedInputStream	FilterReader
	DataInputStream	PushbackReader
	PushbackInputStream	InputStreamReader
	ObjectInputStream	FileReader
	PipedInputStream	PipedReader
	SequenceInputStream	StringReader
	StringBufferInputStream	
S A L I D A S	OutputStream	Writer
	ByteArrayOutputStream	BufferedWriter
	FileOutputStream	CharArrayWriter
	FilterOutputStream	FilterWriter
	BufferedOutputStream	OutputStreamWriter
	DataOutputStream	FileWriter
	PrintStream	PipedWriter
	ObjectOutputStream	PrintWriter
	PipedOutputStream	StringWriter

Imagen 5 Entrada y salida de flujos

## 6.3\_Operaciones básicas y tipos de acceso —

### ➤ Operaciones básicas

Las distintas operaciones que se pueden realizar son:

- ✓ Creación.
- ✓ Consulta.
- ✓ Actualización (altas, bajas, modificación, consulta).
- ✓ Clasificación.
- ✓ Reorganización.
- ✓ Destrucción (borrado).
- ✓ Reunión, fusión.
- ✓ Rotura, estallido.
- ✓

### 6.3.1 Creación de un archivo

Esta operación es la primera que se ejecutara en un archivo de datos, dentro de la misma se realizará una elección de un entorno descriptivo que permita un ágil, rápido y eficaz tratamiento del archivo.

Un archivo tiene que existir para que este pueda ser utilizado, es decir la información del archivo tienen que haber sido almacenadas sobre un soporte y ser utilizable. Para tener una correcta creación de archivos, se necesita tener organización, estructura, localización o reserva de espacio en el soporte de almacenamiento y transferencia del archivo del soporte antiguo al nuevo.

Un archivo puede ser creado por primera vez en un soporte, proceder de otro previamente existente en el mismo o diferente soporte, y también puede ser el resultado de un cálculo o ambas cosas a la vez.

#### Ejemplo:

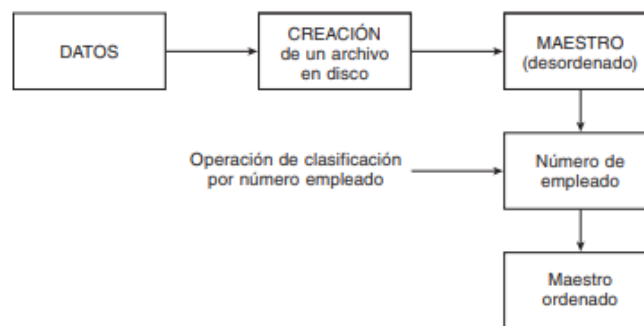


Imagen 6 Creación de un archivo ordenado de empleados.

### 6.3.2. Consulta de un archivo

Esta operación es la que permite al usuario acceder al archivo para conocer el contenido de uno, varios o todos los registros que se contienen.

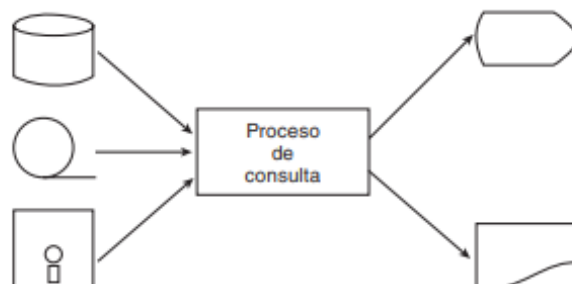


Imagen 7 Consulta de un archivo.

### 6.3.3. Actualización de un archivo

Es la operación que permite generar actualizaciones (cambios y modificaciones) al archivo, lo cual nos permite realizar las siguientes operaciones con sus registros:

- Consulta del contenido de un registro.
- Inserción de un registro nuevo en el archivo.
- Supresión de un registro existente.
- Modificación de un registro.

#### Ejemplo:

Los archivos de una tienda o almacén requieren actualizaciones simultáneamente acerca de las existencias de los artículos en tienda, sus precios y los proveedores.

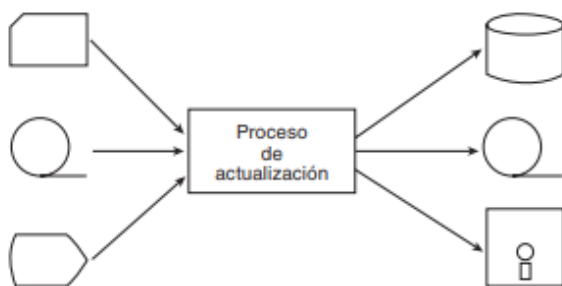


Imagen 8 Actualización de un archivo (I)

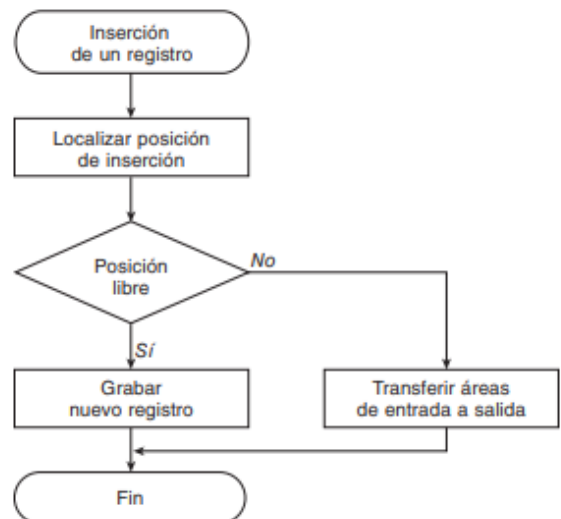


Imagen 9 Actualización de un archivo (II)

### 6.3.4. Clasificación de un archivo

Es una operación muy importante, la clasificación se realiza de acuerdo con el valor de un campo específico, si es alfabético o numérico ascendente (creciente), o descendente (decreciente).

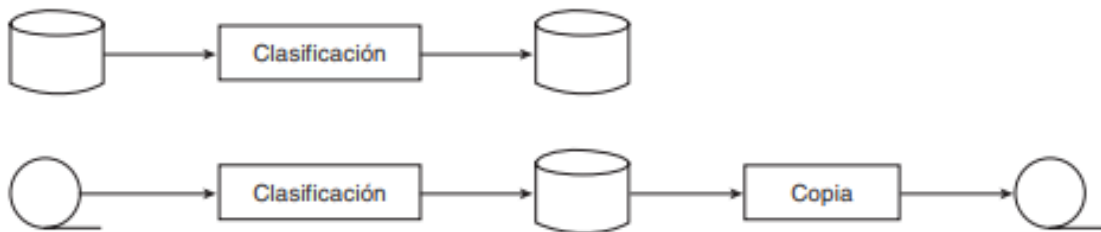


Imagen 10 Clasificación de un archivo

### 6.3.5. Reorganización de un archivo

La reorganización de un archivo consiste en la copia de un nuevo archivo a partir del archivo modificado, con el fin de obtener una nueva estructura que sea lo más óptima posible, ya que las operaciones sobre archivos modifican la estructura inicial de un archivo (los índices, enlaces, zonas de sinónimos, zonas de desbordamiento), haciendo más lento su procesamiento.

### 6.3.6. Destrucción de un archivo

Es cuando se destruye un archivo, el cual ya no vamos a poder utilizar y del cual ya no vamos a obtener acceso a ninguno de sus registros.



Imagen 11 Reorganización de un archivo



### 6.3.7. Reunión, fusión de un archivo

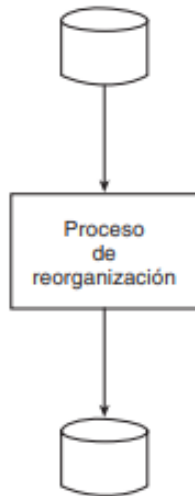


Imagen 12 Reorganización de un archivo

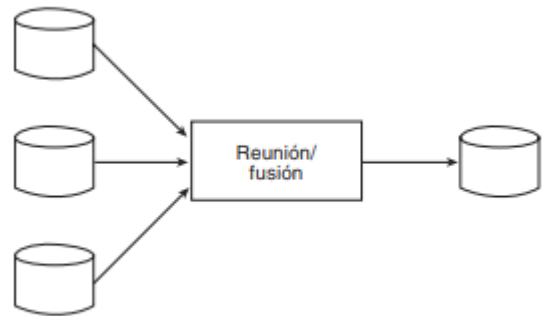


Imagen 13 Fusión de un archivo

- ✓ **Reunión:** Permite obtener un archivo a partir de otros varios.
- ✓ **Fusión:** Es cuando se reúnen varios archivos en uno solo, intercalándose unos en otros, siguiendo criterios determinados.

### 6.3.8. Rotura/estallido de un archivo

Es la operación de obtener varios archivos a partir de un mismo archivo inicial.

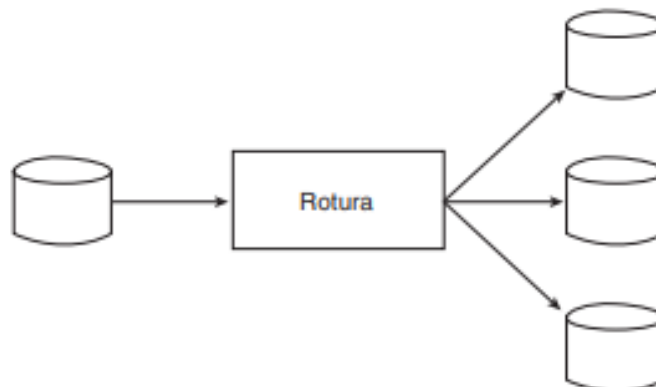


Imagen 14 Rotura de un archivo

## Métodos de acceso

Se refiere a las capacidades que el subsistema de archivos provee para acceder datos dentro de los directorios y medios de almacenamiento en general. A continuación, se presenta una breve descripción de cada uno de los métodos de acceso.

### ✓ Accesos secuenciales

Los registros se almacenan por posición, cada registro tiene el mismo tamaño y el mismo número de campos. El primero registro de un campo se lee como campo clave. Los archivos secuenciales son la forma más común de estructura de archivos.

Se necesita almacenar los valores de cada campo; el nombre del campo y la longitud de cada uno son atributos de la estructura del archivo. Cada registro tiene un campo clave que lo identifica (generalmente es el primero de cada registro). Los registros se almacenan en secuencia por la clave.

Este tipo de acceso se utilizan normalmente en aplicaciones de procesos por lotes, porque es la única organización de archivos que se puede guardar tanto en cintas como en discos.

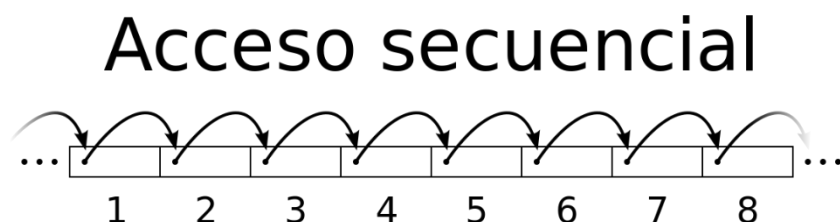


Imagen 15 Acceso secuencial

### ✓ Accesos indexados

Los registros se organizan en una secuencia basada en un campo clave presentando dos características, un índice del archivo para soportar los accesos aleatorios y un archivo de desbordamiento. El índice nos ayuda al momento de realizar una búsqueda ya que se llega rápidamente al registro deseado y el archivo de desbordamiento es similar al archivo de registros usado en un archivo

secuencial, pero está integrado de forma que los archivos de desbordamiento se ubiquen siguiendo un puntero desde su registro predecesor.

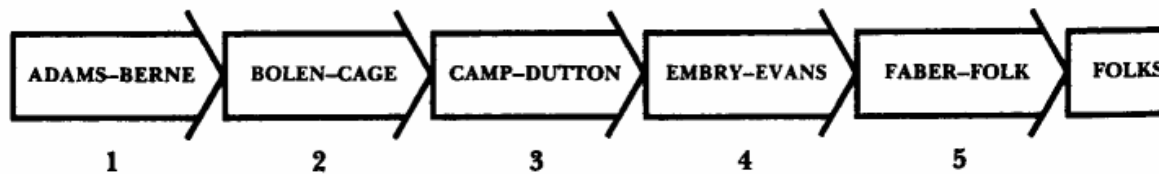


Imagen 16 Acceso indexado

### ✓ Accesos directos o de dispersión

Este acceso aprovecha la capacidad de los discos para entrar en cualquier bloque de dirección que se quiera utilizar, esto requiere de un campo clave para cada registro como los anteriores dos métodos. Su ordenamiento no es secuencial.

Explotan la capacidad de los discos para acceder directamente a cualquier bloque de dirección conocida. Este acceso requiere un campo clave en cada registro.

Los archivos directos son muy usados donde se necesita un acceso muy rápido, donde se usan registros de longitud fija y donde siempre se accede a los registros de una vez.

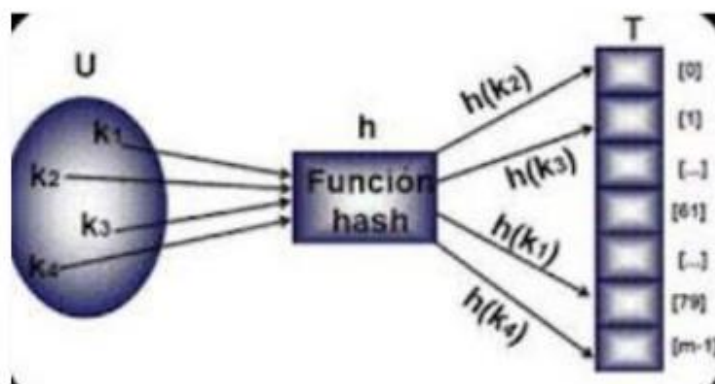


Imagen 17 Acceso de dispersión

## 6.4\_Manejo de objetos persistentes

---

Las clases persistentes son clases en una aplicación que implementan las entidades del problema empresarial. No se considera que todas las instancias de una clase persistente estén en estado persistente ya que una instancia puede ser transitoria o separada.

Muchas aplicaciones Java necesitan tratar con datos persistentes. En la mayoría de los casos, esto significa enfrentarse con una base de datos relacional, posiblemente un base de datos antigua o un Sistema de Manejo de Bases de Datos (DBMS) estándar industrial.

Hibernate y Java Data Objects están diseñados para proporcionar al desarrollador una persistencia transparente, es decir, la aplicación trata con objetos persistentes utilizan un API orientado a objetos sin la necesidad de código SQL en el código Java.

Hibernate tiene un mayor rendimiento si las clases siguen algunas reglas simples, también conocidas como el modelo de programación POJO (Plain Old Java Object). Sin embargo, ninguna regla está declarada estrictamente para cumplirse.

### Ejemplo de POJO:

La mayoría de aplicaciones Java requieren una clase persistente que represente a los felinos.

```
package eg;
import java.util.Set;
import java.util.Date;

public class Cat {
    private Long id; // identifier
    private Date birthdate;
    private Color color;
    private char sex;
    private float weight;
    private int litterId;
    private Cat mother;
    private Set kittens = new HashSet();
    private void setId(Long id) {
```

```

        this.id=id;
    }
    public Long getId() {
        return id;
    }
    void setBirthdate(Date date) {
        birthdate = date;
    }
    public Date getBirthdate() {
        return birthdate;
    }
    void setWeight(float weight) {
        this.weight = weight;
    }
    public float getWeight() {
        return weight;
    }
    public Color getColor() {
        return color;
    }
    void setColor(Color color) {
        this.color = color;
    }
    void setSex(char sex) {
        this.sex=sex;
    }
    public char getSex() {
        return sex;
    }
    void setLitterId(int id) {
        this.litterId = id;
    }
    public int getLitterId() {
        return litterId;
    }
    void setMother(Cat mother) {
        this.mother = mother;
    }
    public Cat getMother() {
        return mother;
    }
    void setKittens(Set kittens) {
        this.kittens = kittens;
    }
    public Set getKittens() {
        return kittens;
    }

```

```
    }  
    public void addKitten(Cat kitten) {  
        kitten.setMother(this);  
        kitten.setLitterId( kittens.size() );  
        kittens.add(kitten);  
    }  
}
```

## Cuatro reglas principales de las clases persistentes

- **Implemente un constructor sin argumentos**

Todas las clases persistentes deben tener un constructor predeterminado (el cual puede tener cualquier modificador de acceso) de modo que Hibernate pueda instanciarlas usando `Constructor.newInstance()`.

- **Proporcione una propiedad identificador**

Cat tiene una propiedad llamada `id`, esta propiedad mapea a la columna de la llave principal de la tabla de la base de datos. La propiedad puede tener cualquier nombre, algún dato primitivo y cualquier tipo de "wrapper" primitivo, `java.lang.String` o `java.util.Date`.

La propiedad identificadora es opcional, pero solo algunas funcionalidades se encuentran disponibles para clases que declaran una propiedad identificadora:

- `Session.saveOrUpdate()`
- `Session.merge()`

- **Prefiera las clases no finales**

Los proxies, dependen de que las clases persistentes sean no finales o de la implementación de una interfaz que declare todos los métodos públicos.

Con Hibernate pueden existir las clases finales que no implementen una interfaz, pero no se podrán utilizar proxies para la recuperación perezosa de asociaciones, lo cual limitará sus opciones para afinar el rendimiento.

Así mismo se debe evitar el declarar métodos public final en las clases no finales.

- **Declare métodos de acceso y de modificación para los campos persistentes**

Cat declara métodos de acceso para todos sus campos persistentes. Lo mejor es proporcionar una indirección entre el esquema relacional y las estructuras de datos internos de la clase, Hibernate persiste las propiedades del estilo JavaBeans, y reconoce los nombres de método de la forma getFoo, isFoo y setFoo.

No es necesario declarar públicas las propiedades, son compatibles con cualquier modificador de acceso.

Los archivos de flujo en la programación orientada a objetos (POO) son utilizados para realizar operaciones de entrada y salida de datos en archivos. Estos archivos permiten almacenar información de manera persistente en el sistema de archivos del equipo.

El proceso de trabajar con archivos de flujo involucra tres pasos principales: apertura, procesamiento y cierre del archivo. Durante la apertura del archivo, se establece una conexión entre el programa y el archivo mediante la creación de un objeto de flujo.

En el procesamiento del archivo, se pueden realizar operaciones de lectura y escritura de datos. La lectura permite obtener información del archivo, mientras que la escritura permite almacenar datos en él. Es importante manejar posibles errores durante el procesamiento del archivo, como archivos que no existen o errores de permisos, es ahí donde entran las excepciones.

Al finalizar las operaciones en el archivo, es necesario cerrarlo adecuadamente para liberar los recursos asociados y garantizar que los datos se guarden correctamente.

Los archivos de flujo son utilizados para interactuar con diferentes tipos de archivos, como texto, binarios o de acceso aleatorio. Estos archivos se pueden utilizar para diversas aplicaciones, como almacenar configuraciones de programas, guardar registros de actividad o realizar operaciones de respaldo de datos.

La manipulación de archivos de flujo en programación orientada a objetos permite crear programas más flexibles y robustos, al ofrecer una forma eficiente de gestionar la entrada y salida de información en archivos.



# Bibliografía

---

## Libros:

Johanes, L (2003) Fundamentos de programación. Algoritmos, estructura de datos y objetos. Tercera edición. McGrawJill.

Johanes, L (2008) Fundamentos de programación. Algoritmos, estructura de datos y objetos. Cuarta edición. McGrawJill.

Johanes, L y Zahonero, I (2011). Programación en Java. Algoritmos y programación orientada a objetos. Primera edición. McGrawJill.

John, S y Raymond, H (2009) Introducción a la programación con Java. Primera edición. McGrawJill.

## Fuentes de internet:

*Archivos Directos o de Dispersión.* (s/f). Edu.mx. Recuperado el 16 de junio de 2023, de

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro26/archivos\\_directos\\_o\\_de\\_dispersin.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro26/archivos_directos_o_de_dispersin.html)

*TPM.* (s/f). Edu.mx. Recuperado el 16 de junio de 2023, de

<https://www.itslr.edu.mx/archivos2013/TPM/temas/s2u6.html>