



TECNOLOGICO NACIONAL DE MEXICO INSTITUTO TECNOLÓGICO DEL VALLE DE OAXACA

Ingeniería Informática

Materia:

ADMINISTRACIÓN Y ORGANIZACIÓN DE DATOS

Unidad 1:

Introducción

Tarea 1:

Archivos en Java

Alumnos:

- Diego Emiliano Peña Rojas
- Axel Tadeo Vásquez López

Docente:

Benedicto Ramírez Santiago

Grupo: A

Semestre: 4

Elaborado:

27 de Febrero 2026



ÍNDICE

| | |
|--|----|
| Introducción | 3 |
| Desarrollo..... | 4 |
| 1.1 Concepto de archivo | 4 |
| 1.2 Tipos de archivos | 7 |
| 1.3 Organización de archivos..... | 8 |
| Secuencial | 9 |
| Aleatoria..... | 9 |
| Indexada | 10 |
| 1.4 Operaciones sobre archivos | 11 |
| 1.5 Almacenamiento y medios físicos | 17 |
| Conclusión | 19 |
| Bibliografía | 20 |



Introducción

En el ámbito de la informática, el manejo de archivos constituye un elemento fundamental para el almacenamiento, organización y recuperación de la información. Los archivos permiten conservar datos de manera permanente, incluso cuando un programa deja de ejecutarse o el equipo es apagado, lo que los convierte en la base del funcionamiento de sistemas operativos, aplicaciones, bases de datos y prácticamente cualquier software moderno. Comprender cómo se estructuran, clasifican y manipulan los archivos es esencial para el desarrollo eficiente de programas y para la correcta administración de la información.

Este trabajo aborda los conceptos esenciales relacionados con los archivos, comenzando por su definición y la importancia que tienen como unidades de almacenamiento dentro de un sistema computacional. Posteriormente, se analizan los distintos tipos de archivos según su contenido, así como los métodos de organización más utilizados para estructurar los datos de forma secuencial, directa o indexada. También se presentan las operaciones principales que pueden realizarse sobre ellos, tales como creación, lectura, escritura, modificación y eliminación. Finalmente, se revisan los medios físicos de almacenamiento que permiten conservar la información, destacando sus características y diferencias tecnológicas.

El propósito de este documento es ofrecer una visión clara y comprensible sobre el manejo de archivos, proporcionando explicaciones detalladas y ejemplos que faciliten su comprensión, especialmente en el contexto de la programación y el desarrollo de software



Desarrollo

1.1 Concepto de archivo

Un archivo es una colección organizada de información que se almacena de manera permanente en un medio físico con el propósito de conservar datos para su uso posterior. En el ámbito de la informática, un archivo constituye una unidad básica de almacenamiento que permite registrar, consultar y manipular información de forma estructurada.

Desde una perspectiva técnica, un archivo es una colección de registros lógicamente relacionados. Esto significa que los datos que contiene guardan relación entre sí y pertenecen a un mismo contexto o propósito.

Archivo como conjunto de bits

Desde un punto de vista más técnico, un archivo o fichero informático es un conjunto de bits almacenados en un dispositivo de almacenamiento.

Un bit es la unidad mínima de información digital y puede tomar dos valores: 0 o 1. Todos los archivos, sin importar su tipo (texto, imagen, audio, video o datos), están formados internamente por secuencias de bits que el sistema interpreta según su formato.

Por ejemplo:

- Un archivo de texto almacena bits que representan caracteres.
- Una imagen almacena bits que representan colores y posiciones.
- Un archivo ejecutable almacena bits que representan instrucciones de programa.

Aunque el usuario visualiza información comprensible, el sistema operativo y el hardware únicamente manejan secuencias binarias.



Archivo como conjunto organizado de información

Otra definición importante establece que un archivo es una colección de información o datos relacionados entre sí, organizados bajo un mismo propósito y almacenados como soporte material de dicha información.

Esto implica que:

- Los datos dentro de un archivo pertenecen al mismo tipo o temática.
- Se utilizan dentro de un mismo método o proceso.
- Sirven como respaldo permanente de la información generada por un sistema.

Por ejemplo:

- Un archivo de ventas contiene únicamente datos relacionados con transacciones.
- Un archivo de empleados almacena exclusivamente información del personal.
- Un archivo de configuración guarda parámetros de funcionamiento de un programa.

La organización permite que la información pueda:

- Ser consultada fácilmente.
- Ser modificada cuando sea necesario.
- Mantener coherencia y consistencia.



Concepto de archivo en programación

En programación, un archivo representa un mecanismo de almacenamiento permanente que permite guardar datos fuera de la memoria principal (RAM). Esto es fundamental porque la memoria RAM es volátil; es decir, pierde su contenido al apagar el equipo o cerrar el programa.

Por ello, los archivos permiten:

- Persistencia de datos.
- Intercambio de información entre programas.
- Respaldo y recuperación de información.
- Organización estructurada de grandes volúmenes de datos.

En lenguajes como Java, los archivos se manejan mediante clases especializadas que permiten:

- Crear archivos.
- Leer información.
- Escribir datos.
- Modificar contenido.
- Eliminar archivos.



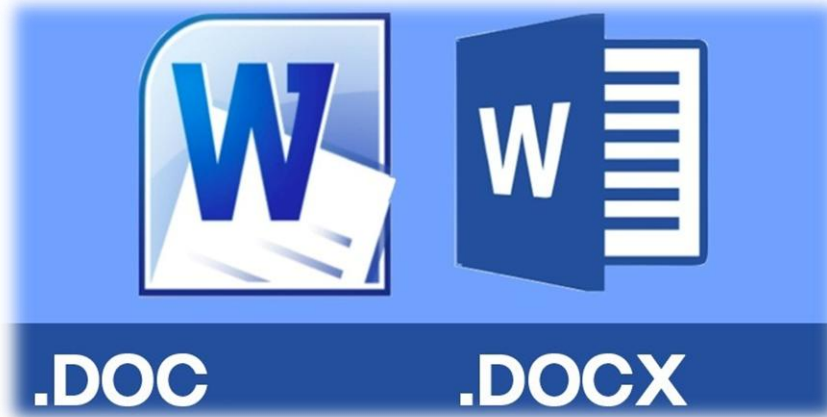
1.2 Tipos de archivos

Los archivos pueden clasificarse de distintas maneras dependiendo de diversos criterios técnicos; sin embargo, una de las clasificaciones más importantes dentro del ámbito de la programación y la administración de datos es aquella que se basa en el contenido que almacenan. Esta clasificación permite comprender la naturaleza de la información guardada, la forma en que será interpretada por el sistema y el tipo de procesamiento que podrá realizarse sobre ella.

Según su contenido

a) Archivos de texto

Los archivos de texto son aquellos que almacenan información en forma de caracteres legibles para el ser humano. Los datos se representan mediante códigos de caracteres como ASCII o Unicode, lo que permite que puedan visualizarse directamente en editores simples como el bloc de notas o cualquier procesador de texto. La información contenida en este tipo de archivo es visible sin necesidad de programas especializados, lo que facilita su lectura y modificación. Debido a que almacenan los datos como caracteres, generalmente ocupan más espacio que los archivos binarios cuando se trata de grandes volúmenes de información. Entre los ejemplos más comunes se encuentran los archivos con extensiones .txt, .csv, .xml y .json. En programación, los archivos de texto se utilizan frecuentemente para guardar registros simples, exportar información entre sistemas o almacenar configuraciones de programas.





b) Archivos binarios

Los archivos binarios son aquellos que almacenan la información en formato binario, es decir, en secuencias de bits que no pueden ser interpretadas directamente por el usuario sin un programa específico. A diferencia de los archivos de texto, los datos no se representan como caracteres visibles, sino como información codificada en su forma interna. Esto los hace más eficientes para almacenar datos complejos y estructuras completas, ya que optimizan el uso del espacio y permiten mayor rapidez en ciertos procesos. No son legibles directamente con editores comunes y requieren aplicaciones especializadas para su interpretación. Entre los ejemplos de archivos binarios se encuentran los archivos con extensión .dat, .exe, .class y .ser, así como imágenes en formatos como .jpg o .png. En programación, los archivos binarios se utilizan para guardar objetos serializados, almacenar imágenes o videos y manejar datos estructurados de manera eficiente.



1.3 Organización de archivos

La organización de archivos se refiere a la manera en que los registros se estructuran y almacenan físicamente dentro de un archivo, así como a la forma en que pueden ser localizados y recuperados posteriormente. Este aspecto es fundamental en los sistemas de información, ya que influye directamente en la eficiencia del acceso, la actualización y la administración de los datos. Dependiendo de las necesidades del sistema, el volumen de información y la frecuencia de consulta, pueden emplearse distintos métodos de organización. Entre los más importantes se encuentran la organización secuencial, la organización aleatoria o directa y la organización indexada.



Secuencial

La organización secuencial es una de las formas más simples y tradicionales de almacenar información. En este tipo de organización, los registros se guardan uno tras otro siguiendo un orden determinado, generalmente basado en un criterio lógico como una clave numérica, una fecha o un identificador. Para acceder a un registro específico, el sistema debe recorrer previamente todos los registros que lo anteceden, lo que implica un acceso lineal.

Este método resulta adecuado cuando los datos se procesan en el mismo orden en que fueron almacenados, como ocurre en la generación de reportes, listados cronológicos o procesos de lectura completa del archivo. Su principal ventaja radica en la simplicidad de implementación y en su eficiencia cuando se realizan operaciones masivas de lectura. Sin embargo, cuando se requiere localizar un registro específico dentro de grandes volúmenes de información, el tiempo de búsqueda puede incrementarse considerablemente, ya que no existe un mecanismo que permita saltar directamente a una posición determinada.

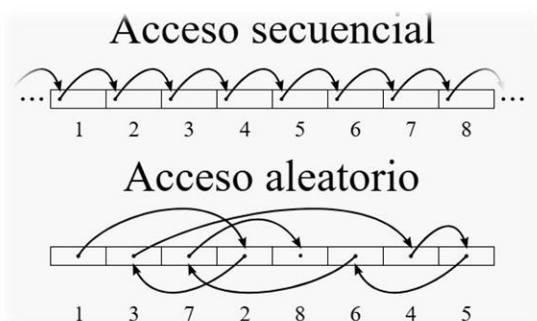


Aleatoria

La organización aleatoria, también conocida como organización directa, permite acceder a cualquier registro del archivo sin necesidad de recorrer los registros anteriores. En este modelo, cada registro se almacena en una posición específica que puede calcularse mediante una fórmula o identificarse a través de una dirección determinada. Esto permite que el sistema acceda directamente al dato requerido, reduciendo significativamente el tiempo de búsqueda.



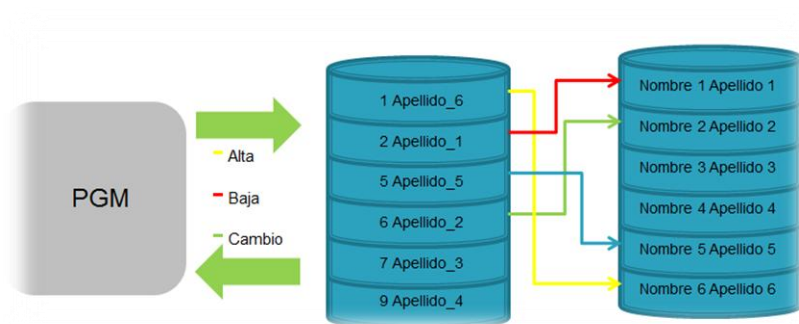
Este tipo de organización es especialmente útil en sistemas donde se realizan consultas frecuentes a registros individuales, como en aplicaciones bancarias, sistemas de inventarios o bases de datos simples. Su principal ventaja es la rapidez en el acceso a la información. No obstante, su implementación suele requerir una planificación más detallada para garantizar que los registros mantengan una estructura uniforme que permita su localización exacta dentro del archivo.



Indexada

La organización indexada combina características de la organización secuencial y de la organización directa. En este modelo, además del archivo principal que contiene los registros, se crea una estructura adicional llamada índice. Este índice almacena referencias que indican la ubicación exacta de cada registro dentro del archivo principal, facilitando así su búsqueda.

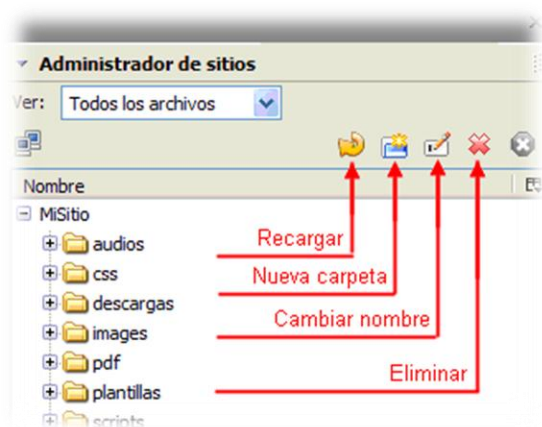
Gracias al uso del índice, el sistema puede localizar rápidamente un registro sin necesidad de recorrer todo el archivo, lo que mejora considerablemente la eficiencia en la recuperación de datos, especialmente cuando se manejan grandes cantidades de información. Esta organización es ampliamente utilizada en sistemas de bases de datos, ya que permite mantener los datos ordenados y, al mismo tiempo, acceder a ellos de manera rápida y estructurada. Sin embargo, requiere espacio adicional para almacenar el índice y una





administración cuidadosa para mantenerlo actualizado cuando los registros son modificados o eliminados.

1.4 Operaciones sobre archivos



Las operaciones sobre archivos son el conjunto de acciones que permiten manipular la información almacenada en un archivo dentro de un sistema informático. Estas operaciones constituyen la base del manejo de datos en cualquier aplicación, ya que permiten crear, consultar, modificar y

eliminar información según las necesidades del usuario o del sistema. En programación, el control adecuado de estas operaciones garantiza la integridad de los datos, la seguridad de la información y el correcto funcionamiento del software.

Las operaciones más comunes que pueden realizarse sobre un archivo incluyen la creación, apertura, lectura, escritura, modificación, cierre y eliminación. Cada una de estas acciones cumple una función específica dentro del ciclo de vida del archivo y suele implementarse mediante instrucciones o clases especializadas del lenguaje de programación utilizado.

- Creación: Esta operación es la primera que se realiza en cualquier archivo, lo cual requiere para su utilización, que exista un archivo, tener organización, estructura, localización o espacio en el soporte. Por ejemplo, para usar la creación cabe decir que al almacenar (Guardar) un archivo si no existía anteriormente se creara utilizando una localización, que se haya establecido anteriormente.



- Apertura: Consiste en preparar el acceso a un archivo para realizar alguna operación de lectura o escritura sobre él.
- Cerrado: Consiste en cerrar el acceso al archivo para evitar que se dañe la información.
- Lectura: Consiste en recorrer los elementos o registros de un archivo de forma secuencial o directa para su visualización o procesamiento.
- Escritura: Consiste en agregar o actualizar los elementos o registros de un archivo.
- Consulta: Esta operación sirve para poder saber qué es lo que se encuentra de datos en nuestro archivo; para eso tiene que verificar cual archivo buscara (que será con el nombre establecido) y además que ese archivo primeramente haya logrado una creación correcta.
- Actualización: Es la operación que permite modificar un dato dentro del archivo y poder volver a almacenarlos. Por ejemplo, el registro de un paciente médico que cuando llego le diagnosticaron una enfermedad pero después de un tiempo, el paciente se rehabilito, lo cual nos queda tener que actualizar esos registros y dar que el paciente se encuentra mejor.
- Ordenamiento o Clasificación: Esta operación solo nos sirve para poder llevar un control con todos los archivos creados, lo cual puede llegar a ser ascendente o descendente, dependiendo del nombre utilizado.
- Copiado o Duplicado: Es solamente crear un archivo idéntico a otro sin tener que afectar al original. Por ejemplo, cuando hacemos un respaldo que es importante



para nosotros o en todo caso cuando le queremos proporcionar a una persona el archivo, tendríamos que realizar el duplicado de la copia.

- **Concatenación:** Este Procedimiento solo se encarga de unir los datos de dos archivos, pero este procedimiento solo realiza que un archivo quede a continuación del otro o en todo caso que el segundo archivo empiece desde el final del primer archivo
- **Función o Intercalación:** Es un poco parecida a la concatenación la cual se encarga de unir datos, pero al contrario del otro esta operación es para poder mezclar ordenadamente los datos de cada uno de los archivos existentes. Por ejemplo; tenemos una serie de datos numéricos ordenados ascendentemente en los dos archivos pero no contienen los mismos números, la operación se encarga de que se unan sus datos pero de una manera que los dos queden ordenados.
- **Partición:** Consiste en que el archivo pueda ser dividido en dos o mas partes para poder realizar una transferencia mas rápida, ya sea en la red u otros accesos.
- **Compactación:** Para esta operación cabe decir que sirve para unir varios registros de un archivo que haya sido particionado, para poder tener al final un archivo único y completo.



En programación, manejar archivos correctamente es fundamental para guardar, leer y actualizar información de forma persistente. Java ofrece varias clases para realizar estas operaciones, como File, FileReader, FileWriter, BufferedReader y BufferedWriter. A continuación, se explican las operaciones más importantes:

Creación de archivos

La creación de un archivo permite que el programa genere un nuevo documento en el sistema de archivos. Es importante verificar si el archivo ya existe para evitar sobrescribir datos.

```
import java.io.File;
import java.io.IOException;

public class CrearArchivo {
    public static void main(String[] args) {
        try {
            File archivo = new File("archivo.txt");
            if (archivo.createNewFile()) {
                System.out.println("Archivo creado: " + archivo.getName());
            } else {
                System.out.println("El archivo ya existe.");
            }
        } catch (IOException e) {
            System.out.println("Ocurrió un error al crear el archivo.");
            e.printStackTrace();
        }
    }
}
```

Este código crea un archivo llamado archivo.txt. Si el archivo ya existe, no se sobrescribe, evitando pérdida de información.



Escritura en archivos

Permite guardar información dentro de un archivo. Se puede escribir desde cero o añadir contenido al final.

```
import java.io.FileWriter;
import java.io.IOException;

public class EscribirArchivo {
    public static void main(String[] args) {
        try {
            FileWriter escritor = new FileWriter("archivo.txt");
            escritor.write("Este es un ejemplo de texto en el archivo.\n");
            escritor.close();
            System.out.println("Se ha escrito correctamente en el archivo.");
        } catch (IOException e) {
            System.out.println("Error al escribir en el archivo.");
            e.printStackTrace();
        }
    }
}
```

Se utiliza FileWriter para escribir texto en archivo.txt. El método close() asegura que los datos se guarden correctamente.

Lectura de archivos

Permite recuperar el contenido de un archivo para procesarlo o mostrarlo en pantalla.

```
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;

public class LeerArchivo {
    public static void main(String[] args) {
        try {
            FileReader fr = new FileReader("archivo.txt");
            BufferedReader br = new BufferedReader(fr);
            String linea;
            while ((linea = br.readLine()) != null) {
                System.out.println(linea);
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error al leer el archivo.");
            e.printStackTrace();
        }
    }
}
```

Se utiliza BufferedReader junto con FileReader para leer línea por línea. Esto permite procesar archivos de manera eficiente.



Modificación de archivos

Modificar archivos consiste en **leer su contenido, cambiar los datos y guardarlos nuevamente**. Por ejemplo, para actualizar información en un archivo de texto:

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class ModificarArchivo {
    public static void main(String[] args) {
        try {
            Scanner entrada = new Scanner(System.in);
            System.out.print("Ingrese nuevo contenido: ");
            String texto = entrada.nextLine();
            FileWriter fw = new FileWriter("archivo.txt");
            fw.write(texto);
            fw.close();
            System.out.println("Archivo modificado correctamente.");
        } catch (IOException e) {
            System.out.println("Error al modificar el archivo.");
            e.printStackTrace();
        }
    }
}
```

Aquí el usuario ingresa un nuevo texto que reemplaza el contenido anterior. FileWriter sobrescribe el archivo existente.

Eliminación de archivos

Java permite eliminar un archivo cuando ya no es necesario, liberando espacio en el almacenamiento.

```
import java.io.File;

public class BorrarArchivo {
    public static void main(String[] args) {
        File archivo = new File("archivo.txt");
        if (archivo.delete()) {
            System.out.println("Archivo eliminado correctamente.");
        } else {
            System.out.println("No se pudo eliminar el archivo.");
        }
    }
}
```

Se utiliza el método delete() de la clase File. Devuelve true si el archivo se borró con éxito, false en caso contrario.



1.5 Almacenamiento y medios físicos

El almacenamiento de la información se refiere al proceso mediante el cual los datos son guardados de manera permanente o temporal en un dispositivo físico, permitiendo su conservación, recuperación y uso posterior. En el contexto de los sistemas informáticos, el almacenamiento constituye un elemento esencial, ya que posibilita que los archivos creados y manipulados por los programas permanezcan disponibles incluso después de que el equipo sea apagado.

Los medios físicos de almacenamiento son los dispositivos donde se registran los datos en forma de bits. Estos dispositivos pueden clasificarse de acuerdo con la tecnología que utilizan para conservar la información, destacándose principalmente los medios magnéticos, ópticos y de estado sólido. Cada uno presenta características particulares en cuanto a capacidad, velocidad, durabilidad y costo.

- Medios magnéticos

Los medios magnéticos almacenan la información mediante la orientación de partículas magnéticas sobre una superficie. Entre los dispositivos más representativos se encuentran los discos duros tradicionales (HDD). Estos dispositivos han sido ampliamente utilizados debido a su gran capacidad de almacenamiento y su costo relativamente accesible. Funcionan mediante platos giratorios y cabezales que leen y escriben datos magnéticamente. Aunque ofrecen grandes volúmenes de almacenamiento, su velocidad es menor en comparación con tecnologías más recientes y pueden ser más sensibles a golpes físicos.

- Medios ópticos



Los medios ópticos utilizan tecnología láser para leer y escribir información sobre una superficie reflectante. Ejemplos comunes incluyen los CD, DVD y Blu-ray. En estos dispositivos, los datos se almacenan en forma de pequeñas marcas que son interpretadas por el láser como información binaria. Aunque su uso ha disminuido con el avance de otras tecnologías, continúan siendo utilizados para distribución de software, almacenamiento multimedia y copias de seguridad. Su principal ventaja es la facilidad de transporte y el bajo costo, aunque su capacidad y velocidad son limitadas en comparación con otros medios.

- Medios de estado sólido

Los medios de estado sólido almacenan la información mediante circuitos electrónicos y memoria flash, sin partes mecánicas móviles. Entre ellos se encuentran las unidades SSD y las memorias USB. Estos dispositivos ofrecen mayor velocidad de acceso, menor consumo de energía y mayor resistencia a impactos físicos. Debido a estas características, se han convertido en una de las opciones más utilizadas en equipos modernos. Aunque históricamente su costo era más elevado que el de los discos duros magnéticos, actualmente su precio es cada vez más accesible.

En el ámbito de la programación, independientemente del medio físico utilizado, el sistema operativo actúa como intermediario entre el software y el hardware, gestionando el acceso a los dispositivos de almacenamiento. De esta manera, cuando un programa en Java crea o modifica un archivo, el desarrollador no necesita preocuparse directamente por el tipo de medio físico, ya que el sistema se encarga de administrar la escritura y lectura de los datos en el dispositivo correspondiente.



Conclusión

El estudio de los archivos y de los mecanismos que intervienen en su almacenamiento, organización y manipulación resulta esencial dentro del campo de la informática y la programación. A través de este trabajo se pudo comprender que un archivo no solo representa un conjunto de datos, sino también una herramienta fundamental para garantizar la persistencia de la información y facilitar el funcionamiento de los sistemas computacionales.

La clasificación de los archivos, su estructura interna y los diferentes métodos de organización —secuencial, aleatoria e indexada— permiten elegir el modelo más adecuado según las necesidades del sistema. Asimismo, el análisis de las operaciones básicas sobre archivos demuestra la importancia de controlar adecuadamente la creación, lectura, escritura, modificación y eliminación de los datos, asegurando así la integridad y consistencia de la información. Los ejemplos prácticos en Java evidencian cómo estos conceptos se aplican en un entorno de programación real.

Finalmente, la revisión de los medios físicos de almacenamiento permitió reconocer la evolución de las tecnologías empleadas para conservar los datos, destacando sus características, ventajas y limitaciones. Todo lo anterior reafirma que el manejo adecuado de archivos es una competencia indispensable para cualquier estudiante o profesional del área informática, ya que constituye la base para la construcción de aplicaciones robustas, organizadas y orientadas al manejo eficiente de la información.



Bibliografía

- Deitel, P. J., & Deitel, H. M. (2008). Cómo programar en Java (7.^a ed., traducción de A. V. R. Elizondo). México: Pearson Educación. (Edición autorizada de la obra en inglés Java™ How to Program, 7th Edition, Pearson Education, 2007). ISBN: 978-970-26-1190-5
- Deitel, P., & Deitel, H. (2017). Java: How to program (11th ed.). Pearson.
- Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems (7th ed.). Pearson.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating system concepts (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). Modern operating systems (4th ed.). Pearson.
- Sommerville, I. (2016). Software engineering (10th ed.). Pearson.
- Oracle. (2023). Java Platform, Standard Edition documentation. <https://docs.oracle.com/en/java/>