



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Sábado, 25/03/2023
Hora de inicio:	19:00

Introducción a la programación y computación 1 [D]

José Eduardo Morales García

Ejemplo de código para leer un archivo de texto

- El manejo de archivos es una de las tareas más comunes en la programación. En Java, existen varias clases y métodos para leer, escribir y manipular archivos de diferentes tipos, como texto, imágenes y datos binarios. En esta presentación, exploraremos las diferentes técnicas y herramientas que podemos utilizar para manejar archivos en Java.




Tipos de archivos en Java: En primer lugar, repasaremos los diferentes tipos de archivos que podemos manejar en Java, como archivos de texto, archivos de imagen y archivos binarios. Explicaremos las diferencias entre cada tipo y cuál es la mejor forma de manejarlos.



Clases para manejar archivos: A continuación, presentaremos las clases principales que podemos utilizar para manejar archivos en Java, como `File`, `FileReader`, `FileWriter`, `InputStream`, `OutputStream`, entre otras. Explicaremos cuándo y cómo utilizar cada una de estas clases.



Lectura y escritura de archivos de texto: En esta sección, nos enfocaremos en cómo leer y escribir archivos de texto en Java. Veremos ejemplos de cómo utilizar las clases `FileReader` y `FileWriter` para leer y escribir archivos de texto, y cómo manejar excepciones que pueden ocurrir durante la lectura o escritura de archivos.

- 
- ▶ Lectura y escritura de archivos binarios: En esta sección, nos enfocaremos en cómo leer y escribir archivos binarios en Java. Veremos ejemplos de cómo utilizar las clases `InputStream` y `OutputStream` para leer y escribir archivos binarios, y cómo manejar excepciones que pueden ocurrir durante la lectura o escritura de archivos.
 - ▶ Manipulación de archivos: Finalmente, presentaremos algunas técnicas para manipular archivos en Java, como copiar archivos, mover archivos, eliminar archivos y cambiar el nombre de archivos. Veremos ejemplos de cómo utilizar las clases `File` y otras clases auxiliares para llevar a cabo estas tareas.

Tipos de archivos en Java

- **Archivos de texto:** Contienen datos en formato de texto plano, como un archivo CSV o un archivo de configuración. Se pueden leer y escribir utilizando las clases `FileReader` y `FileWriter`.
- **Archivos de imagen:** Contienen imágenes en diferentes formatos, como JPEG, PNG o BMP. Se pueden leer y escribir utilizando las clases `ImageIO` y `BufferedImage`.
- **Archivos binarios:** Contienen datos en formato binario, como un archivo ejecutable o un archivo de sonido. Se pueden leer y escribir utilizando las clases `InputStream` y `OutputStream`.

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

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

Clases para manejar archivos en Java

- ▶ En Java, existen varias clases que nos permiten manejar archivos, tales como File, FileReader, FileWriter, InputStream y OutputStream.
- ▶ **File:** Representa un archivo o un directorio en el sistema de archivos. Proporciona métodos para crear, leer, escribir y manipular archivos y directorios.
- ▶ **FileReader** y **FileWriter:** Proporcionan una forma de leer y escribir archivos de texto.
- ▶ **InputStream** y **OutputStream:** Proporcionan una forma de leer y escribir archivos binarios.

```
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("El archivo ha sido creado");
            } else {
                System.out.println("El archivo ya existe");
            }
        } catch (IOException e) {
            System.out.println("Error al crear el archivo");
        }
    }
}
```


Lectura y escritura de archivos de texto

- Para leer y escribir archivos de texto en Java, podemos utilizar las clases **FileReader** y **FileWriter**. Estas clases proporcionan métodos para leer y escribir caracteres en un archivo de texto.

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

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

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

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



Parte práctica