

ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

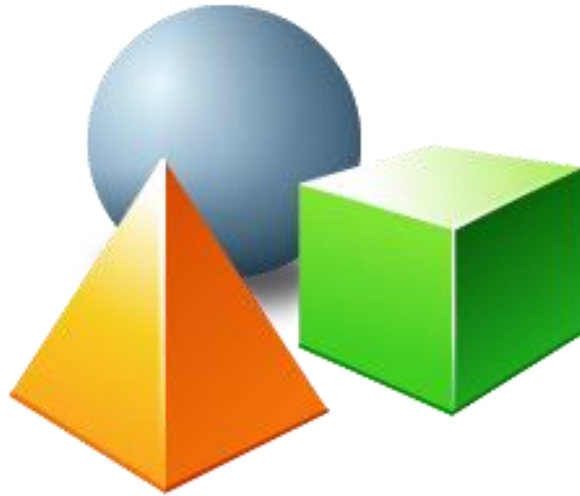
ARCHIVOS

Eric Gustavo Coronel Castillo

youtube.com/DesarrollaSoftware

gcoronelc@gmail.com





Temas

- Objetivo
- Almacenamiento secundario
- Archivos
- Archivos de texto
- Archivos binarios
- Proyecto ejemplo



OBJETIVO

Java proporciona un numero amplio de clases, muy útiles para trabajar sobre archivos a través del paquete java.io.

Utilizar archivos como elementos de almacenamiento secundario.



ALMACENAMIENTO SECUNDARIO

- La memoria de la computadora es considerada almacenamiento primario. El almacenamiento primario es volátil.
- Cuando queremos preservar la información es necesario guardar la información en un almacenamiento de tipo no volátil: este es el almacenamiento secundario.



La información se debe grabar en un
almacenamiento secundario.

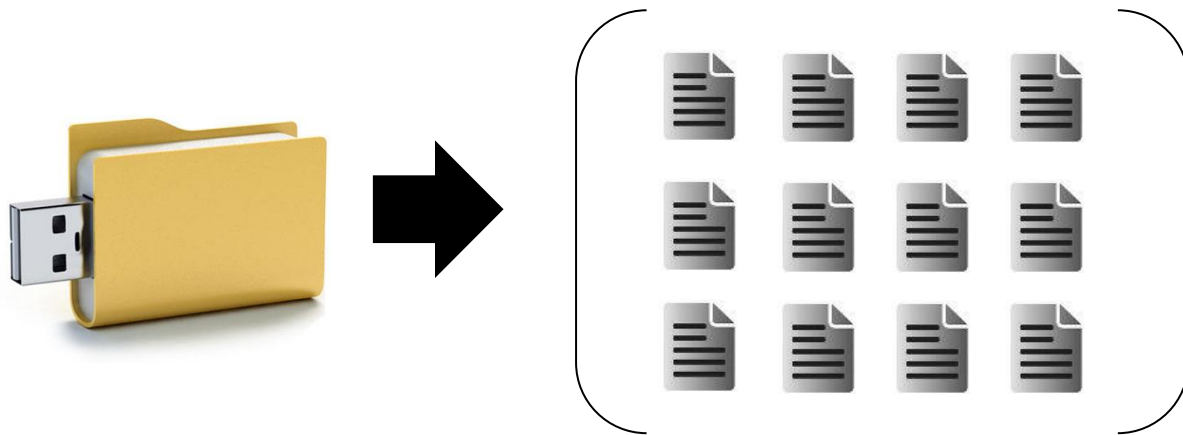




ARCHIVOS

ARCHIVOS DE TEXTO

- Son el almacenamiento secundarios por excelencia.
- Existen los siguientes tipos de archivos:
 - archivos de texto
 - archivos binarios secuenciales



ARCHIVOS DE TEXTO

- Son aquellos archivos que sólo almacenan datos de tipos carácter o cadenas de caracteres.
- Son utilizados para:
 - Información del tipo registro.
 - Guardar información que no necesita de un formato específico.



```
xxxxxxx xxx xxxx  
xxxx xxxx xxx xx  
xxxxxxx xxxx xxxx  
xxx xxxxx xxxxxx  
xxxxxxx xxxx xxxx  
xxxx xxxx xxxxxx  
xxxxxxx xxxx xxxx  
xxxx xxxx xxxxxx
```



ARCHIVOS BINARIOS

- Un archivo binario contiene información de cualquier tipo codificada en binario para el propósito de almacenamiento y procesamiento.
- Los datos se guardan de manera secuencial, es decir, uno detrás de otro.





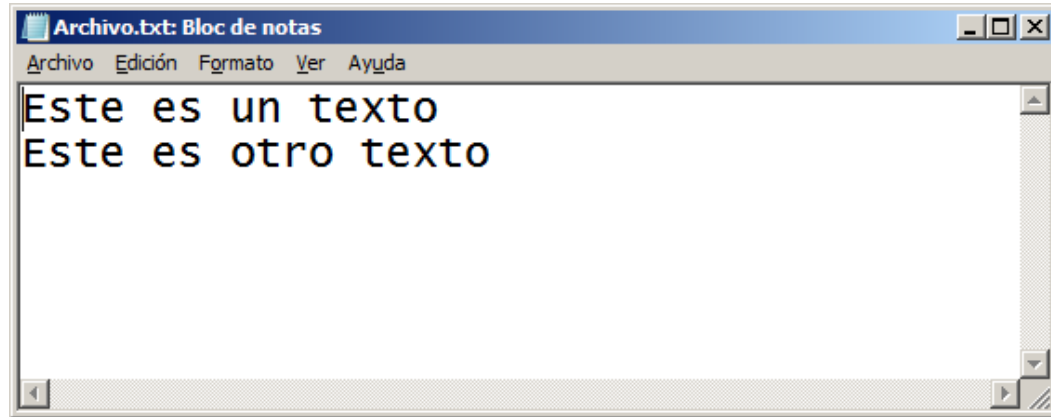
ARCHIVOS DE TEXTO

- Crear un archivo de texto

```
public static void main(String[] args) {  
    PrintStream printStream = null;  
    try {  
        String fileName = "D:\\\\Archivo.txt";  
        printStream = new PrintStream(fileName);  
        printStream.println("Este es un texto");  
        printStream.println("Este es otro texto");  
    } catch (Exception e) {  
        System.err.println(e);  
    } finally {  
        try {  
            printStream.close();  
        } catch (Exception e) { }  
    }  
}
```


ARCHIVOS DE TEXTO

- Crear un archivo de texto



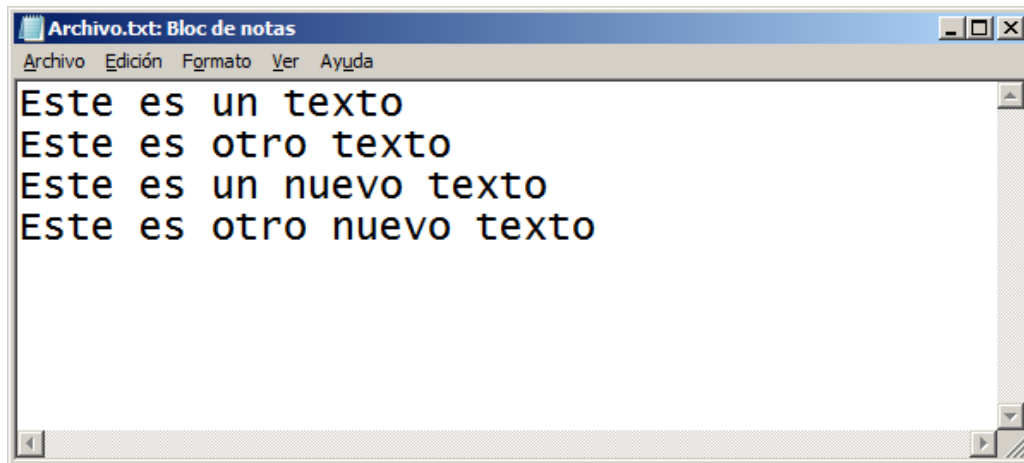
- Agregar líneas a un archivo de texto

```
public static void main(String[] args) {  
    PrintStream printStream = null;  
    try {  
        String fileName = "D:\\\\Archivo.txt";  
        FileOutputStream stream = new FileOutputStream(fileName, true);  
        printStream = new PrintStream(stream);  
        printStream.println("Este es un nuevo texto");  
        printStream.println("Este es otro nuevo texto");  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } finally {  
        try {  
            printStream.close();  
        } catch (Exception e) { }  
    }  
}
```



ARCHIVOS DE TEXTO

- Agregar líneas a un archivo de texto

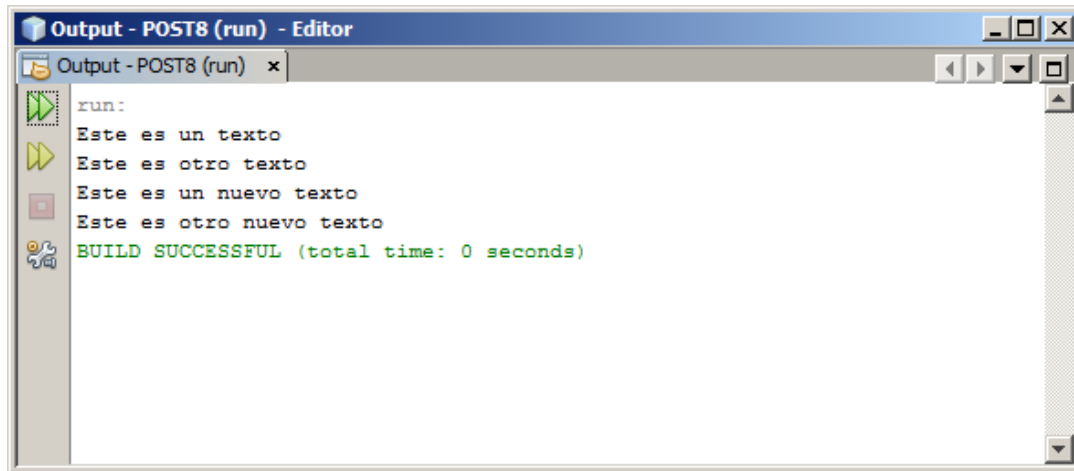


- Leer un archivo de texto

```
public static void main(String[] args) {  
    FileReader fileReader = null;  
    try {  
        String fileName = "D:\\\\Archivo.txt";  
        fileReader = new FileReader(fileName);  
        BufferedReader bufferedReader = new BufferedReader(fileReader);  
        String linea = bufferedReader.readLine();  
        while (linea != null) {  
            System.out.println(linea);  
            linea = bufferedReader.readLine();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        try {  
            fileReader.close();  
        } catch (Exception e) { }  
    }  
}
```

ARCHIVOS DE TEXTO

- Agregar líneas a un archivo de texto





ARCHIVOS BINARIOS

- Crear un archivo binario

```
try {  
    OutputStream file = new FileOutputStream("C:\\\\Datos.dat");  
    OutputStream buffer = new BufferedOutputStream(file);  
    ObjectOutputStream output = new ObjectOutputStream(buffer);  
    output.writeObject(new Artículo("Art1",34.5));  
    output.writeObject(new Artículo("Art2",56.2));  
    output.writeObject(new Artículo("Art3",67.43));  
    output.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```



■ Leer un archivo binario

```
try {  
    InputStream file = new FileInputStream("C:\\\\Datos.dat");  
    InputStream buffer = new BufferedInputStream(file);  
    ObjectInput input = new ObjectInputStream(buffer);  
    Object aux = input.readObject();  
    while (aux != null) {  
        if (aux instanceof Articulo) {  
            System.out.println(aux);  
        }  
        aux = input.readObject();  
    }  
    input.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

PROYECTO EJEMPLO

- La empresa de estudio de mercado **Vía Éxitos** esta desarrollando un software para procesar las encuestas de sobre la preferencia de los limeños por las bebidas gaseosas.
- Usted debe desarrollar una aplicación en Java que registre en un archivo de texto las encuestas y debe generar un archivo con el resultado obtenido.
- También se pide el resultado en formato de grafico:
 - Tipo Pastel
 - Tipo Barras
- Las bebidas a estudiar son: Coca Cola, Inka Kola, Guarana, Big Cola, KR, Oro, Otro.



ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

Gracias

Eric Gustavo Coronel Castillo

youtube.com/DesarrollaSoftware

gcoronelc@gmail.com





Gustavo Coronel



gcoronelc.github.io

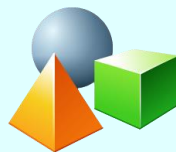
youtube.com/DesarrollaSoftware

facebook.com/groups/desarrollasoftware



FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos

JDBC



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC