

TALLER DE LENGUAJE DE PROGRAMACIÓN I

Entorno de CONSOLA y GRÁFICOS

UNIDAD 1

Escritura y Lectura de Archivos Planos Aleatorios

JAVA

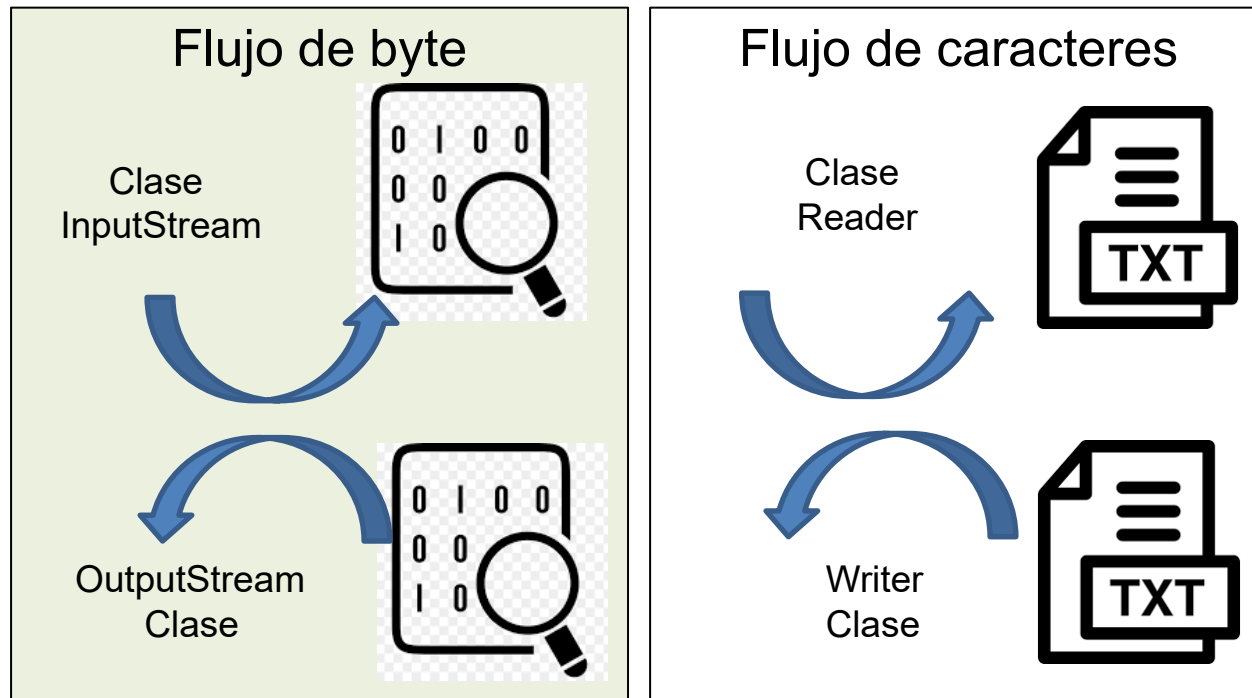
OBJETIVO

- Lectura y Escritura de Archivos Planos Aleatorios
 - Secuencias (STREAMS)
 - Flujo de byte
 - Flujo de caracteres
 - Manejo aleatorio de datos
 - Clase RandomAccessFile Flujo de byte

Trabajo Independiente

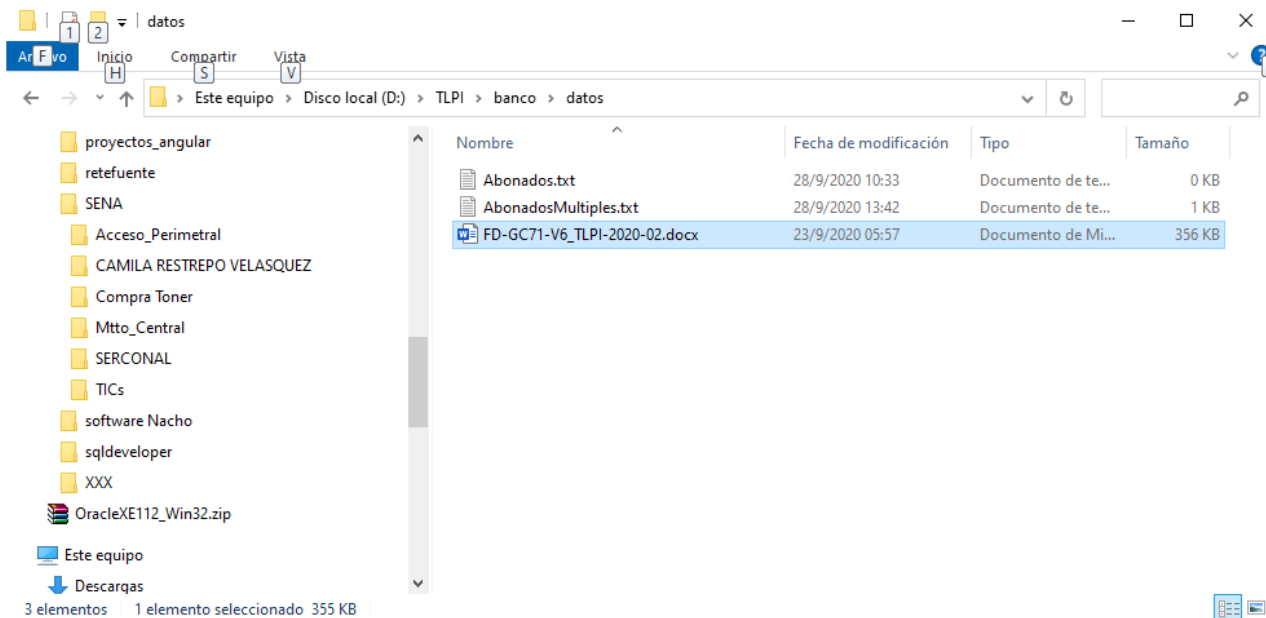
- Ejercicio

Secuencias (STREAMS)



Lectura y Escritura de Archivos Planos Secuenciales

Definimos un archivo de datos para leer



Lectura y Escritura de Archivos Planos Secuenciales

Definimos la clase que escriba el archivo

```
package planos_binarios;
import java.io.*;
public class LeerArchivo
{
    public LeerArchivo()
    {
        try
        {
            FileInputStream archivo_entrada = new FileInputStream("D:\\TLPI\\banco\\datos\\FD-GC71-V6_TLPI-2020-02.docx");
            boolean final_archivo = false;
            int contar = 0;
            int[] ar_salida = new int[364217];
            while(!final_archivo) {
                int byte_entrada = archivo_entrada.read();
                if(byte_entrada == -1) {
                    final_archivo = true;
                }
                else{
                    System.out.println(byte_entrada);
                    ar_salida[contar] = byte_entrada;
                    contar++;
                }
            }
            System.out.println(contar);
            archivo_entrada.close();
            CopiarArchivo copiar = new CopiarArchivo(ar_salida);
        }catch(IOException e) {}
    }
}
```

Lectura y Escritura de Archivos Planos Secuenciales

Definimos la clase que copia el archivo

```
package planos_binarios;
import java.io.*;
public class CopiarArchivo
{
    public CopiarArchivo(int[] copia_archivo)
    {
        try
        {
            FileOutputStream archivo_nuevo = new FileOutputStream("D:\\TLPI\\banco\\datos\\FD-GC71-V6_TLPI-2020-02_copia.docx");
            for(int i = 0 ;i < copia_archivo.length; i++ )
            {
                archivo_nuevo.write(copia_archivo[i]);
            }
            archivo_nuevo.close();
        }catch(IOException e) {}
    }
}
```

Lectura y Escritura de Archivos Planos Secuenciales

Definimos la clase principal

```
package planos_binarios;

public class inicio
{
    public static void main(String[] args)
    {
        LeerArchivo leer = new LeerArchivo();
    }
}
```

Clase RandomAccessFile

A diferencia de las clases Reader, Writer, InputStream y OutputStream, que leen líneas enteras byte a byte, la clase **RandomAccessFile** permite acceder a una línea específica y a una ubicación específica de la línea, funcionalidad que permite manejar el concepto de Bases de Datos en archivos planos.

Lectura y Escritura de Archivos Planos Aleatorias

Ejercicio con la Clase RandomAccessFile

Desarrollar una aplicación que permita registrar, buscar, eliminar y listar a las personas registradas en un archivo plano, para el caso de la búsqueda, se debe hacer por la identificación de la persona.

Los datos que se capturaran son:

Identificación de tipo entero.

Nombre completo de tipo alfanumérico

Sexo de tipo carácter.

Lectura y Escritura de Archivos Planos Aleatorias

Paso 1. Crear el archive de almacenamiento de datos

- Creamos un proyecto llamado plano_aleatorios
- Creamos un paquete llamado plano_aleatorios
- Creamos una clase principal donde se cree el archivo y muestre el menú.

Menú principal

1. Registrar
2. Modificar
3. Eliminar
0. Salir

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

Algunos de los métodos de la clase RandomAccessFile

- Método `getFilePointer()`
 - Devuelve la posición actual del puntero del archivo
- Método `seek()`
 - Coloca el puntero del archivo en una posición determinada
- `length()`
 - Devuelve la longitud del archivo

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

El almacenamiento de datos se realiza con un número de byte dado

- String → 1 byte por cada carácter.
 - int → 4 byte
 - double → 8 byte
- etc.

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

Tener presente el movimiento del puntero

- Cada vez que escribe un dato o lee, el puntero se mueve.
- El movimiento del puntero depende de los tipos de datos

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

Métodos para escribir en un archivo

- byte → `writeByte(int i)`
- int → `writeInt(int i)`
- float → `writeFloat(float i)`
- double → `writeDouble(double i)`
- String → `writeUTF(String s)`

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

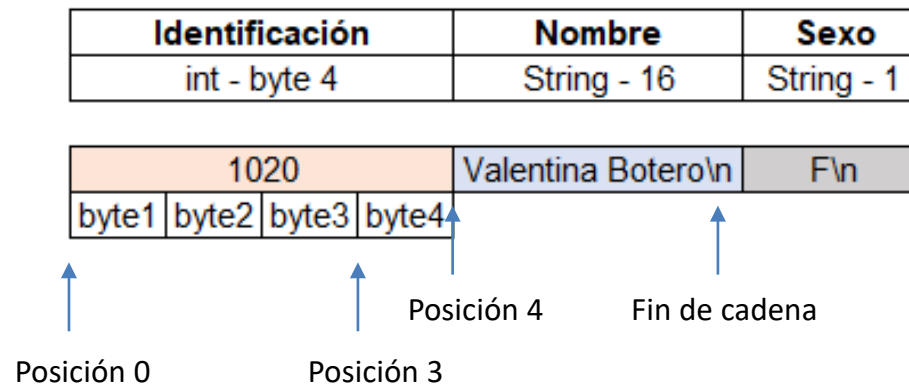
Métodos para leer un archivo

- byte → `readByte()`
- int → `readInt()`
- long → `readLong()`
- float → `readFloat()`
- double → `readDouble()`
- String → `readLine()` , Lee hasta encontrar `\n`

Lectura y Escritura de Archivos Planos Aleatorias

Paso 2. Acceso al archivo de datos

Concepto de almacenamiento



Lectura y Escritura de Archivos Planos Aleatorias

Código del ejemplo – Clase Escribir Registros

```
package plano_aleatorios;

import java.io.*;

public class EscribirRegistros
{
    public EscribirRegistros(int wid, String wnombre, String wsexo )
    {
        try
        {
            RandomAccessFile archivo = new
RandomAccessFile("D:\\TLPI\\planos_aleatorios\\datos.txt","rw");
            archivo.seek(archivo.length());

            archivo.writeInt(wid);
            archivo.writeUTF(wnombre);
            archivo.writeUTF(wsexo);

            archivo.close();
        }catch(IOException e) {System.out.print(e);
        }
    }
}
```

Lectura y Escritura de Archivos Planos Aleatorias

Código del ejemplo – Clase Buscar Registros

```
package plano_aleatorios;
import java.io.*;
public class BuscarRegistro
{
    public BuscarRegistro() {}

    public String EncontrarRegistroIdentificacion(int wid)
    {
        String retorno="";
        try
        {
            RandomAccessFile archivo = new RandomAccessFile("D:\\TLP\\planos_aleatorios\\datos.txt","rw");
            archivo.seek(0);
            while(archivo.getFilePointer() < archivo.length())
            {
                if(archivo.readInt() == wid)
                {
                    retorno = archivo.readUTF();
                    archivo.seek(archivo.length());
                }
                else
                {
                    retorno = "No lo encontro";
                    retorno = archivo.readUTF();
                    retorno = archivo.readUTF();
                }
            }
            archivo.close();
        }catch(IOException e) {System.out.print(e);}
        return retorno;
    }
}
```

Lectura y Escritura de Archivos Planos Aleatorias

Código del ejemplo – Clase Listar Registros

```
package plano_aleatorios;
import java.io.*;
public class ListarRegistros
{
    public ListarRegistros()
    {
        String wcadena="";
        try
        {
            RandomAccessFile archivo = new RandomAccessFile("D:\\TLPI\\planos_aleatorios\\datos.txt","r");
            archivo.seek(0);
            while(archivo.getFilePointer() < archivo.length())
            {
                wcadena = archivo.readInt() + " ";
                wcadena += archivo.readUTF() + " ";
                wcadena += archivo.readUTF() + " ";
                System.out.println(wcadena);
            }
            archivo.close();
        }catch(IOException e) {System.out.print(e);}
    }
}
```

Lectura y Escritura de Archivos Planos Aleatorias

Código del ejemplo – Clase Principal parte 1

```
package plano_aleatorios;
import java.util.Scanner;
public class inicio
{
    public static void main(String[] args)
    {
        Scanner pedir = new Scanner(System.in);
        int opcion;
        System.out.print("1. Registrar\n2. Buscar\n3. Listar\n0. Salir ");
        System.out.println();
        System.out.println("Selecciones la opción:");
        int identificacion;
        String nombre;
        String sexo;
        do{
            opcion = pedir.nextInt();
            switch(opcion)
            {
                case 1:
                    System.out.println("Escriba la Identificación");
                    identificacion = pedir.nextInt();
                    System.out.println("Escriba el nombre");
                    nombre = pedir.next();
                    System.out.println("Escriba el Sexo");
                    sexo = pedir.next();
                    System.out.println("Termino de Capturar datos");
                    System.out.println("Menú Principal");
                    System.out.print("1. Registrar\n2. Buscar\n3. Listar\n0. Salir ");
                    new EscribirRegistros(identificacion,nombre,sexo);
                    break;
```

Lectura y Escritura de Archivos Planos Aleatorias

Código del ejemplo – Clase Principal parte 2

```
tiene como nombre " + nombre);

        case 2:
            System.out.println("Escriba la Identificación buscada");
            identificacion = pedir.nextInt();

            BuscarRegistro encontro = new BuscarRegistro();
            nombre = encontro.EncontrarRegistroIdentificacion(identificacion);
            System.out.println("El registro con Identificación: " + identificacion + "

        case 3:
            new ListarRegistros();
            System.out.println("Menú Principal");
            System.out.print("1. Registrar\n2. Buscar\n3. Listar\n0. Salir ");
            break;

        case 0:
            System.out.println("Gracias por usar nuestros servicios, vuelva

        default:
            System.out.println("Opción no valida, intente de nuevo.");
            System.out.print("1. Registrar\n2. Buscar\n3. Listar\n0. Salir ");

        }while (opcion != 0);
        pedir.close();
    }
}
```

Trabajo Independiente

- Desarrollar la opción de modificar registros