



Arreglos y archivos

Archivos, arrays y Strings

***Implementar una aplicación
básica de consola utilizando
las buenas prácticas y
convenciones para resolver
un problema de baja
complejidad acorde al
lenguaje Java***

- Unidad 1: Flujo, ciclos y métodos
- Unidad 2: Arreglos y archivos
- Unidad 3: Programación orientada a objetos
- Unidad 4: Pruebas unitarias y TDD



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Aplicar los métodos de la clase File.*

¿Qué entendemos por
archivos en
programación?



**/* Métodos para facilitar
el trabajo con archivos */**

String y Join

// String

```
String letras = "a,b,c,d,e,f";  
ArrayList<String> letrasSeparadas = new  
ArrayList<String>(Arrays.asList(letras.split(",")));  
System.out.println(letrasSeparadas);
```

```
// [a, b, c, d, e, f]
```

// Join

```
String letrasJuntas = String.join(" -> ",letrasSeparadas);  
System.out.println(letrasJuntas);
```

```
// a -> b -> c -> d -> e -> f
```

Crear archivo

En IDE Eclipse

En Eclipse:

1. Clic derecho sobre el proyecto
2. New
3. File
4. Ingresamos el nombre del archivo

Dentro del archivo escribiremos la siguiente información:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Abrir archivo

Librería FileReader

```
String nombre = "data";
try {
    FileReader fr = new FileReader(nombre);
    //código donde haremos lectura del archivo

    fr.close();
}
catch (Exception e){
    System.out.println("Excepcion leyendo fichero "+ nombre + ": " + e);
}
```


Leer archivo

Usando método read();

```
String nombre = "data";
try {
    FileReader fr = new FileReader(nombre);
    int data=fr.read();
    String lectura = "";
    while(data != -1) {
        lectura += (char)data;
        data = fr.read(); //leemos el siguiente caracter
    }
    System.out.println(lectura); //1,2,3,4,5,6,7,8,9,10
    fr.close();
}
catch (Exception e){
    System.out.println("Excepcion leyendo fichero "+ nombre + ": " + e);
}
```

¿Qué pasa si queremos
leer múltiples líneas y
almacenarlas de manera
separada?



Leer archivo por líneas

Agregamos la librería `BufferedReader`

```
String nombre = "data";
try {
    FileReader fr = new FileReader(nombre);
    BufferedReader br = new BufferedReader(fr);

    String data = br.readLine();
    while(data != null) {//retorna -1 cuando no hay más caracteres por leer
        System.out.println(data); //1,2,3,4,5,6,7,8,9,10
        data = br.readLine();
    }
    br.close();
    fr.close();
}
catch (Exception e){
    System.out.println("Excepcion leyendo fichero "+ nombre + ": " + e);
}
```

Transformando los datos a un array

```
String nombre = "data";
FileReader fr = null;
BufferedReader br = null;
String data = "";
ArrayList<String> numerosString;
ArrayList<Integer> numeros = new ArrayList<Integer>();
try {
    fr = new FileReader(nombre);
    br=new BufferedReader(fr);
    data = br.readLine();
    numerosString = new ArrayList<String>(Arrays.asList(data.split(",")));
    for(String temp : numerosString) {
        numeros.add(Integer.parseInt(temp));
    }
    System.out.println(numeros);
    br.close();
    fr.close();
}
catch (Exception e){
    System.out.println("Excepcion leyendo fichero "+ nombre + ": " + e);
}
```

Leyendo múltiples líneas

```
// data2
```

```
21  
10  
6  
9  
11  
0  
2  
3  
50
```

```
String nombreArchivo = "data2";  
FileReader fr = null;  
BufferedReader br = null;  
String data = "";  
ArrayList<Integer> numeros = new ArrayList<Integer>();  
try {  
    fr = new FileReader(nombreArchivo);  
    br=new BufferedReader(fr);  
  
    data = br.readLine();  
    while (data != null) {  
        numeros.add(Integer.parseInt(data));  
        data = br.readLine();  
    }  
    br.close();  
    fr.close();  
}  
catch (Exception e){  
    System.out.println("Excepcion leyendo fichero "+ nombreArchivo + ": " + e);  
}  
System.out.println(numeros); //[21, 10, 6, 9, 11, 0, 2, 3, 50]
```

Ejercicio: Dado el data2

Se pide crear un programa que tome los datos de ese archivo y construya un arreglo con los mismos, pero transformando todos los valores mayores de 20 a un máximo de 20.

```
// Solución
```

```
String nombre = "data2";  
ArrayList<Integer> numeros = readFile(nombre);  
int i;  
for(i=0;i<numeros.size();i++) {  
    if(numeros.get(i) >20) {  
        numeros.set(i, 20);  
    }  
}  
System.out.println(numeros);
```

Guardando los resultados

Incorporando las librerías `FileWriter` y `PrintWriter`.

Creamos un método para escribir el archivo.

```
void writeFile(String nombreArchivo, ArrayList<Integer> numeros) {  
    FileWriter archivo = null;  
    PrintWriter pw = null;  
    try {  
        archivo = new FileWriter(nombreArchivo);  
        pw = new PrintWriter(archivo);  
        int i;  
        for(i = 0 ; i<numeros.size() ; i++) {  
            pw.println(numeros.get(i));  
        }  
        pw.close();  
        archivo.close();  
    }  
    catch(Exception e){  
        System.out.println("Fichero " + nombreArchivo + "no se pudo crear" + e);  
    }  
}
```

¡Importante!

- Si el archivo no existe, lo creará
- Si el archivo existe, lo sobrescribirá

Para solucionarlo:

```
FileWriter archivo = new FileWriter(nombre,true);
```


`/* Problema con dos array */`

Ejercicio

Algoritmo

- El usuario ingresa un String
- Buscamos el índice en el arreglo de alumnos de ese String
- Si existe el índice: buscamos la nota dentro del arreglo de notas
- Si no: mostramos que no pudimos encontrar la nota.

```
ArrayList<Integer> notas = new ArrayList<Integer>(Arrays.asList(5,9,6,8,4));  
ArrayList<String> alumno = new  
ArrayList<String>(Arrays.asList("Julia", "María", "Teresa", "Diego", "Pedro"));
```

Ejercicio

Algoritmo

```
Scanner sc = new Scanner(System.in);

ArrayList<Integer> notas = new ArrayList<Integer>(Arrays.asList(5,9,6,8,4));
ArrayList<String> alumnos = new
ArrayList<String>(Arrays.asList("Julia", "María", "Teresa", "Diego", "Pedro"));

String nombreABuscar = sc.nextLine();
int indice = alumnos.indexOf(nombreABuscar);
if(indice != -1)
    System.out.printf ("La nota de %s es: %d\n", nombreABuscar, notas.get(indice));
else
    System.out.println("Alumno no encontrado");
```

Versión con métodos

Se nos pide construir un método que reciba los arrays, el nombre de la persona y de como resultado su nota. En caso de no existir, retornar -1.

```
static int busqueda(ArrayList<Integer> notas, ArrayList<String> alumnos, String nombre) {  
  
    int indice = alumnos.indexOf(nombre);  
    if(indice != -1)  
        return notas.get(indice);  
    else  
        return -1;  
}
```

Operando sobre valores en dos arrays

Ejemplo de ventas de una empresa: datos de ventas de una empresa en un arreglo y la de la otra empresa en el otro arreglo. Donde cada posición indica las ventas diarias:

```
ArrayList<Integer> v1 = new ArrayList<Integer>(Arrays.asList(100,20,50,70,90));  
ArrayList<Integer> v2 = new ArrayList<Integer>(Arrays.asList(150,30,50,20,30));
```

¿Cómo podemos obtener las ventas diarias de las 2 tiendas en conjunto? ¿Por elemento o por índice?

```
ArrayList<Integer> vt = new ArrayList<Integer>();  
int n = v1.size();  
int i;  
for(i=0;i<n;i++){  
    vt.add(v1.get(i)+v2.get(i));  
}  
System.out.println(vt);
```

Ejemplo



Torneo

Construir un listado de todos contra todos

Por ejemplo, si hay 3 equipos: e1, e2, e3, sería:

e1	v.s	e2
----	-----	----

e1	v.s	e3
----	-----	----

e2	v.s	e1
----	-----	----

e2	v.s	e3
----	-----	----

e3	v.s	e1
----	-----	----

e3	v.s	e2
----	-----	----



Torneo

Solución 1

```
ArrayList<String> a = new ArrayList<String>(Arrays.asList("Equipo 1","Equipo 2","Equipo 3","Equipo 4","Equipo 5"));
ArrayList<String> b = new ArrayList<String>(Arrays.asList("Equipo 1","Equipo 2","Equipo 3","Equipo 4","Equipo 5"));
ArrayList<String> t = new ArrayList<String>();

for(String tempA : a){
    for(String tempB : b){
        t.add(tempA + " v.s " + tempB);
    }
}

System.out.println(String.join("\n",t));
```



Torneo

Solución 2: Eliminando repetidos

```
ArrayList<String> a = new ArrayList<String>(Arrays.asList("Equipo 1","Equipo 2","Equipo 3","Equipo 4","Equipo 5"));
ArrayList<String> b = new ArrayList<String>(Arrays.asList("Equipo 1","Equipo 2","Equipo 3","Equipo 4","Equipo 5"));
ArrayList<String> t = new ArrayList<String>();

for(String tempA : a){
    for(String tempB : b){
        if(tempA != tempB){
            t.add(tempA + " v.s " + tempB);
        }
    }
}

System.out.println(String.join("\n",t));
```

Torneo

Solución 3

```
ArrayList<String> a = new ArrayList<String>(Arrays.asList("Equipo 1","Equipo 2","Equipo 3","Equipo 4","Equipo 5"));
ArrayList<String> t = new ArrayList<String>();

for(String tempA : a){
    for(String tempB : a){
        if(tempA != tempB){
            t.add(tempA + " v.s " + tempB);
        }
    }
}

System.out.println(String.join("\n",t));
```

/* Arrays dentro de arrays */

¿Qué son los arrays?

Introducción

Los arrays pueden contener otros arrays en su interior:

Veremos cómo operar sobre ellos utilizando:

- Arreglos estáticos
- Arreglos dinámicos

Arreglos estáticos

- Primer par de corchetes: indica las filas que tendrá el arreglo
- Segundo par de corchetes: indica la cantidad de columnas

```
int[][] a = new int [4][2];  
a[0][0] = 4;  
System.out.println(a[0][0]);
```

Iterando un arreglo estático

```
int [][] a = {{1,2,3},{4,5,6}};  
int i,j;  
for(i=0;i<2;i++){  
    for(j=0;j<3;j++) {  
        System.out.printf("%d ", a[i][j]);  
    }  
    System.out.printf("\n");  
}
```

Tamaño del array

a.length nos retorna 3, que corresponde a los 3 sub arreglos que tenemos, y:

- a[0].length retorna 3
- a[1].length retorna 2
- a[2].length retorna 5 que corresponde a la cantidad de elementos de cada subarreglo.

```
int[][] a = {{2, 4, 1}, {6, 8}, {7, 3, 6, 5, 1}};  
int i,j;  
for(i=0;i<a.length;i++){  
    for(j=0;j<a[i].length;j++) {  
        System.out.printf("%d ", a[i][j]);  
    }  
    System.out.printf("\n");  
}
```

Creando arreglos estáticos de dimensiones variadas

```
int[][] arrayDeArray = new int [4][2]; // Creamos que el arreglo tendrá 4 filas, pero todas de tamaño 2.

int num[][] = new int[3][];
num[0] = new int[3];
num[1] = new int[2];
num[2] = new int[5];

for(i=0;i<num.length;i++){
    for(j=0;j<num[i].length;j++) {
        System.out.printf("%d ", num[i][j]);
    }
    System.out.printf("\n");
}
```


¿Cuál es la diferencia entre
arreglos estáticos y
dinámicos?





Próxima sesión...

- *Desafío*

{desafío}
latam_

*Academia de
talentos digitales*

