

POO 4

Juan Carlos Figueroa Duran

Otoño-2015

Universidad del Bío-Bío
Departamento de Ciencias de la Computación y Tecnología de la
Información

1 Lectura y escritura de archivos

Entrada y salida estandar

Lectura: Define variables para lectura desde el archivo "entrada.txt"

Opción 1:

```
1 BufferedReader archEntrada = new BufferedReader(new FileReader("
    entrada.txt"));
```

Opción 2:

```
1 Scanner archivo = new Scanner(new FileReader("entrada.txt"));
```

Escritura: Define variable para grabar en el archivo "resultado.txt"

Opción 1:

```
1 BufferedWriter archSalida = new BufferedWriter(new FileWriter("
    resultado.txt"));
```

Opción 2:

```
1 Printstream archSalida = new PrintStream("resultado.txt");
```

Ejemplo de lectura

```
1 import java.io.BufferedReader;
2 import java.io.FileNotFoundException;
3 import java.io.FileReader;
4 import java.io.IOException;
5 public class LeerArchivo{
6     public static void main(String[] args) {
7         try {
8             //El nombre del archivo puede especificar una ruta
9             String nombreArchivo = "archivo.txt";
10            FileReader fr = new FileReader(nombreArchivo);
11            BufferedReader bf = new BufferedReader(fr);
12            String linea;
13            //Se debe verificar si se llego al final del archivo, linea != null
14            while ((linea = bf.readLine())!=null) {
15                System.out.println(linea);
16            }
17        } catch (FileNotFoundException fnfe){
18            fnfe.printStackTrace();
19        } catch (IOException ioe){
20            ioe.printStackTrace();
21        }
22    }
23 }
```

Ejemplo de escritura

```
1 import java.io.*;
2 public class EscribirArchivo {
3     public static void main(String[] args) {
4
5         //El nombre del archivo puede especificar una ruta
6         String nombreArchivo = "archivo.txt";
7         File arch = new File(nombreArchivo);
8         // Revisar si existe el archivo
9         if (arch.exists())
10            System.out.println("El archivo " + nombreArchivo + " ya existe");
11         else {
12             try{
13                 BufferedWriter bw =
14                     new BufferedWriter(new FileWriter(nombreArchivo));
15                 for (int x=0;x<10;x++){
16                     bw.write("Fila numero " + x + "\n");
17                 }
18                 bw.close();
19             } catch (IOException ioe){
20                 ioe.printStackTrace();
21             }
22         }
23     }
24 }
```

```
1 bw.close();
```

- Es una instrucción importante al momento de realizar una escritura dentro de un archivo. Ya que depende del finalizar la escritura del archivo en disco.
- Una vez terminado todos los cambios realizados al archivo, se debe invocar a al método **close()** de **BufferWriter** para que este pueda enviar todo el flujo almacenado en el buffer al archivo residente en disco.