

## Tarea 1

Las llamadas al sistema se encuentran marcadas en el siguiente ejemplo de un programa que convierte un número decimal un número hexadecimal y viceversa.

```
1  import java.io.*;
2  import java.util.Scanner;
3
4  //Creando mi propias excepciones
5  //Excepción que indica que todas son Letras
6  class ContainsLettersException extends Exception{ ...
10 }
11
12 //Excepción que indica que no es un numero hexadecimal
13 class InvalidHexException extends Exception{ ...
17 }
18
19 //Excepción que indica que se ingreso un número negativo
20 class NegativeValueEnteredException extends Exception{ ...
24 }
25
26
111 class Conversiones{ ...
134 }
135
136 public class Main{
137     public static void main(String[] args){
138         int d=5;
139         String linea = new String();
140         String entrada = new String();
141         Scanner sc = new Scanner(System.in); <--Llamada al sistema: Obtiene los datos de entrada.
142         File fichero = null;
143         FileReader archivo = null;
144         BufferedReader buffer = null;
145
146         Conversiones c = new Conversiones(d, entrada);
147
148         while(sc.hasNext()){
149             linea = sc.nextLine();
150         }
151         sc.close();
152
153         if("input.txt".equals(linea)){
154             try{
155                 fichero = new File("input.txt"); <--Llamada al sistema: Manipulación de archivo. Crea un archivo
156                 archivo = new FileReader(fichero); <--Llamada al sistema: Manipulación de archivo. Lee un archivo
157                 buffer = new BufferedReader(archivo);
158                 while((entrada = buffer.readLine()) != null){
159                     c.convertir(entrada);
160                 }
161             }catch(Exception e){
162                 e.printStackTrace(); <--Llamada al sistema: Obtiene información acerca de donde ha generado la excepción
163             }finally{
164                 try{
165                     if(archivo != null) <--Llamada al sistema: Se obtiene información acerca del status del archivo
166                         archivo.close(); <--Llamada al sistema: Manipulación de archivo. Cierra un archivo
167                 }catch(IOException e){ <--Llamada al sistema: Se obtiene información acerca del status del archivo
168                     System.out.println(e.getMessage());
169                 }
170             }
171         }
```

```
171
172 }else{
173     try{
174         fichero = new File("input2.txt"); <--Llamada al sistema: Manipulación de archivo. Crea un archivo
175         archivo = new FileReader(fichero); <--Llamada al sistema: Manipulación de archivo. Lee un archivo
176         buffer = new BufferedReader(archivo);
177         while((entrada = buffer.readLine()) != null){
178             c.convertir(entrada);
179         }
180     }catch(Exception e){ <--Llamada al sistema: Obtiene información acerca de donde ha generado la excepción
181         e.printStackTrace();
182     }finally{
183         try{
184             if(archivo != null)
185                 archivo.close(); <--Llamada al sistema: Manipulación de archivo. Cierra un archivo
186             }catch(IOException e){ <--Llamada al sistema: Se obtiene información acerca del status del archivo
187                 System.out.println(e.getMessage());
188             }
189         }
190     }
191 }
192
193
194
```

Una llamada al sistema es un método o función que puede invocar un proceso para solicitar un cierto servicio al sistema operativo.<sup>1</sup>

La manipulación de un archivo implica una llamada al sistema, ya que el sistema operativo tiene que parar los procesos que está realizando para acceder a un recurso, en este caso, para acceder a un archivo y poder leerlo, crearlo, escribir sobre él o borrarlo.

También, considero que las excepciones involucran una llamada al sistema, ya que las excepciones son un tipo de interrupción que notifica al sistema operativo que algo ha salido mal, por lo que el S.O. tiene que terminar la ejecución de los procesos que están implicados a consecuencia de la excepción.

De igual manera, considero que pedir datos de entrada con la función `new Scanner(System.in)` (en la línea 11 de código) también involucra una llamada al sistema porque los procesos que se están ejecutando en el momento en el que se le pide algún dato al usuario, tienen que parar y esperar hasta que el usuario ingrese un dato.

---

<sup>1</sup> Universidad de Sevilla. (19 de febrero de 2019). *Llamadas al sistema*. Sistemas Operativos. Recuperado de [https://1984.lsi.us.es/wiki-ssoo/index.php/Llamadas\\_al\\_sistema](https://1984.lsi.us.es/wiki-ssoo/index.php/Llamadas_al_sistema)