

## Ejercicio 01.10

Para este trabajo he desarrollado un programa en Java cuyo objetivo es realizar una conversión básica de un archivo escrito en C# a su equivalente en Java. La idea no es construir un traductor completo entre ambos lenguajes, ya que eso sería un proyecto muy complejo, sino aplicar las transformaciones necesarias para que el fichero proporcionado como ejemplo pueda adaptarse y ejecutarse en Java con una sintaxis correcta.

El programa comienza leyendo un fichero de entrada, en este caso el Program.cs (el que teníamos en Aules), y genera un fichero de salida llamado program.java. Para ello se emplean clases estándar de Java como `BufferedReader` y `PrintWriter`, que permiten manejar fácilmente la lectura y escritura de archivos de texto. Además, se añaden comprobaciones sobre los argumentos que recibe el programa para que el usuario pueda indicar desde la línea de comandos los nombres de los ficheros de entrada y salida.

```

1  import java.io.*;
2  import java.util.*;
3
4  public class Main {
5      public static void main(String[] args) {
6          if (args.length < 2) {
7              System.out.println("Uso: java Main <ficheroEntrada.cs> <ficheroSalida.java>");
8              return;
9          }
10
11          String ficheroEntrada = args[0];
12          String ficheroSalida = args[1];
13
14          try (BufferedReader br = new BufferedReader(new FileReader(ficheroEntrada));
15              PrintWriter pw = new PrintWriter(new FileWriter(ficheroSalida))) {
16
17              String linea;
18              while ((linea = br.readLine()) != null) {
19                  String convertida = convertirLinea(linea);
20                  pw.println(convertida);
21              }
22
23              System.out.println("Conversión completada. Archivo generado: " + ficheroSalida);
24          } catch (IOException e) {
25              e.printStackTrace();
26          }
27      }
28  }

```

En cuanto a las transformaciones implementadas, lo primero que se convierte son los `using` de C#, que se cambian por las sentencias `import` en Java.

```

private static String convertirLinea(String linea) { 1 usage
    // Imports y namespaces
    String original = linea;
    linea = linea.replace( target: "using System;", replacement: "import java.util.*;");
}

```

Después, se sustituyen los tipos de datos básicos: por ejemplo, string pasa a ser String con mayúscula inicial, y bool se reemplaza por boolean. También se adaptan las llamadas a consola, de manera que Console.WriteLine se convierte en System.out.println. Otro cambio importante es el método de lectura de teclado, ya que en C# se utiliza Console.ReadKey, mientras que en Java se emplea la clase Scanner para capturar la entrada.

```
// Tipos básicos
línea = línea.replace( target: "string", replacement: "String");
línea = línea.replace( target: "bool", replacement: "boolean");

// Console.WriteLine -> System.out.println
línea = línea.replace( target: "Console.WriteLine", replacement: "System.out.println");

// Console.ReadKey(true) -> new Scanner(System.in).next().charAt(0)
if (línea.contains("Console.ReadKey")) {
    línea = línea.replace( target: "var letra = Console.ReadKey(true);",
        replacement: "Scanner sc = new Scanner(System.in);\n        char letra = sc.next().charAt(0);");
}
```

Por otro lado, se han tenido en cuenta las convenciones de Java a la hora de nombrar métodos y de colocar las llaves de apertura de los bloques. En el código original los métodos aparecen con inicial mayúscula y las llaves suelen escribirse en una línea aparte. En el fichero resultante, los métodos se renombraron a minúsculas (por ejemplo, Ejercicio1 pasó a ejercicio1) y las llaves de apertura se mantienen en la misma línea de la declaración, siguiendo el estilo habitual de Java.

```
// Métodos y main en minúsculas
línea = línea.replace( target: "Main", replacement: "main");
línea = línea.replace( target: "static void Ejercicio1()", replacement: "static void ejercicio1()");
línea = línea.replace( target: "static void Ejercicio2()", replacement: "static void ejercicio2()");

// Llaves: en C# suelen ir en nueva línea, en Java las dejamos en la misma línea
línea = línea.replaceAll( regex: "\\s*\\{", replacement: "{");

return línea.equals(original) ? línea : línea;
}
```

El resultado es un fichero Java que mantiene la estructura del programa original, pero adaptado a las reglas del nuevo lenguaje. Aunque no se trata de un conversor genérico, cumple con la finalidad de transformar correctamente el ejemplo proporcionado, mostrando las diferencias fundamentales entre la sintaxis de C# y la de Java.

```
PS C:\Users\annaa\OneDrive\Escritorio\DAM2\Acceso a datos\Ejercicio 01.10\Ejercicio 01.10\src> java Main Program.cs program.java
Conversión completada. Archivo generado: program.java
```

### Código generado:

```
import java.util.*;

namespace Proyecto1
{
    class Program
    {
        static int numero;
        double numDecimal;
        boolean condicional;
        char letra;
        String cadena;

        static void ejercicio1()
        {
            int a = 5, b = 5;
            if(a>b)
            {
                System.out.println("A es mayor que B");
            }
            else
            {
                System.out.println("B es mayor o igual que A");
            }
        }
    }
}
```

```
        static void ejercicio1()
        {
            int a = 5, b = 5;
            if(a>b)
            {
                System.out.println("A es mayor que B");
            }
            else
            {
                System.out.println("B es mayor o igual que A");
            }
        }

        static void ejercicio2()
        {
            System.out.println("Introduce una letra");
            Scanner sc = new Scanner(System.in);
            char letra = sc.next().charAt(0);
            if (letra.KeyChar >= 'A' && letra.KeyChar <= 'Z')
            {
                System.out.println("La letra es mayúscula");
            }
            else
            {
                System.out.println("La letra no es mayúscula");
            }
        }
        static void main(String[] args)
        {
            Ejercicio1();
            Ejercicio2();
        }
    }
}
```