

APARTADO 1

Se nos pide crear una aplicación para generar un fichero de EMPLEADOS.DAT que contenga los datos de los empleados y basándonos en ese fichero creemos el xml con la clase DOM de la API de java.

Para ello he creado 3 clases una para empleados que contendrá los datos del mismo, una clase para los métodos para la gestión de entrada/salida de datos del fichero y una clase principal donde mostraremos un menú con 4 opciones:

```

Output - Tarea2AD (run) ... X
run:
-----
Menú de gestión del archivo EMPLEADOS.DAT
-----
1.-Introducir datos de empleado
2.-Mostrar datos de todos los empleados
3.-Generar fichero XML de empleados
4.-Salir
Elegir opción: |
  
```

1. Ingresar nuevo empleado y añadirlo al fichero o crearlo si no existe.

```

-----
Menú de gestión del archivo EMPLEADOS.DAT
-----
1.-Introducir datos de empleado
2.-Mostrar datos de todos los empleados
3.-Generar fichero XML de empleados
4.-Salir
Elegir opción: 1

Introduzca los datos del empleado
-----
Introduzca código
6
Introduzca nombre
Rosa
Introduzca dirección
Calle clavel
Introduzca salario
897,56
Introduzca comisión
234,5
Se han añadido los datos correctamente al fichero EMPLEADOS.DAT
  
```

La aplicación solicita los datos de empleado, los comprueba y los añade al fichero mostrándonos por pantalla un mensaje de confirmación

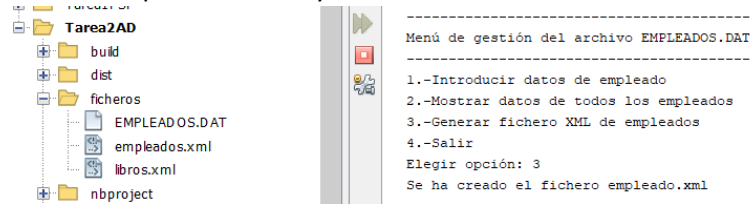
2. La segunda es mostrar el contenido del fichero por pantalla, aunque no lo pide la práctica nos sirve para visualizar el contenido sin tener que abrir el fichero.

```

-----
Menú de gestión del archivo EMPLEADOS.DAT
-----
1.-Introducir datos de empleado
2.-Mostrar datos de todos los empleados
3.-Generar fichero XML de empleados
4.-Salir
Elegir opción: 2
Código |Nombre |Dirección |Sueldo |Comisión
(,22,Daniel,Calle Santa Tecla,1256.89,2.4
%,11,Jose,Avenida Castilla,1078.45,4.6
!,13,Luis,Calle Kamela,2134.78,8.1
%,89,Mario,Rua do Fontei,1856.93,235.6
',93,Paco,Calle Corredera,1556.23,456.14
!,6,Rosa,Calle clavel,897.56,234.5
  
```

Podemos ver el contenido del fichero que antes de introducir el último empleado ya contenía 5 empleados como nos pedía la tarea

3. La tercera genera el fichero empleados.xml leyendo el con el contenido del fichero EMPLEADOS.DAT



Podemos ver a la izquierda que se ha generado el fichero mostrando por pantalla un mensaje de confirmación.

```

empleados.xml x
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <empleados>
3    <empleado código="22">
4      <nombre>Daniel</nombre>
5      <direccion>Calle Santa Tecla</direccion>
6      <salario>1256.89</salario>
7      <comision>2.4</comision>
8    </empleado>
9    <empleado código="11">
10     <nombre>Jose</nombre>
11     <direccion>Avenida Castilla</direccion>
12     <salario>1078.45</salario>
13     <comision>4.6</comision>
14   </empleado>
15   <empleado código="13">
16     <nombre>Luis</nombre>
17     <direccion>Calle Kamela</direccion>
18     <salario>2134.78</salario>
19     <comision>8.1</comision>
20   </empleado>
21   <empleado código="89">
22     <nombre>Mario</nombre>
23     <direccion>Rua do Fontei</direccion>
24     <salario>1856.93</salario>
25     <comision>235.6</comision>
26   </empleado>
27   <empleado código="93">
28     <nombre>Paco</nombre>
29     <direccion>Calle Corredera</direccion>
30     <salario>1556.23</salario>
31     <comision>456.14</comision>
32   </empleado>
33   <empleado código="6">
34     <nombre>Rosa</nombre>
35     <direccion>Calle clavel</direccion>
36     <salario>897.56</salario>
37     <comision>234.5</comision>
38   </empleado>
39 </empleados>
40

```

Este sería el fichero xml que nos genera la aplicación haciendo uso de DOM como nos pide la tarea

4. La opción Salir cierra la aplicación.

APARTADO 2

Aquí nos piden que mostremos por pantalla el fichero “libros.xml” que nos facilita la propia tarea utilizando DOM y SAX de la API de java.

Para ello creamos una clase libros que contendrá los datos del mismo, un manejador de archivos necesario para parsear con SAX y una clase principal o main donde mostramos al usuario un menú con las 2 opciones de visualización para que sea el quien seleccione la opción:

```
Output - Tarea2AD (run) ...X
run:
MENÚ DE SELECCIÓN PARA MOSTRAR LAS ETIQUETAS DEL FICHERO 'LIBROS.XML'
-----
1.-Visualizar el fichero con la API SAX
2.-Visualizar el fichero con la API DOM
3.-Salir
Elija opción: |
```

1. Visualizar con SAX el fichero

```
Output - Tarea2AD (run) ...X
2.-Visualizar el fichero con la API DOM
3.-Salir
Elija opción: 1

Publicado en: 1994
El título es: TCP/IP Illustrated

El apellido del autor es: Stevens
El nombre del autor es: W.

La editorial del libro es: Addison-Wesley
El precio del libro es: 65.95
-----

Publicado en: 1992
El título es: Advan Programming for Unix environment

El apellido del autor es: Stevens
El nombre del autor es: W.

La editorial del libro es: Addison-Wesley
El precio del libro es: 65.95
-----

Publicado en: 2000
El título es: Data on the Web

El apellido del autor es: Abiteboul
El nombre del autor es: Serge

El apellido del autor es: Buneman
El nombre del autor es: Peter

El apellido del autor es: Suciu
El nombre del autor es: Dan

La editorial del libro es: Morgan Kaufmann editorials
El precio del libro es: 39.95
-----
```

Esta sería el resultado de la visualización con SAX, normalizado y tabulado

2. Visualizar con DOM el fichero

```
MENÚ DE SELECCIÓN PARA MOSTRAR LAS ETIQUETAS DEL FICHERO 'LIBROS.XML'
-----
1.-Visualizar el fichero con la API SAX
2.-Visualizar el fichero con la API DOM
3.-Salir
Elija opción: 2

Titulo del libro: TCP/IP Illustrated
  Año de publicación: 1994
  Autor o autores:
    apellidos: Stevens
    nombre: W.

  Editorial: Addison-Wesley
  Precio: 65.95
-----

Titulo del libro: Advan Programming for Unix environment
  Año de publicación: 1992
  Autor o autores:
    apellidos: Stevens
    nombre: W.

  Editorial: Addison-Wesley
  Precio: 65.95
-----

Titulo del libro: Data on the Web
  Año de publicación: 2000
  Autor o autores:
    apellidos: Abiteboul
    nombre: Serge

    apellidos: Buneman
    nombre: Peter

    apellidos: Suciu
    nombre: Dan

  Editorial: Morgan Kaufmann editorials
  Precio: 39.95
-----
```

Esta sería el resultado de la visualización con DOM, normalizado y tabulado

En conclusión, como podemos ver el resultado de la visualización es prácticamente el mismo, la diferencia principal entre los 2 está en la forma en que leen el fichero, SAX lo hace secuencialmente y DOM va leyendo por nodos que son los elementos y atributos del xml. De cara al usuario el resultado es el mismo porque la diferencia está por detrás, con clases y métodos distintos.