



Acceso a Datos

UT01 - Ficheros

Act.03 - Gestión de Datos de Empleados

UT01. Act03

Gestión de Datos de Empleados

Objetivos

En esta actividad, pondrás en práctica habilidades de manejo de archivos y procesamiento de datos en Kotlin. Trabajarás con archivos de texto plano para leer información de empleados, crearás un archivo XML a partir de esos datos utilizando DOM, realizarás modificaciones en el XML y luego leerás el archivo XML para mostrar los datos en la consola. Esta práctica integra conocimientos sobre flujos de caracteres, manipulación de XML y técnicas de procesamiento de archivos.

Realice las siguientes tareas:

1. Lectura de empleados desde archivo de texto:

- Debes leer un archivo de texto con los datos de los empleados utilizando flujos de caracteres.
- El archivo debe contener una lista de empleados, donde cada línea representa un empleado y los campos están separados por comas.
- El fichero a leer lo puedes encontrar en el siguiente [enlace](#)
- Los campos que contiene cada línea son:
 - ID: Identificador único del empleado.
 - Apellido: Apellido del empleado.
 - Departamento: Departamento al que pertenece el empleado.
 - Salario: Salario actual del empleado.

2. Generación de un archivo XML:

- A partir de los datos leídos del archivo de texto, deberás generar un archivo XML que contenga la información de los empleados. Utiliza el modelo DOM para crear este archivo XML.
- El archivo XML debe seguir la estructura siguiente:

```
<empleados>
  <empleado id="101">
    <apellido>García</apellido>
    <departamento>Recursos Humanos</departamento>
    <salario>3000</salario>
  </empleado>
  <empleado id="102">
    <apellido>López</apellido>
    <departamento>Desarrollo</departamento>
    <salario>4000</salario>
  </empleado>
  <!-- Otros empleados -->
</empleados>
```

3. Modificación de un nodo en el archivo XML:

- Implementa una función que permita modificar el salario de un empleado en base a su ID. Por ejemplo, si se le pasa la ID 102 y el nuevo salario de 4500, la función debe modificar el nodo correspondiente en el archivo XML:

```
<empleado id="102">
  <apellido>López</apellido>
  <departamento>Desarrollo</departamento>
  <salario>4500</salario>
</empleado>
```

4. Lectura del archivo XML modificado y salida en consola:

- Finalmente, debes leer el archivo XML resultante y mostrar la información de todos los empleados en la consola de la siguiente forma:

```
ID: 101, Apellido: García, Departamento: Recursos Humanos, Salario: 3000
ID: 102, Apellido: López, Departamento: Desarrollo, Salario: 4500
...
```

Sugerencias:

- Utiliza las clases *Path*, *Files* y *BufferedReader* para leer el archivo de texto.
- Usa las clases de la biblioteca *javax.xml.parsers* y *org.w3c.dom* para manipular el XML.
- Considera el manejo adecuado de excepciones para los casos en los que el archivo no exista o los datos no tengan el formato adecuado. (investigar sobre Kotlin el tema de las excepciones)
- Asegúrate de cerrar los flujos de datos al terminar de leer o escribir los archivos. (recuerda, *.use{}*)

RAs y CEs evaluados

RA 1. Desarrolla aplicaciones que gestionan información almacenada en ficheros identificando el campo de aplicación de los mismos y utilizando clases específicas.

- c) Se han utilizado clases para recuperar información almacenada en un fichero XML.
- d) Se han utilizado clases para almacenar información en un fichero XML.
- e) Se han utilizado clases para convertir a otro formato información contenida en un fichero XML.
- f) Se han previsto y gestionado las excepciones.

Indicadores	Niveles de logro		
	Bien	Regular	Insuficiente
Recupera la información del XML	Imprime correctamente la información almacenada en el fichero XML 100%	Imprime la información del fichero XML pero de una manera defectuosa 50%	Recupera de manera errónea la información del XML. No imprime correctamente los campos 0%
Escribe la información en el XML	Escribe y modifica de manera correcta la información en el XML	Escribe de manera correcta la información en el XML	Presenta fallos tanto al escribir como al modificar la información en el XML
Lee correctamente la información del CSV	Vuelca correctamente la información en el XML	Presenta ciertos fallos a la hora de volcar la información al XML	No traslada correctamente la información del CSV al XML

Excepciones y limpieza del código	La aplicación no termina de forma inesperada en ninguno de los casos Tanto la estructuración como los comentarios del código aportan valor.	La aplicación termina de forma inesperada en algunos casos El código es limpio y estructurado	La aplicación termina de forma abrupta en casos muy simples Realiza todo en el main sin usar funciones ni clases
-----------------------------------	--	--	---