

# PRESA. UNIDAD 1. ACCESO A LOS ARCHIVOS PARTE 2. EJERCICIOS NO EVALUABLES

## PRESA. Acceso a Datos (ADA) (a distancia en inglés)

### Unidad 1. ACCESO A LOS EXPEDIENTES

#### Parte 2. Archivos. XML y XSL. Ejercicios no evaluables

Abelardo Martinez

Basado y modificado de Sergio Badal ([www.sergiobadal.com](http://www.sergiobadal.com))

Año 2023-2024

# Aspectos a tener en cuenta

## Importante

Si buscas las soluciones navegando por Internet o preguntando al oráculo de ChatGPT te estarás engañando. Tenga en cuenta que ChatGPT no es infalible ni todopoderoso.

Es una gran herramienta para acelerar tu trabajo una vez que dominas una materia, pero utilizarla como atajo a la hora de adquirir habilidades y conocimientos básicos socava gravemente tu aprendizaje. Si lo utiliza para obtener soluciones o consejos por su cuenta, consulte también detenidamente las soluciones propuestas. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación ampliada que encontrarás en el "Aula Virtual".

# Consejos para programar

Recomendamos seguir los siguientes estándares de codificación:

- Una instrucción por línea.
  - Agregue comentarios para que su código sea más claro y legible.
  - Utilizar la notación húngara para reconocer el tipo de variables a primera vista.
  - Recuerda que hay varias formas de implementar una solución, así que elige la que más te guste.
- Recomendamos encarecidamente utilizar soluciones basadas en buffer.

# 1. Modo consola. Lectura de archivos XML con SAX

## Actividad (no evaluable)

Crear un programa en Java para gestionar PIZZAS en un restaurante italiano imprimiendo y utilizando un menú específico. Después de cada opción, el usuario debería ver el mismo menú hasta que se presione la opción cero. No dudes en compartir tus dudas en el foro UNIT.

ATENCIÓN: Utilice las excepciones adecuadas al acceder a archivos.

Opciones de menú:

- Presione 1 para "Pedir pizzas hasta que el usuario ingrese cero como ISBN" ◦ Para cada pizza necesitamos el ID (Entero), el Nombre (Cadena con espacios) y el Precio (Doble), agregados al ArrayList de pizzas.
  - Una vez que se ingresa cero como ID, todas las pizzas se guardarán en un archivo XML llamado "pizzas.xml" (usando DOM), con un formato adecuado, sobrescribiendo todo el archivo si existe.
  - ATENCIÓN: ¡El ID debe ser un número entero! Para cada pizza, debes pedir un número entero (en bucle) hasta que el usuario ingrese un número entero o cero para imprimir nuevamente el menú.
  - ATENCIÓN: ¡El precio debe ser el doble! Por cada pizza, deberás pedir una doble (en bucle) hasta que el usuario ingresa un doble (número de coma flotante).
- Presione 2 para "Listar todas las pizzas almacenadas (usando SAX)"
  - Simplemente lea el archivo XML usando SAX e imprima toda la información de la pizza.
- Presione 3 para "Eliminar todas las pizzas"
  - Simplemente elimine el archivo XML.
- Presione 0 para "Salir"

Ejemplo de menú:

\*\*\*\*\*

MENU

\*\*\*\*\*

=====

- 0. Exit
- 1. Add pizzas
- 2. List all pizzas
- 3. Remove all pizzas
- 4. Generate HTML report using XSL

=====

Select an option:

## 2. Modo consola. Administrar archivos XML con DOM

### Actividad (no evaluable)

Crear un programa en Java para gestionar ANIMALES en una clínica veterinaria imprimiendo y utilizando un menú específico. Después de cada opción, el usuario debería ver el mismo menú hasta que se presione la opción cero. No dudes en compartir tus dudas en el foro UNIT.

ATENCIÓN: Utilice las excepciones adecuadas al acceder a archivos.

Opciones de menú:

- **Presione 1 para "Preguntar por animales (pacientes) hasta que el usuario ingrese cero como ID"**
  - Para cada animal necesitamos el ID (Cadena), Nombre (Cadena con espacios), Especie (Cadena), Raza (Cadena con espacios) y Edad actual (Entero) y Esterilizado (Cadena), guardados en un ArrayList de animales. Si un animal está esterilizado, escribimos "sí" y "no" en caso contrario.
  - Una vez que se ingresa cero como ID, todos los animales se guardarán en un archivo XML con una identificación adecuada formato.
  - ATENCIÓN: ¡La edad debe ser entera!
- **Presione 2 para "Listar todos los animales almacenados (usando DOM)"**
  - Simplemente lea el archivo XML usando DOM e imprima cada animal.
- **Presione 3 para "Esterilizar una raza específica"**
  - Después de preguntar por la Raza (ejemplo: PERRO), recorremos el DOM y cambiamos Esterilizado a "sí" para cada animal cuya raza coincida con la raza introducida anteriormente.
  - Luego (sobre)escribimos el XML nuevamente.
- **Presione 4 para "Esterilizar a personas que no sean adultos"**
  - Recorremos el DOM y cambiamos el campo Esterilizado a "sí" para cada animal. (edad < 2) con Esterilizado "no".
  - Luego (sobre)escribimos el XML nuevamente.
- **Presione 5 para "Eliminar todos los animales"**

- Simplemente elimine el archivo XML.

- Presione 0 para "Salir"

Ejemplo de menú:

\*\*\*\*\*

MENU

\*\*\*\*\*

- ```
=====
0. Exit
1. Add animals
2. List all animals
3. Sterilise a specific breed
4. Sterilise non-adults
5. Remove all animals
6. Generate HTML report using XSL
=====
```

Select an option:

### 3. Modo consola. Convertir XML a HTML

#### Actividad (no evaluable)

Añade una nueva opción a los menús de los ejercicios 2 (pizzas) y 3 (animales) para “Generar informe HTML usando XSL”. No dudes en compartir tus dudas en el foro UNIT.

- Busque una plantilla predefinida en Internet para mostrar una lista de elementos.
- Simplemente cree el XSL necesario para generar este HTML con este CSS.
- Por ejemplo, puedes usar este: ◦ <https://codepen.io/segundofret/pen/nxVKOD>

## HelvetiList

Zurich

Geneva

Winterthur

Lausanne

Lucerne

Aquí tienes más plantillas:

- <https://designshack.net/articles/css/5-simple-and-practical-css-list-styles-you-can-copy-and-pegar/>
- <https://speckyboy.com/html-lists-style/>
- <https://dev.to/hadrysmateusz/custom-list-styles-using-marker-before-4gi4>



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)