

**Universidad de Costa Rica**  
**Facultad de Ingeniería**  
**Escuela de Ciencias de la Computación e Informática**

CI-1310 Sistemas Operativos  
Grupo 01  
II Semestre

**Tarea 0: Manejo de etiquetas en archivos XML**

**Profesor:**  
Francisco Arroyo

**Estudiantes:**  
Valeria Zamora Zamora | B37743

**7 de setiembre del 2018**

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>3</b>
<b>3. Descripción</b>	<b>4</b>
<b>4. Diseño</b>	<b>5</b>
<b>5. Desarrollo</b>	<b>6</b>
<b>6. Manual de usuario</b>	<b>7</b>
Requerimientos de Software . . . . .	7
Compilación . . . . .	7
Especificación de las funciones del programa . . . . .	7
<b>7. Casos de Prueba</b>	<b>8</b>

## 1. Introducción

En esta tarea se pretende crear una primera versión de una herramienta capaz de extraer las etiquetas XML de archivos con este formato. Además, se obtiene un archivo de salida con los datos del archivo sin las etiquetas. Por otro lado, se busca familiarizarse con el ambiente de programación en Linux, específicamente utilizando Fedora y C++.

## 2. Objetivos

- Familiarizar al estudiante con los ambientes y herramientas de programación disponibles en el sistema operativo UNIX, al menos con algunos de sus sabores: Linux, BSD.

### 3. Descripción

El programa debe desarrollarse en lenguaje de programación C o C++ en cualquier ambiente (UNIX, Linux, DOS, WINXX, etc.) pero debe correr adecuadamente en los ambientes UNIX de los laboratorio de la ECCI Hacer un programa que en C o C++ que tome como entrada un archivo XML, contabilize las marcas, que son tiras de caracteres encerradas por los delimitadores «z »z separadas por espacios. Haga una lista de marcas no repetidas ordenada alfabéticamente. Cuente además la cantidad de veces que aparece cada marca dentro del archivo original.

Por ejemplo, en «hola mundo» la etiqueta es "hola", en «hola» también; en </hola> la etiqueta es /holaz es distinta a "hola". Además, de manera opcional, utilizando un parámetro t, especificado desde la línea de comandos (tarea -t hola.xml), genere otro archivo en el que se haya eliminado todas las marcas. El resultado del programa deberá sacar un listado con todas las marcas diferentes en ese archivo como por ejemplo, (...; <PLAN>, 1; ... </PLAN>, 1 ...) donde uno representa las veces que apareció esta palabra en el archivo. Note que las etiquetas "PLANz /PLAN" son diferentes.

## 4. Diseño

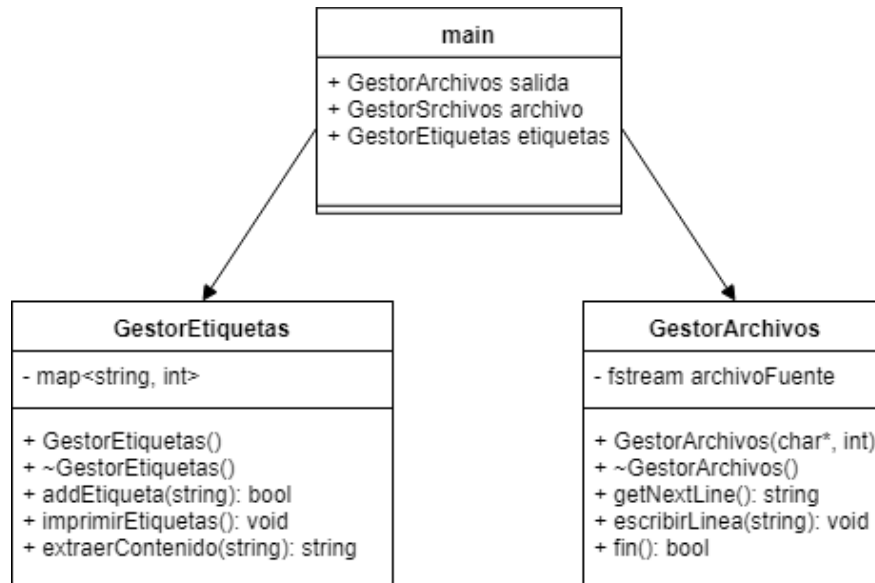


Figura 1: Diagrama de clases

## 5. Desarrollo

Para resolver el problema planteado se creó el diseño presente en la sección anterior. Como se puede ver en la figura 1, se crearon dos clases, una encargada de la gestión de archivos de entrada y salida y otra encargada de las etiquetas. Las etiquetas se almacenan en la estructura de datos map de C++, donde se mantienen en orden alfabético para si posterior despliegue. Los archivos de entrada se analizan línea por línea con el fin de no tener que cargar todas las líneas a memoria directamente. Cada línea es analizada por el gestor de etiquetas, el cual identifica las mismas y las almacena en la estructura de datos mencionada. Además, extrae el texto que no forma parte de la etiqueta para poder desplegarlo en otro archivo posteriormente.

## 6. Manual de usuario

Para correr el programa, una vez compilado, se requiere de un archivo etiquetas de la forma de XML. Este archivo debe ser indicado como parámetro en la línea de comandos a la hora de correr el programa. Si se indica el parámetro `-t` a la hora de correr el programa se va a generar un archivo `.txt` que contiene el texto del archivo de entrada sin las etiquetas. Para correrlo desde la consola de Linux se puede utilizar alguno de los siguientes dos comandos, dependiendo de si se desea el archivo de salida o no.

```
./Tarea0 prueba.xml
```

```
./Tarea0 -t prueba.xml
```

### Requerimientos de Software

- **Sistema Operativo:** Linux (Fedora).
- **Arquitectura:** 64 bits.
- **Ambiente:** no hay ningún ambiente requerido.

### Compilación

Para compilar el programa, se utiliza `gcc` por medio de alguna de las siguientes sentencias:

```
g++ -o tarea0 main.cpp GestorArchivos.cpp GestorEtiquetas.cpp
```

### Especificación de las funciones del programa

El programa requiere de un archivo `.xml` o similar para extraer las etiquetas, este debe ser pasado como parámetro en el momento de ejecución indicando el nombre completo del mismo, incluyendo su extensión.

## 7. Casos de Prueba

Se utilizaron como casos de prueba los archivos suministrados por el profesor. A continuación, se muestran los resultados obtenidos.

### Prueba 1: plan de estudios

En esta prueba se utilizó el archivo .xml que contiene el plan de estudios de la carrera Ciencias de la Computación e Informática. El resultado obtenido se muestra en la figura 2. Como se puede ver en la figura 3, si se utiliza el parámetro '-t' se indica al programa que se desea un archivo de salida. Parte del archivo sin etiquetas se muestra en la figura 4.

```
[valeriazamora@localhost Tarea_XML]$ g++ -o Tarea0 *.cpp
[valeriazamora@localhost Tarea_XML]$ ./Tarea0 prueba.xml
Etiquetas:
/CREDITOS: 41
/CURSO: 41
/NIVEL: 8
/NOMBRE: 41
/PLAN: 1
/REQUISITOS: 1
/SIGLA: 41
/TITLE: 1
CREDITOS: 41
CURSO: 41
NIVEL: 8
NOMBRE: 41
PLAN: 1
REQUISITOS: 32
REQUISITOS/: 9
SIGLA: 41
TITLE: 1
```

Figura 2: Salida del programa

```
^[[A[valeriazamora@localhost Tarea_XML]$ ./a.out -t prueba.xml
Etiquetas:
/CREDITOS: 41
/CURSO: 41
/NIVEL: 8
/NOMBRE: 41
/PLAN: 1
/REQUISITOS: 1
/SIGLA: 41
/TITLE: 1
CREDITOS: 41
CURSO: 41
NIVEL: 8
NOMBRE: 41
PLAN: 1
REQUISITOS: 32
REQUISITOS/: 9
SIGLA: 41
TITLE: 1
```

Figura 3: Salida del programa



```

Ciencias de la Computacion e Informatica
id="1">

CI1010
Introducción a Computación e Informatica
4

LM1030
Ingles Intensivo I
4

EG0124
Humanidades I
6 |

MA0129
Matematica para Computacion I
4

EF
Actividad Deportiva
0

id="2">

CI1101
Programación I
4
r1="CI1010">

CI1204
Estructuras Discretas Aplicadas II
4
r1="MA0129"/>

```

Figura 4: Archivo sin etiquetas

## Prueba 2: orden de libros

En esta prueba se utilizó el archivo .xml que contiene la información de libros. El resultado obtenido se muestra en la figura 5. Para correr el programa indicándole que se desea un archivo de salida se utiliza el '-t' como se puede ver en la figura 6, parte del archivo sin etiquetas se muestra en la figura 7.

```
[valeriazamora@localhost Tarea_XML]$ ./Tarea0 prueba_libros.xml
Etiquetas:
/Cardnum: 1
/Custom: 1
/ID: 5
/Item: 5
/Manifest: 1
/Name: 1
/Order: 1
/Quantity: 5
/Receipt: 1
/Subtotal: 1
/Tax: 1
/Title: 5
/Total: 1
/UnitPrice: 5
Cardnum: 1
Customer: 1
ID: 5
Item: 5
Manifest: 1
Name: 1
Order: 1
Quantity: 5
Receipt: 1
Subtotal: 1
Tax: 1
Title: 5
Total: 1
UnitPrice: 5
```

Figura 5: Salida del programa con el archivo de libros

```
[valeriazamora@localhost Tarea_XML]$ ./a.out -t prueba_libros.xml
Etiquetas:
/Cardnum: 1
/Custom: 1
/ID: 5
/Item: 5
/Manifest: 1
/Name: 1
/Order: 1
/Quantity: 5
/Receipt: 1
/Subtotal: 1
/Tax: 1
/Title: 5
/Total: 1
/UnitPrice: 5
Cardnum: 1
Custom: 1
ID: 5
Item: 5
Manifest: 1
Name: 1
Order: 1
Quantity: 5
Receipt: 1
Subtotal: 1
Tax: 1
Title: 5
Total: 1
UnitPrice: 5
```

Figura 6: Salida del programa

---

xmlns="http://www.example.com/myschema.xml">

Bill Buckram  
234 234 234 234

209  
Duke: A Biography of the Java Evangelist  
1  
\$10.75

208  
100% Pure: Making Cross Platform Deployment a Reality  
1  
\$10.75

204  
Making the Transition from C++ to the Java(tm) Language  
1  
\$10.75

202  
Web Servers for Fun and Profit  
1  
\$10.75

210  
I Think Not: Dukes Likeness to the Federation Insignia  
1  
\$10.75

\$53.75  
\$4.43  
\$58.18

Figura 7: Archivo de libros sin etiquetas