

Instituto Tecnológico de Costa Rica

Escuela de Computación

I Proyecto Programado

Lenguajes de Programación

Eliomar Rodríguez Arguedas

Sede San Carlos

Setiembre 2017

## Tabla de contenidos

1. Introducción .....	3
2. Análisis del problema .....	4
3. Solución del problema .....	5
4. Análisis de resultados .....	7
5. Recomendaciones .....	9
6. Conclusiones .....	10
7. Bibliografía .....	11

## **1. Introducción**

En la actualidad, los sitios web utilizan HTTP (Protocolo de Transferencia de Hipervínculos) para la obtención de datos en todos los formatos posibles como JSON y XML y así manejar estos datos como se desea, lo que viene siendo un API REST en sí. Twitter, YouTube, los sistemas de identificación de Facebook y cientos de empresas utilizan este nuevo enfoque para desarrollo de proyectos y servicios web porque es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet. (BBVAOPEN4U, 2016)

Para dar solución a la propuesta del proyecto, se debe acceder al servidor de IDE-HN (Infraestructura de Datos Espaciales de la región Huetar Norte) donde se encuentra la información, a través de HTTP mediante un protocolo REST API que provee la plataforma de donde se extraerán los datos a manejar para luego mediante dos opciones, ejecutar el programa y de esta manera poder desplegar al usuario la información extraída del servidor para que tenga una opción donde puede elegir cuales de la capas recuperadas del servidor desea guardar, y en caso que desee guardar todas las capas también habrá una opción, además luego tendrá la posibilidad de ver que datos almacenó en el archivo.

## 2. Análisis del problema

Se requiere la creación de un pequeño programa que permita la extracción de datos de un servidor de IDE-HN la cual es una plataforma que ofrece servicios de gestión de datos geoespaciales de la Región Huetar Norte, con el fin de contribuir al mejoramiento de las capacidades institucionales para la gestión del territorio. (IDE, s.f.)

El programa debe permitir acceder a dicha información y poder extraerla con la finalidad de que el usuario la pueda ver y en dado caso, elegir con que información quedarse para luego proceder a mostrarla si así lo desea. Para acceder a dicha información para poder manipularla a gusto primero se debe obtener la información de la página, hay dos formas de hacer esto. La primera es utilizar una librería que convierta de formato XML a formato JSON para así poder acceder a la información de una manera mucho más rápida y eficiente, otra opción es acceder al XML del servidor y mediante una librería extraer la información sin la necesidad de tener que almacenar el XML en sí en un archivo para luego analizar y extraer lo necesario para avanzar hasta encontrar la información deseada.

En caso de extraer el XML en un archivo y luego analizar el archivo implica tener que manipular cadenas de texto para realizar operaciones como concatenación, identificación de subcadenas, comparación de cadenas, lectura de datos del archivo, entre otras cosas.

La manera en la que se debe llamar el programa son 4 las cuales según el enunciado son:

1. **./nombre\_proyecto**
2. **./nombre\_proyecto -p**
3. **./nombre\_proyecto -f**
4. **./nombre\_proyecto -f -p**

Más adelante se explica cada una de ellas y como se desarrollaron en el proyecto.

### 3. Solución del problema

Primero se debió instalar la librería CURL para poder utilizarla en C y luego se debió incluir en el IDE a utilizar (Code::Blocks) la librería como tal mediante la instrucción “**-lcurl**” en la parte de “**Other linker options:**” que se encuentra en “**barra de opciones → Settings → Compiler... → Linker setting**”.

La conexión al servidor se realizará mediante el uso de comandos propios de la librería CURL la cual permite la transferencia de datos mediante URLs para así lograr la extracción de la información con el fin de procesarla de la manera deseada.

Para extraer la información en sí se realiza mediante el uso del código dado por el profesor que consiste en la extracción del cuerpo completo del XML de la página y su almacenamiento en un archivo de extensión txt mediante la URL de la página que se desea guardar.

Al tener el XML de la página principal (nivel 1) en un archivo llamado **etapa1.txt** se procede a recorrer dicho archivo en busca de la etiqueta **atom** la cual contiene un **href** con la dirección de otra página (nivel 2), al tener el link de la segunda etapa se procede a enviarlo a la función encargada de extraer todo el XML de la página para proceder a recorrer el archivo donde se almacenó el segundo nivel (**etapa2.txt**), al recorrer el archivo se puede ver que solamente contiene dos etiquetas **atom** (lo que se busca para poder llegar al nivel 3 y extraer la información), la etiqueta **atom** que interesa es la segunda ya que el **href** de la primera redirecciona a un sitio que no interesa, al tener el link del ultimo nivel se envía a la función encargada de extraer todo el XML de la página y almacenarla en un archivo llamado **etapa3.txt**. Al tener el archivo de la última página (nivel 3), se recorre y se procede a la extracción del **title** y la primer línea del **abstract** que es la que interesa, esta información es guardada en una estructura (lista simple enlazada) que tiene un espacio destinado para el title y otro para el abstract, con eso ya se habrá extraído la información de la primera capa y se aplica el mismo procedimiento para el resto de capas de la página principal.

Para poder extraer información de cada etiqueta XML del archivo, es necesario el manejo de char para poder extraer una subcadena la cual tiene la dirección URL de la siguiente etapa como se mencionó anteriormente, para poder detectar que URL obtener, se busca la etiqueta <atom ya que en las primeras dos etapas ahí es donde se encuentra la URL de la siguiente etapa, al encontrarnos en la segunda etapa se usa un contador buscando etiquetas <atom y cuando el contador esté en 2 quiere decir que es el atom que se necesita para poder acceder a la etapa 3, cuando ya se encuentre en la etapa 3, hay que buscar las etiquetas <title> y <abstract> para proceder a extraer la primer línea de ambas etiquetas y almacenarlas en una lista enlazada simple (struct).

Cuando se ha extraído toda la información y almacenados en la lista enlazada simple, se procede a la parte en la que el usuario puede elegir que desea guardar o guardar todo. En el main es necesario el uso de ciertos parámetros los cuales son útiles para pasar parámetros si se desea desde consola, los parámetros que se deben poner en el código del main son: **argc** el cual contiene la cantidad de parámetros incluidos el nombre, **argv []** es un vector que contiene todos los parámetros escritos desde consola incluidos el nombre.

Según el enunciado, se tendrán dos maneras para la ejecución del programan. La primera opción trata de ejecutar el programa desde la terminal de Linux sin interfaz y la otra es desde la ejecución de la aplicación manejando la información extraída del servidor desde la interfaz del programa, ambos modos de ejecución deben acceder al servidor, extraer la información y almacenarla para poder procesarla cuando se esté ejecutando el programa.

Las cuatro formas de ejecución desde la terminal de Linux (luego de acceder a “**Programa/bin/Debug/**”) son:

- Para la primera opción se deberá ejecutar ingresando “**./nombre\_proyecto -f**” para que se guarde toda la información en un archivo llamado “**data.txt**”.
- La segunda opción es “**./nombre\_proyecto -f -p**” y lo que hace es guardar todas las capas de información y luego desplegarle la información de las capas guardadas.
- La tercer opción es “**./nombre\_proyecto**” para que se ejecute el programa y una vez en modo de ejecución preguntará que opciones NO desea guardar de la lista que se muestra en pantalla, se ingresa el número de la capa a descartar y se presiona enter, seguidamente preguntará si se desea continuar descartando capas, si se ingresa la opción de seguir descartando opciones pues continuará, caso contrario se terminará la ejecución del programa
- La cuarta opción es “**./nombre\_proyecto -p**” para que se ejecute el programa, seleccione las capas que desea guardar como la tercer opción pero además de todo lo que hace la tercer opción, muestra las capas almacenadas en el archivo.

Si se desea ver el contenido del archivo data.txt después de haber ejecutado el proyecto solamente se debe acceder a “**Programa/bin/Debug/**” donde programa es la carpeta donde se encuentra el proyecto en sí (proyecto1.cbp) y el archivo que tiene el código del proyecto (main.c).

## 4. Análisis de resultados

A continuación, se presenta una tabla con las actividades a realizar para la elaboración del proyecto en si, algunas tardaron más tiempo que otras en implementarse debido a algunos errores que siempre surgen o salen a la luz a último momento.

Actividad	Descripción	% de avance
Utilización de la librería CURL en el lenguaje C		100
Acceder al servidor	Mediante un comando de la librería CURL que permite el acceso	100
Extrae el XML de las páginas y lo guarda en archivos	Mediante un comando de la librería CURL extrae el código XML de la página y lo almacena en un archivo txt para extraer la información luego	100
Almacena la información extraída en memoria	La información se almacena en <b>structs</b> , luego de haber recorrido el archivo	100
Tener 4 modos de ejecución desde la terminal	. /nombre_proyecto pregunta que capas desea guardar ./nombre_proyecto -p pregunta que capas desea guardar y al final muestra que capas guardó ./nombre_proyecto -f guarda todas las capas . /nombre_proyecto -f -p guarda todas las capas y al finalizar muestra las capas que guardó	100
Dentro de las dos maneras de ejecución permite ingresar la opción -p	Esta opción despliega las capas que el usuario guardó en el archivo	100
Almacenar en el archivo las capas seleccionadas por el usuario, ya sea unas cuantas o todas		100
Validar que el paso de parámetros	No debe poder pasar más de dos parámetros cuales deben ser el nombre del proyecto, y el segundo y único debe ser -f para que guarde todas las capas de información, caso omiso se le notifica el error	100

Validar que las opciones ingresadas dentro de la ejecución del proyecto	No se debe permitir que ingrese letras o valores que no son los que se le solicitan, caso omiso se le notifica el error	100
El uso de la aplicación debe ser intuitivo		100
Muestra la información de una manera ordenada y enumerada		100
El formato de guardado en el archivo debe ser bueno	El resultado final del archivo debe quedar ordenado de la misma manera que se mostró al usuario	100
Carpeta donde se encuentra video que prueba la correcta ejecución del proyecto <a href="https://drive.google.com/drive/folders/0Bxde4gIUnnxlOEZpa05ZTDZ6cEk?usp=sharing">https://drive.google.com/drive/folders/0Bxde4gIUnnxlOEZpa05ZTDZ6cEk?usp=sharing</a>		100



## 5. Recomendaciones

Durante el desarrollo del proyecto, la extracción de los datos que se encuentran dentro de las etiquetas XML se tornó algo difícil por la manera en la que se realizó este proceso. Se recomienda buscar alguna librería que convierta de formato XML a formato JSON ya que sería mucho más sencillo extraer la información utilizando este formato. En el caso de este proyecto, se manejó la extracción de datos a puro manejo de char, cortando las cadenas mediante la instrucción **strtok** el cual recibe dos parámetros, el primero de ellos es la cadena de char que se desea cortar, el segundo parámetro es el delimitador (hasta donde se desea cortar la cadena). Así es como se manejó la extracción de datos, quizá con la ayuda de una librería este proceso pudo haber sido realizado de una manera mucho más sencilla y eficiente, solo habría que investigar más para encontrar la librería ideal.

## **6. Conclusiones**

Al concluir este proyecto, se logró entender cómo acceder a un servidor mediante HTTP utilizando un protocolo REST API. El método utilizado para la obtención de los datos es un poco lento y tedioso, si se fuera hecho uso de alguna librería utilizable en C la cual se encargue de convertir de XML a JSON para que la tarea de extraer la información fuese mucho más eficiente y menos difícil a diferencia de cómo se manejó (manejo de cadenas, concatenación, Split de cadena mediante un separador, etc.).

El haber utilizado una librería que convierta de formato XML a formato JSON implica que el estudiante debía capacitarse en el uso de la misma, pero por el corto lapso de tiempo que se tenía para la elaboración del programa, se decidió utilizar los conocimientos que se tienen hasta el momento para poder avanzar rápido y de esta manera aprovechar al máximo el tiempo disponible.

## 7. Bibliografía

BBVAOPEN4U. (23 de Marzo de 2016). *BBVA Open4U*. Obtenido de <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>

IDE. (s.f.). *IDE Región Huetar Norte*. Obtenido de <http://www.idehn.tec.ac.cr/>