



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Tarea 2

Modelamiento de Procesos e Información

Sección: 1

Integrantes:

Christopher Cromer

Mateo 16 - 20

Profesores:

Christian Vidal

Jorge Elgueta

Índice

Índice	1
Estrategia	2
libxml2 funciones	2
Crear el documento de XML	3
Crear un nodo(etiqueta)	3
Agregar nodo al documento de XML	3
Agregar un atributo a un nodo	3
Agregar nodo a un otro nodo	3
Agregar texto a un nodo	3
Generar salida	3

Estrategia

La estrategia que es empleada en el programa es leer un archivo de configuración “.conf” que está en el siguiente formato de XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE config SYSTEM "config.dtd">
<config>
  <output>chris.xml</output>
  <bible>El Santo Evangelio Según</bible>
  <book>SAN MATEO</book>
  <chapter>
    <name>MATEO</name>
    <number>16-20</number>
  </chapter>
</config>
```

Este archivo contiene el nombre de archivo de salida, cuál biblia, libro y capítulos el programa va a buscar. En el caso de los capítulos es un rango separado por un guión. Luego el programa empieza a leer el archivo línea por línea. Cada línea leída es formateada en UTF-8 usando las funciones de *encoding.c* para que los tildes y otros caracteres especiales no tengan problemas en la búsqueda ni en el archivo de salida. Todas las líneas convertidas ahora están guardadas en un array de string. Cuando todo el texto es convertido a UTF-8 el programa empieza a buscar la biblia, libro y capítulos que corresponden al archivo de configuración. Entonces cuando los 3 criterios son encontrados comienza a guardar los datos de la biblia en un struct de la siguiente forma:

```
struct chapter_data {
    int chapter;
    int current;
    int verses;
    char **verse;
} typedef CHAPTER;
struct book_data {
    int current;
    int chapters;
    CHAPTER **chapter;
} typedef BOOK;
```

Luego los datos guardados en el struct se usan para generar el XML correspondiente.

libxml2 funciones

Estas son las funciones que se utilizan para trabajar con XML usando la librería libxml2.

Crear el documento de XML

```
xmlDocPtr doc = xmlNewDoc(BAD_CAST "1.0");
```

Crear un nodo(etiqueta)

```
xmlNodePtr node = xmlNewNode(NULL, BAD_CAST "nombre");
```

Agregar nodo al documento de XML

```
xmlDocSetRootElement(doc, node);
```

Agregar un atributo a un nodo

```
xmlNewProp(node, BAD_CAST "atributo", BAD_CAST "valor");
```

Agregar nodo a un otro nodo

```
xmlAddChild(nodoPadre, nodoHijo);
```

Agregar texto a un nodo

```
xmlNodePtr node = xmlNewText(BAD_CAST "texto");
```

Generar salida

```
xmlSaveFormatFileEnc("-", doc, "UTF-8", 1);
```

El `BAD_CAST` es un macro que se usa para cast un char a `xmlChar`. En la función `xmlSaveFormatFileEnc`, el primer argumento es el nombre del archivo de salida, o un guión si desea que el XML salga en la consola.