RTHC: Libertad para el formato RTF

Esteban Manchado Velázquez, José Miguel Santos Espino *
Facultad de Informática, Universidad de Las Palmas de Gran Canaria
{a2092,jomis}@dis.ulpgc.es

20 de julio de 2000

Esta ponencia forma parte del III Congreso Hispalinux[1] virtual, celebrado en Noviembre de 2000. Puedes participar en él en la sección de eventos de BarraPunto[2]

Resumen

El formato RTF es ampliamente utilizado en los sistemas Windows. Los sistemas abiertos, y en especial GNU/Linux, no pueden dejar de usarlo si pretenden ser sistemas de uso universal. El proyecto RTHC consiste en una biblioteca de clases en C++ que facilitan el desarrollo de programas que manejan RTF, así como un conversor de RTF a HTML basado en esta biblioteca. RTHC queda a disposición de la comunidad de usuarios y programadores a través de la licencia GPL, para permitir que el formato RTF sea fácilmente *domesticado*.

1 Introducción

La comunidad del *software* libre conoce y aprecia desde hace años la importancia que tienen los sistemas y normas abiertos en la producción y difusión de documentos. Los sistemas abiertos nos permiten intercambiar documentos con otras personas sin pérdida de información y a un bajo coste. Los programadores pueden desarrollar con facilidad software que maneje estas clases de documentos, etc.

Sin embargo, los sistemas cerrados siguen siendo los más comunes en la actualidad, y una buena proporción de los documentos escritos hoy día se encuentra almacenada en formatos propietarios. Si el software libre, y en especial GNU/Linux, pretende tener alcance universal, debe proporcionar herramientas para manejar cómodamente los formatos cerrados de documentos.

Un caso paradigmático de formato propietario de amplísimo uso es el «Formato de Texto Enriquecido» (*Rich Text Format*, RTF) de Microsoft [1], utilizado por la *suite* ofimática MS-Office y por múltiples aplicaciones de otras firmas. En GNU/Linux hay diversos programas que trabajan con este formato, bien de forma nativa o mediante conversiones. Como ejemplos de estos, podemos citar:

StarOffice Un paquete ofimático, de prestaciones similares al Office, que trabaja con varios formatos. El procesador de textos entiende el formato RTF y es capaz de escribir documentos en él. A pesar de que era un programa propietario y cerrado, actualmente es libre, aunque su envergadura lo hace poco práctico como ejemplo, y probablemente no está pensado para ser extensible.

rtf2html Un conversor de RTF a HTML, disponible para varias plataformas, distribuido como *shareware*. Su código fuente no es utilizable públicamente.

AbiWord [2] Un pequeño procesador de textos libre, que entiende el formato RTF. Actualmente, el programa en general y el filtro de RTF en particular están en un estado relativamente primitivo.

Ted Un programa parecido al WordPad de Microsoft, que trabaja con el RTF como formato nativo. Sus pretensiones son modestas.

[1] `http://congreso.hispalinux.es`

[2] `http://barrapunto.com/eventos`

Como podemos ver, en GNU/Linux podemos visualizar e incluso escribir documentos RTF, pero a nuestro juicio siguen existiendo carencias: se echa en falta una biblioteca genérica para manejar documentos RTF, que sea utilizable por los programadores para tareas como: escribir aplicaciones sencillas (p.ej. filtros UNIX que extraigan los títulos o los autores de un conjunto de documentos RTF); importar o exportar RTF en procesadores de textos, etc. La reciente apertura del código de StarOffice puede ser prometedora, pero la complejidad de este software disuade su aprovechamiento para utilidades sencillas.

Otra carencia importante es la documentación del software, que tantas veces es descuidada por los desarrolladores y que impide su reutilización. A esto hay que añadir una dificultad inherente al RTF, y es que las especificaciones publicadas por Microsoft no son precisamente un dechado de claridad.

2 El proyecto RTHC

Ante lo aquí expuesto, nos podemos plantear tres vías para culminar la apertura del formato RTF y llevarlo al redil de los entornos abiertos:

- Documentar adecuadamente el formato, «abriéndolo» en la medida de lo posible, para permitir que otras personas diseñen programas que, siguiendo la especificación mejorada, entiendan los documentos en el susodicho formato.
- Construir un programa abierto capaz de visualizar documentos RTF.
- Construir un programa, abierto o no, capaz de convertir documentos RTF a algún formato abierto, como HTML o PostScript.

En sintonía con estos planteamientos, los autores decidimos emprender un trabajo de fin de carrera consistente en estudiar el formato RTF, documentarlo, y construir una biblioteca de clases en C++, llamada *libFreeRTF*, capaz de analizar documentos RTF. Además de lo anterior, se ha implementado un programa conversor de RTF a HTML, llamado RTHC[3]. Este conversor está basado en *libFreeRTF* y sirve de ejemplo de uso de esta biblioteca.

La arquitectura software de RTHC facilita la reusabilidad. Está organizado en componentes bastante desacoplados, con responsabilidades bien delimitadas: análisis del documento RTF, almacenamiento en memoria y recorrido/interpretación. Este enfoque permite modificar el comportamiento de cada uno de componentes sin necesidad de alterar el resto. El diseño de *libFreeRTF* es simple y extensible, lo que permite fácilmente modificarlo, adaptarlo y mejorarlo. Por otro lado, sirve como documentación del RTF y como ejemplo de implementación de un analizador de este formato.

No sólo se pretende documentar el RTF y diseñar una implementación de prueba, sino que se intenta potenciar el desarrollo de *software* libre en la Universidad de Las Palmas de Gran Canaria. Por ello, además de desarrollar el código, se ha empaquetado en los formatos más populares (RPM y Debian) y se ha investigado sobre las maneras más apropiadas y aceptadas por la comunidad, de difusión y publicación del programa. Este esfuerzo ayudará a emprender futuros proyectos de desarrollo de *software* libre.

Seguidamente se hará una introducción al formato RTF, se explicará cómo funciona el conversor de RTF a HTML y se describirá a grandes rasgos la arquitectura de la biblioteca *libFreeRTF*.

3 El formato RTF

RTF es uno de los formatos nativos del Microsoft Word. Teóricamente, todas y cada una de sus opciones son representables en este formato. Teniendo en cuenta que las opciones de Word se han ido añadiendo a lo largo de los años, el RTF es bastante desordenado.

Los elementos básicos que constituyen un documento RTF son:

Palabras de control una serie de caracteres alfabéticos, precedidos de una barra invertida y seguidos, optativamente, por un parámetro numérico (que puede ser negativo).

Símbolos de control una barra invertida seguida de un solo caracter no alfabético.

Caracter especial una barra invertida seguida de un apóstrofo y dos caracteres hexadecimales.

Grupos elementos contenidos entre llaves.

Texto todo lo demás.

[3] «RTF To HTML Converter», o «Rotten Text Happily Converted»

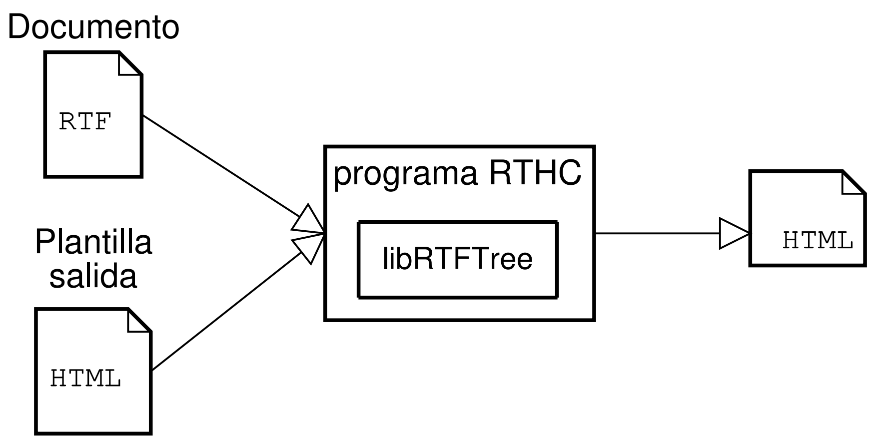


Figura 1: Arquitectura funcional de RTHC

Estos elementos se agrupan en el documento RTF prácticamente sin orden establecido. Uno puede encontrarse cualquier combinación de palabras de control, texto y grupos, aunque ciertas combinaciones tienen significados definidos, como las tablas o las listas. Las especificaciones oficiales tan sólo definen cierta estructura para la cabecera del documento, proponiendo una descripción en seudo-BNF incompleta e inexacta.

La casi inexistente sintaxis de RTF hace que diseñar un analizador sea complicado. Por si fuera poco, el propio Microsoft Word, la implementación que se ha seguido como modelo, incumple bajo ciertas condiciones la especificación oficial.

4 El programa conversor

El programa RTHC se ofrece como una típica aplicación UNIX para línea de órdenes, que recibe como entrada un documento RTF y lo convierte a una salida HTML. RTHC, bajo ciertas condiciones, puede usarse como filtro en líneas de órdenes UNIX.

Para personalizar el aspecto del documento de salida HTML, se ha optado por usar plantillas. Éstas son simples documentos HTML que tienen referencias a ciertas variables que RTHC entiende y sustituye. De esta manera damos flexbilidad total al usuario para definir el aspecto final de la salida. Una alternativa a las plantillas consiste en dar cientos de opciones para personalizar todos los elementos imaginables del aspecto, como las barras de botones, los títulos, los encabezados y pies, etc. Este modelo es el que utiliza el programa *shareware* rtf2html, pero nos parece complicado, poco práctico, y no tan flexible como las plantillas.

Hay aspectos no personalizables mediante plantillas, como el nivel al que se cortan los documentos, si se desea generar un índice o si hay que generar una página con marcos (*frames*). Para estos casos se utilizan opciones de línea de órdenes, según el estilo GNU (p.ej. `--with-frames`).

La correspondencia entre las palabras de control RTF y el contenido HTML se establece con ficheros de configuración. El conversor usa correspondencias por defecto, así que también se evita la necesidad de ficheros de configuración gigantescos (otro inconveniente de rtf2html).

Volviendo a las plantillas, es destacable el hecho de que se utiliza una sintaxis familiar para todos los programadores de UNIX: el signo dólar seguido del nombre de la variable, que es la convención utilizada por Perl y PHP. Hablando de estos dos programas, hemos de destacar que las plantillas pueden usarse para que RTHC genere programas en Perl o PHP, de manera que la salida del conversor ¡se puede ejecutar como un programa! Esto permite usar RTHC para invocar operaciones avanzadas, empotrarlo como CGI en un servidor web, etc.

5 Implementación

Para alcanzar la pretensión inicial de máxima reutilización, las clases que ocultaran la complejidad del RTF tenían que ser fáciles de usar, a la vez que reflejaran fielmente la estructura de los documentos RTF. Y además, pretendíamos que *libFreeRTF* fuera fácil de extender y modificar.

Hemos utilizado patrones de diseño comunes [3], explotado los genéricos estándares de C++ (STL) [4], y hemos empleado la notación UML [5] para representar la arquitectura. Este diseño basado en objetos y

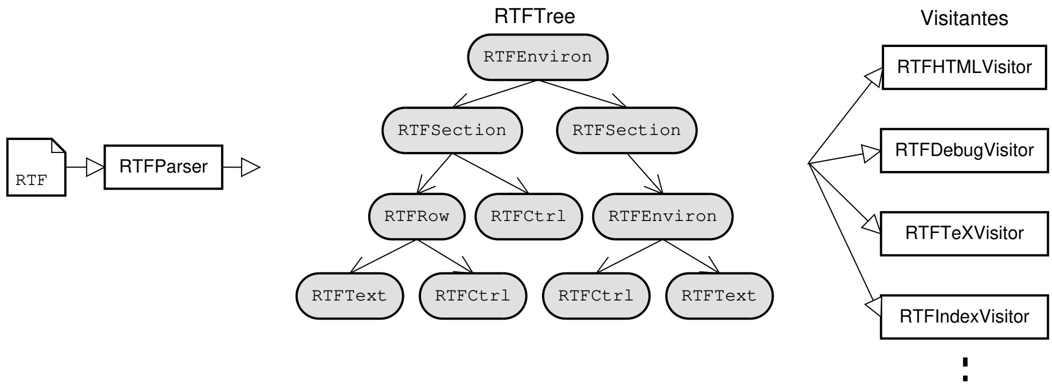


Figura 2: Arquitectura interna de *libFreeRTF*

patrones comunes garantiza legibilidad y un corto tiempo de aprendizaje para un desarrollador que se enfrente por vez primera con RTHC.

5.1 Arquitectura de *libFreeRTF*

libFreeRTF es básicamente un conjunto de clases para almacenar y construir un árbol que representa a un documento RTF. El árbol pretende abstraer la estructura de RTF, convirtiendo cada palabra de control en un nodo. Algunos elementos de RTF se resisten a la abstracción, como ocurre con las referencias cruzadas y las listas.

En los casos en los que la complejidad nos prohibía encapsular ciertas propiedades en nodos específicos, se optó por documentar en la medida de lo posible[4] esos casos, para ayudar al programador.

El árbol RTF tiene que ser sencillo de recorrer, y las operaciones efectuadas en los recorridos han de ser independientes de la estructura interna del árbol, ya que se debe permitir cualquier recorrido imaginable. Esto se soluciona con el patrón de diseño «visitante», que consiste en definir por un lado un conjunto de operaciones y por otro los tipos de nodos que puede tener el árbol. Estos dos conjuntos son en realidad jerarquías de clases, en los que cada una es una operación o un tipo de nodo.

En definitiva, la arquitectura del componente *libFreeRTF* queda como una clase `RTFTree` que almacena en memoria un documento en formato RTF, una clase `RTFParser` que construye árboles `RTFTree` dado un fichero de texto enriquecido, una jerarquía de clases `RTFContents`, compuesta por las clases de nodos que se pueden guardar en un árbol `RTFTree`, y una jerarquía de visitantes `RTFVisitor`, que llevan a cabo cierta operación sobre el árbol, como mostrar su estructura, contar el número de caracteres, traducirlo a otro formato, etc.

5.1.1 Árbol del documento: la clase `RTFTree`

La clase RTFTree es la que guarda todo el contenido de un documento RTF, y da una interfaz limpia tanto para construirlo (cosa que hará `RTFParser`) como para recorrerlo (el trabajo de los descendientes de `RTFVisitor`). Básicamente, se compone de la información encontrada en la cabecera, que afecta a todo el documento, y de un puntero de tipo `RTFContents*`, que apunta a los nodos del árbol.

5.1.2 La jerarquía `RTFContents`

Es la jerarquía de clases de los entes que pueden formar parte de un documento RTF: desde `RTFText` para texto llano, hasta `RTFRow` para guardar filas de una tabla, o `RTFSpecialChar` para los símbolos dados como valores hexadecimales de cierta tabla de caracteres. Los métodos encontrados en esta clase están orientados a la construcción del árbol y a su recorrido.

5.1.3 Analizador sintáctico: la clase `RTFParser`

Esta clase es responsable de recorrer el fichero RTF para codificar en un árbol toda la información. Ofrece un método público para construir árboles `RTFTree` a partir de un flujo de datos del que leer un documento RTF

[4]Con frecuencia, las especificaciones de formato complicadas venían acompañadas de una documentación especialmente pobre o nula, así que la información es pura especulación

(`newTree`). Esta clase se apoya en un analizador léxico, generado con *flex++*, que es quien realmente recibe el flujo de datos y devuelve los diferentes símbolos léxicos.

Fue en la implementación de esta clase donde se encontraron todos los fallos de la documentación, las omisiones, y demás dificultades relacionadas directamente con las especificaciones RTF.

5.1.4 La jerarquía `RTFVisitor`

Es la jerarquía que permite producir resultados una vez que disponemos de un árbol RTF. La clase raíz, `RTFVisitor`, define operaciones virtuales que manipulan (*visitan*) cada uno de los distintos tipos de nodos del árbol, por ejemplo `visitRTFText()`, `visitRTFStar()`, `visitRTFPard()`, etc. El programador puede definir clases derivadas de `RTFVisitor` que implementen comportamientos diferentes para estas operaciones y, en consecuencia, toda la variedad imaginable de aplicaciones sobre un árbol RTF: generar HTML, descubrir errores, extraer estadísticas...

La biblioteca *libFreeRTF* proporciona tres clases visitantes: `RTFDebugVisitor`, que muestra una representación en texto de la estructura del árbol; `RTFPlainTextVisitor`, que traduce a texto sin formato el documento original; `RTFHTMLVisitor`, que traduce a HTML y es el corazón de la aplicación RTHC.

En principio, todas las clases visitantes deberían ser adaptadas si se añade un nuevo tipo de nodo a `RTFTree`, pero tal y como está implementado, sólo hace falta reescribir código en la clase base, proporcionando una nueva operación con un comportamiento por defecto. Con ello se aumenta la reusabilidad de la biblioteca.

5.2 Reusabilidad y extensibilidad

La biblioteca *libFreeRTF* permite escribir con relativa sencillez programas de todo tipo que trabajen con documentos RTF. La separación entre las clases del árbol y los visitantes facilita la construcción de conversores a otros formatos como TEX, PostScript o SGML. Sólo hay que implementar un heredero de la clase visitante `RTFVisitor` sin tener que intervenir en el árbol o el analizador sintáctico. Además, se pueden reutilizar los comportamientos predefinidos de las clases visitantes proporcionadas por la biblioteca.

En esta línea, no debería ser difícil implementar un conversor a la emergente norma XML. Por su similitud con HTML, se puede aprovechar gran parte de la clase `RTFHTMLVisitor`.

Otras aplicaciones que pueden resolverse con esta técnica son: un programa que construya índices o tablas de contenido; utilidades de línea de órdenes que devuelvan estadísticas de los documentos. Incluso es planteable hacer un conversor de RTF a RTF: dado que los RTF generados por el propio Word de Microsoft contienen bastante información redundante, se puede hacer un visitante que «embellezca» el RTF original y lo convierta en algo más manejable.

El formato RTF puede estar sujeto a modificaciones a capricho de la empresa propietaria. El proyecto RTHC trata de facilitar la adaptación a estos cambios por dos vías: primero, gracias a una documentación que aclara los aspectos más oscuros de las especificaciones de Microsoft; segundo, gracias a que *libFreeRTF* es una biblioteca abierta y gratuita, que queda a disposición de la comunidad para que evolucione. La mayoría de los cambios futuros en el formato RTF sólo requerirán intervenir en el árbol y el analizador. Los visitantes existentes pueden ser reutilizados casi sin cambios.

El formato nativo de Word es prácticamente una representación binaria de RTF. Por ello, con leves cambios en el analizador sintáctico, se puede disponer de los servicios de RTHC para manejar documentos Word.

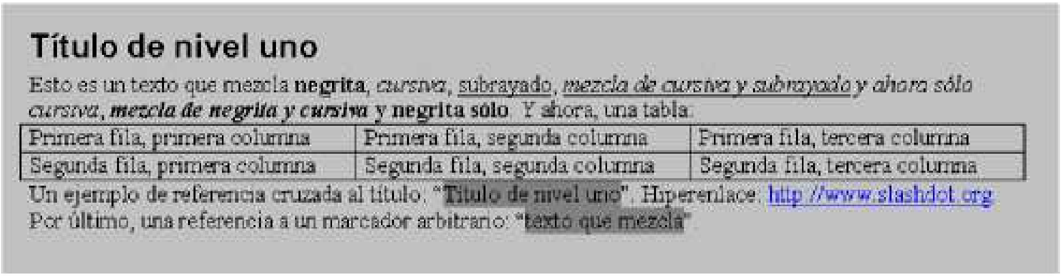
6 Método de trabajo

6.1 Los medios

Los medios de los que se dispuso para hacer el proyecto fueron la escasa documentación oficial de Microsoft, el programa Microsoft Word, para crear documentos de prueba con los que suplir las omisiones y fallos de la especificación, varios manuales de HTML y el potente y versátil entorno de programación de GNU/Linux.

6.2 Orientación a objetos

Como ya se ha mencionado, se han utilizado lenguaje C++, STL, patrones de diseño y UML como herramientas para elaborar *libFreeRTF*. Nos resulta llamativo que, a pesar de la disponibilidad de C++ desde hace muchos años, se siguen escribiendo tantas bibliotecas con interfaces en C simple, que entorpecen la reusabilidad.



Título de nivel uno

Esto es un texto que mezcla **negrita**, *cursiva*, subrayado, *mezcla de cursiva y subrayado* y *ahora sólo cursiva*, ***mezcla de negrita y cursiva*** y **negrita sólo**. Y ahora, una tabla:

Primera fila, primera columna	Primera fila, segunda columna	Primera fila, tercera columna
Segunda fila, primera columna	Segunda fila, segunda columna	Segunda fila, tercera columna

Un ejemplo de referencia cruzada al título: "Título de nivel uno". Hiperenlace: http://www.slashdot.org
Por último, una referencia a un marcador arbitrario: "texto que mezcla"

Figura 3: Documento original visto con Word 97

6.3 Sólo software libre

Todo el trabajo, excepto la creación de documentos de prueba, se desarrolló en una computadora personal tipo Pentium, con Debian GNU/Linux [6]. El programa se creó con el compilador *g++*, las funciones y clases de la *libc* y *libstdc++*, y el constructor de analizadores léxicos *flex++*. La documentación se escribió en LaTeX, con el editor *vim*. Los diagramas se dibujaron con el programa *dia*. Para ver e imprimir la documentación, se utilizaron el *ghostview* y el *xdvi*, y el sistema de impresión de BSD, junto con el paquete *magicfilter*. En definitiva, *todas* las herramientas utilizadas para construir el programa y la documentación fueron completamente libres, la mayoría de ellas distribuidas por la licencia GPL. Las únicas herramientas no libres utilizadas fueron el Microsoft Word, para crear los documentos RTF de prueba, y el Netscape Navigator, que es gratuito, para ver los resultados de la conversión.

6.4 Dificultades encontradas

El estudio del formato RTF fue en gran medida *empírico*. A menudo, la información del formato se dedujo de los análisis de documentos escritos para tal efecto. Esto es debido a la pobre especificación oficial, plagada no sólo de omisiones alarmantes, sino también de algunas equivocaciones.

Otra de las dificultades encontradas fue la propia arquitectura del formato. Éste es innecesariamente complicado y bastante anárquico, y está lleno de casos particulares. Por si fuera poco, al ser un formato pensado para programas y no para personas, se basa principalmente en estilos físicos que casi lo acercan más al PostScript que al HTML o al LaTeX. El formato está hecho a medida de las necesidades de Microsoft Word, no pensado para su transportabilidad; por ello muchas características estructurales resultan extrañas o engorrosas.

7 Ejemplos de uso

A continuación se presentan dos ejemplos de salidas producidas por RTHC: uno de comparación entre el texto original y el convertido, y otro de demostración de los índices flotantes.

El primero se compone de dos capturas de pantalla. La primera, que puede verse en la figura 7, es el documento representado por Word 97. La segunda es la conversión del documento, representada por el navegador Netscape 4.73 de Linux. Se puede comprobar cómo todo el formato se mantiene perfectamente, así como el texto.

El segundo ejemplo muestra una ventana de índice flotante, con la que se puede recorrer el documento mientras se lee en la ventana principal.

8 Estado actual del proyecto

Actualmente, RTHC es capaz de convertir texto con formato, tablas simples, listas numeradas y no numeradas y enlaces y referencias cruzadas. La omisión más importante son las imágenes y diagramas, aunque tampoco se reconocen los atributos complejos de las tablas ni otros elementos de menor importancia, como los objetos incrustados.

Hay una página sobre el proyecto en la dirección `http://so.dis.ulpgc.es/ a2092/rthc`, y se invita a toda la comunidad a colaborar mejorando tanto *libFreeRTF* como el propio programa RTHC.

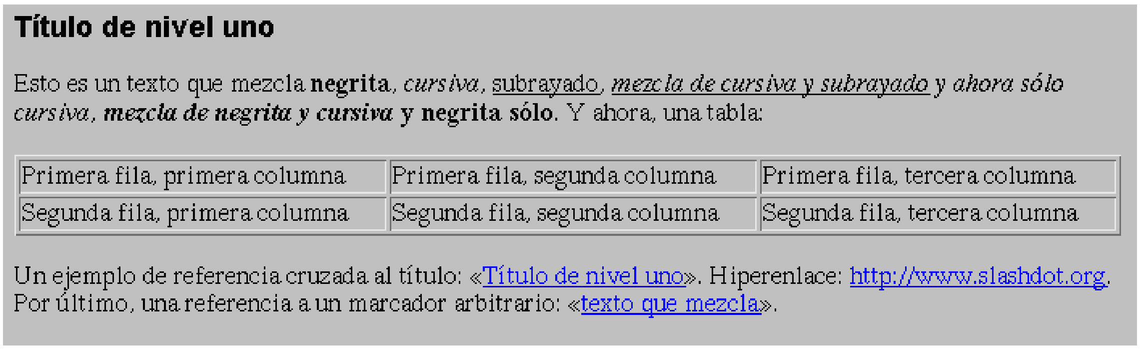


Figura 4: Documento convertido visto con Netscape

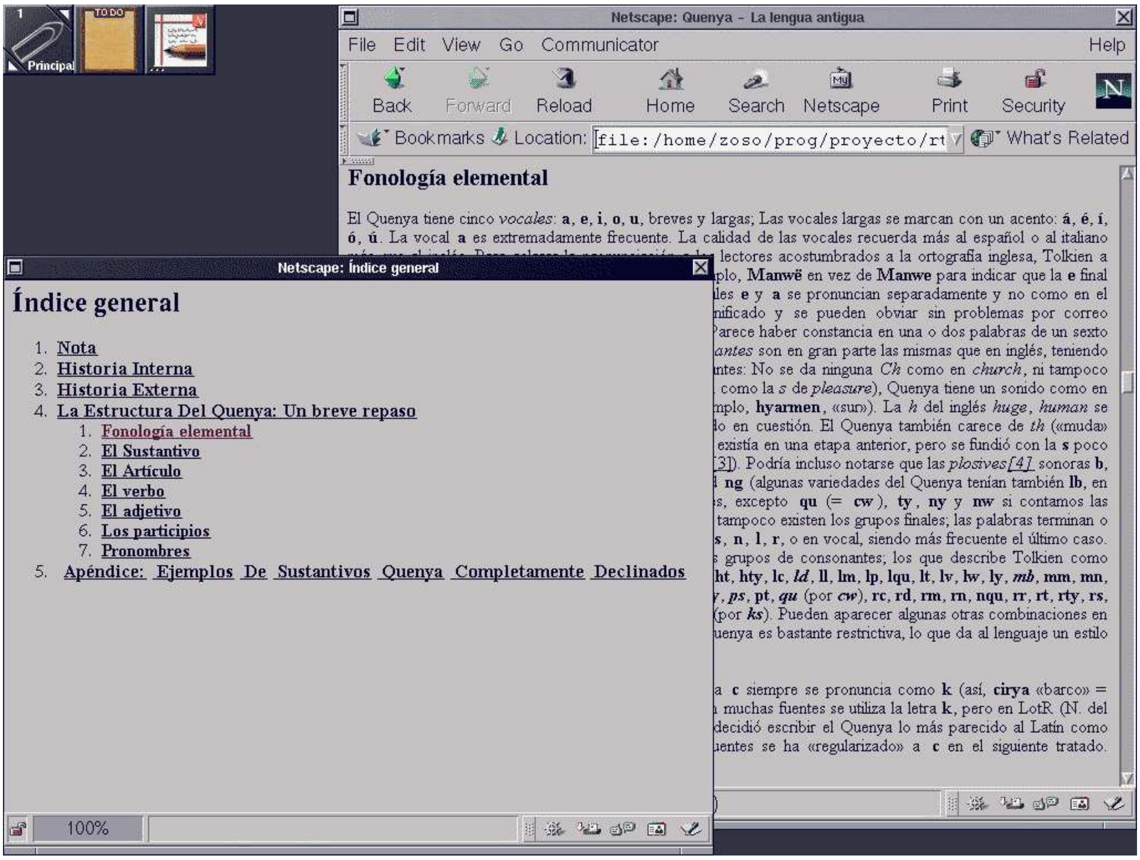


Figura 5: Ejemplo de ventana de índice flotante

9 Conclusiones

Tras el satisfactorio desarrollo de este trabajo, podemos extraer las siguientes conclusiones:

- Gracias a RTHC, a comunidad del software libre dispone de una biblioteca software con la que manejar fácilmente documentos RTF.
- Se ha puesto a disposición una extensión de la documentación que aclara oscuridades de la confusa especificación original.
- RTHC está diseñado para ser legible, reusable y extensible.
- Para desarrollar RTHC sólo se ha usado software libre, lo cual demuestra el potencial de este enfoque.

10 Referencias

1. «Rich Text Format (RTF) Specification», de Microsoft Corp.
2. Página oficial de AbiSource, autores del AbiWord (`http://www.abisource.com`)
3. «Design Patterns. Elements of Reusable Object-Oriented Software» Gamma, Helm, Johnson, Vlissides. 1995.
4. «STL tutorial & reference guide: C++ programming with the standard template library» David R. 1996, Addison-Wesley.
5. «The Unified Modeling Language User Guide» Booch, Rumbaugh, Jacobson.
6. Página oficial de la distribución Debian (`http://www.debian.org`)
7. Una lista de descripciones de formatos de ficheros en la Red (`http://www.wotsit.org`)
8. Un proyecto parecido, con el formato nativo de Word `http://www.wvware.com`