

ELABORACIÓN DE RECURSOS DIDÁCTICOS CON GUADALINEX



Paco Villegas

5 de marzo de 2006

Índice general

7. Editores HTML. Integrando nuestros apuntes.	2
7.1. Introducción	2
7.2. Editores HTML	3
7.2.1. NVU	4
7.2.2. Más editores de HTML	11
7.3. Integrando nuestros apuntes.	12
7.3.1. Introducción	12
7.3.2. Editor Reload	13
7.4. Ampliación: Scribus	25

Capítulo 7

Editores HTML. Integrando nuestros apuntes.

Siempre me ha fascinado Internet, incluso antes de que existiera.

(*Servidor Apache*, RICH BOWEN & KEN COAR, Prentice Hall)

7.1. Introducción

Está claro que Linux se lleva muy bien con Internet, van cogidos de la mano. Disponemos de varios navegadores tanto en modo texto (`lynx`) como en modo gráfico (Mozilla, Netscape, Amaya, Konqueror, etc). De hecho, desde que comenzó el curso, seguro que estas herramientas han sido algunas de las más usadas. Pero antes de meternos de lleno en el tema, aclararé algunos de los conceptos que aparecerán a lo largo de este apartado¹.

¿Qué es HTML? *HyperText Markup Language*² o HTML es un lenguaje utilizado para desarrollar páginas Web y documentos. HTML no es un lenguaje de programación; se trata más de un lenguaje de marcación, lo suficientemente flexible para mostrar texto y otros elementos (gráficos, sonido, etc) en una amplia variedad de visores³.

HTML indica a los navegadores Web cómo deben mostrar los documentos Web; sin embargo, la información que HTML ofrece sobre el formato es muy general. Cuando escribamos documentos HTML hay que tener en cuenta que su aspecto dependerá del navegador que esté usando la persona que visualiza la página Web. Todos los navegadores Web tratan de formatear los documentos HTML "lo mejor posible" sin embargo, lo que vemos, puede que no sea lo que vea otra persona que esté usando un navegador distinto (o el mismo pero con otro sistema operativo).

La sintaxis básica de HTML está compuesta de tres elementos: etiquetas, atributos y URLs. Estos componentes se refieren respectivamente al formato y presentación de la información, a los detalles de ciertas acciones y a la ubicación de otros archivos y documentos.

Etiquetas Las etiquetas son los bloques básicos de construcción de HTML. Son la parte de HTML que indica a los navegadores Web cómo tienen que mostrar el texto o los gráficos y les proporciona otro tipo de información sobre el formato. Las etiquetas se escriben entre los símbolos mayor que y menor que (`<etiqueta nombre>`).

La mayoría de las etiquetas constan de una etiqueta de inicio y otra de final (`</etiquetanombre>`).

¹Para seguir con el tema no es necesario ser un especialista en páginas Web, pero está claro que aclarar un poco en qué consiste este lenguaje no está de más.

²Lenguaje de marcación de hipertexto

³En la vida diaria se utiliza este tipo de lenguaje en todo momento, pero probablemente no lo llamamos así. Cada vez que subrayamos o queremos poner énfasis en un escrito estamos utilizando un lenguaje de marcación.



Atributos Algunas veces, las etiquetas necesitan especificar información exacta como, por ejemplo, dónde está ubicado un archivo. Los atributos se utilizan con etiquetas para proporcionar más información sobre cómo debe implementarse la etiqueta. Por ejemplo, sea la siguiente etiqueta: ``

`` es una etiqueta que le dice al navegador que muestre una imagen gráfica. Pero, ¿qué imagen? Aquí es donde interviene el atributo. En el ejemplo, el campo del atributo es `SRC= «ejemplo.gif»`. Éste proporciona información detallada sobre cómo debe ser interpretada la etiqueta ``. En este caso, el archivo `ejemplo.gif` va a ser la imagen a mostrar⁴.

URL Se accede a los recursos de la Web mediante una dirección descriptiva conocida como una *uniform resource locator*⁵. Todo a lo que se accede en la Web tiene un URL. HTML utiliza URL para especificar la ubicación de los archivos necesarios, o de otras páginas Web conectadas por medio de enlaces de hipertexto. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia) es conocido por su URL.

Un documento HTML típico está compuesto de dos secciones lógicas: la cabecera y el cuerpo. La cabecera contiene la información sobre el documento; el cuerpo, la información misma del documento.

- la sección de la cabecera está enmarcada por etiquetas `<HEAD>` y `</HEAD>`. La sección cabecera de una página HTML contiene información relacionada con el mismo documento. Podemos colocar diversas etiquetas en la sección cabecera, aunque la que más se usa es `<TITLE>`. El título del documento, siempre que se halle enmarcado por las etiquetas `<TITLE>` y `</TITLE>`, aparece en el título de la barra del navegador Web.
- el cuerpo lo está por etiquetas `<BODY>` y `</BODY>`. La sección cuerpo de una página HTML contiene la mayoría de los elementos que pueden verse al visualizar la página con un navegador Web. Ahí es donde se introducen todos los elementos que se desea que vean los demás: texto, gráficos, enlaces (URLs), listas, tablas, ...

7.2. Editores HTML

Cualquier editor puede crear HTML. No deje que nadie le diga lo contrario. (*HTML 4*, ANNE PHILIPS)

Como HTML es un lenguaje de marcación basado en ASCII, lo único que necesitamos es un editor que nos permita grabar los archivos en el formato ASCII y un navegador Web con el que poder visualizar las páginas Web mientras las realizamos. No necesitamos estar conectados a Internet para crear páginas Web. Cualquier navegador Web nos permite abrir un archivo HTML local y visualizarlo como si lo hubiera sacado de Internet.

Existen herramientas para crear documentos HTML sin tener que conocer las etiquetas del HTML y que permiten realizar páginas Web de forma cada vez más sencilla. Varias de las nuevas herramientas evitan que tengamos que utilizar los códigos del HTML. Todo lo que tenemos que hacer es escribir las palabras, formatear el documento como si de un tratamiento de textos se tratase y guardarlo en el lugar apropiado. Así que si nuestro objetivo es crear una Web sencilla, la forma en que lo podemos hacer es usar alguna de las herramientas estudiadas⁶ y usar la posibilidad de exportación a formato html:

- HotPotatoes, Wink, LyX, ... se estudió en los temas en los que se trataron
- OpenOffice merece mención especial ya que incorpora un editor de HTML. En cualquier caso, en general, para conseguir ficheros en formato HTML, lo conseguiremos desde **Archivo**→**Exportar** o **Archivo**→**Enviar**→**Crear documento HTML**.

⁴Para conseguir que el navegador muestre con más rapidez el gráfico, hay que usar los atributos largo y ancho con la etiqueta ``.

⁵Localizador Universal de Recursos. Por ejemplo: <http://www.picasa.org>

⁶Si no queremos usar un editor HTML o si ya disponemos de los documentos que queremos incluir en la Web, podemos convertirlos a HTML con cualquiera de los programas estudiados.



Aunque todas estas herramientas hacen que el escribir con HTML resulte menos dificultoso, no permiten hacer todo lo que uno quisiera. Es útil usar un editor HTML. De todas maneras, si deseamos controlar el resultado final, casi siempre tendremos que editar directamente el HTML para lograr el aspecto deseado.

7.2.1. NVU

NVU es un editor de páginas web multiplataforma basado en Mozilla Composer, pero de ejecución independiente, y añade características nuevas como soporte integrado de CSS y mejor gestión del protocolo FTP para actualización de los ficheros. La página principal del proyecto es <http://www.nvu.com/>.

<http://nave.escomposlinux.org/productos/nvu/>

NVU es un programa que nos va a permitir escribir nuestras páginas Web sin necesitar conocer de forma exhaustiva la sintaxis del lenguaje HTML. Se trata de una especie de Mozilla Composer mejorado⁷, cuyo objetivo es permitirnos crear de forma fácil y rápida nuestras páginas Web.

Entre sus características merece la pena destacar:

- Dispone de un editor de hojas de estilo
- Integra un *Site Manager* (administrador de archivos del proyecto) desde el que podemos gestionar nuestros ficheros de forma local o remota (vía FTP)
- Asistente para la creación y edición de tablas.
- Asistente para la creación de formularios.
- Corrector ortográfico
- Una buena guía de uso (en inglés) accesible desde el menú **Ayuda**

Instalación

Hay que bajarse el paquete⁸ `nvu-1.0.0.x86.package`. Podéis hacerlo desde el Moodle del curso (también tendréis que bajar el fichero `autopackage.tar.bz2`) o desde <http://www.nvu.com/download.html> Una vez bajado hay que ejecutar el comando (si lo hemos hecho desde `nvu.com` necesitaremos conexión a internet porque durante la instalación necesita bajarse el archivo `autopackage.tar.bz2`)

```
$sh nvu-1.0.0.x86.package
```



Este método es válido tanto para los centros de enseñanza como para los ordenadores en los que seamos el root del sistema. La única diferencia es que en el momento que nos lo solicite podemos poner la contraseña del root u optar por **No Password** si no la conocemos.

Al ejecutar el comando nos avisa que necesita bajarse el paquete de internet o, si ya lo tenemos lo usará, en ambos casos aceptamos con Y

⁷De hecho se basa en él.

⁸

- También podemos optar por bajarnos el fichero `nvu-1.0.es-ES.linux-i686.tar.bz2` y descomprimirlo en nuestro *home* de usuario.

```
$tar xjf nvu-1.0.es-ES.linux-i686.tar.bz2
```

Para ejecutarlo solo hemos de situarnos en el directorio creado y ejecutar

```
$/nvu
```

- Existen paquetes para Debian Sarge que no funcionan sin actualizar Guadalinex2004.



Uso

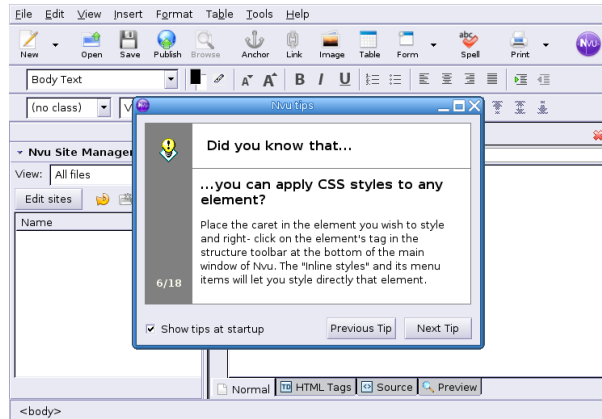
Para ejecutarlo, lo hacemos desde un terminal⁹

```
$nvu
```

o desde la entrada que aparece en¹⁰

Menú Debian ▷ **Application** ▷ **Network** ▷ **NVU**

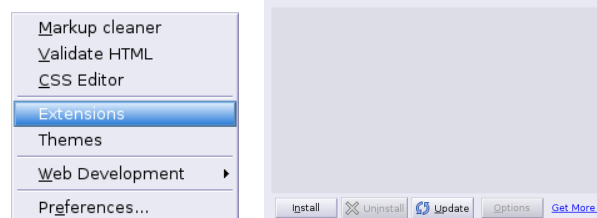
La primera vez que lo abrimos, una ventana emergente nos solicita que pulsemos sobre el botón **OK** para que haga un ping y aumente el contador del número de descargas/usuarios del programa.



Castellanizarlo Está en inglés, así que pongámoslo en castellano. Para eso necesitamos el fichero `nvu-1.0.es-ES.langpack.xpi`, lo podéis bajar del Moodle del curso.

Para instalarlo:

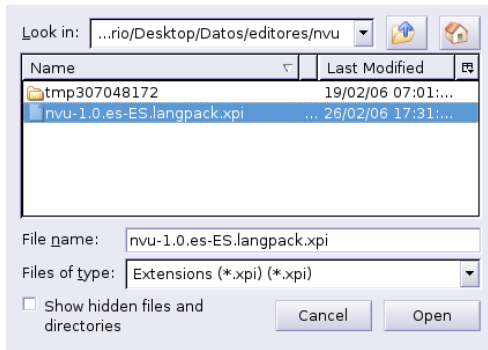
1. Seleccionaremos la opción de menú **Tools**→**Extensions** y se mostrará una ventana con las extensiones instaladas (en nuestro caso ninguna). Pulsaremos ahora sobre el botón **Install**.



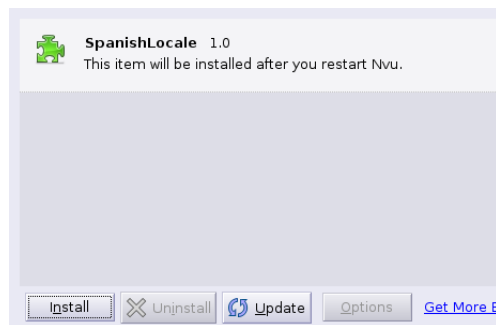
2. Se abrirá una ventana que nos va a permitir navegar por el sistema de archivos. Debemos situarnos en el directorio que contiene la extensión para el idioma castellano (`nvu-1.0.es-ES.langpack.xpi`) y seleccionarlo. Pulsamos sobre **Open** y aparecerá una nueva ventana en la que se nos avisa de que la extensión no está firmada y solicitándonos confirmación para instalarla.

⁹Para que funcione correctamente, después de instalarlo, hay que reiniciar el terminal o ejecutar `exec bash`

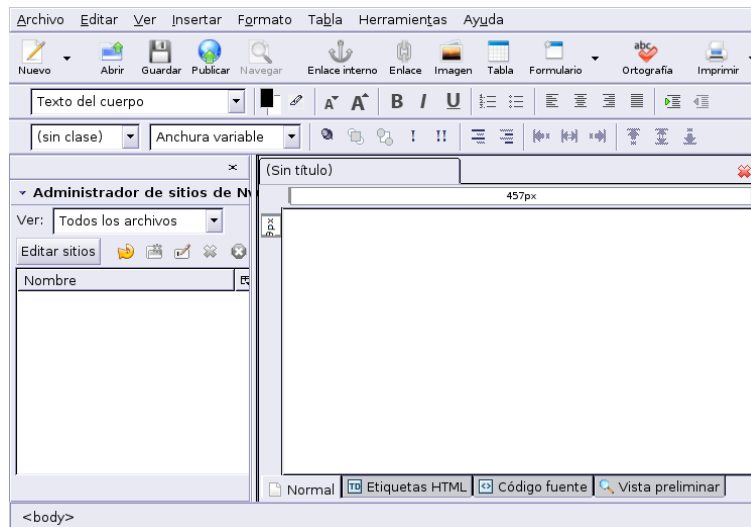
¹⁰Sólo en el caso de que lo hayamos instalado como root.



Pulsaremos sobre el botón **Install now** y se nos mostrará la ventana con las extensiones que se instalarán. Para terminar cerramos la ventana y el programa



Al reiniciar nvu ya deberían aparecer los menús en castellano.



Su interfaz es intuitiva y está organizada de la forma habitual: la barra de menú, tres barras de herramientas que nos facilitan la elaboración de nuestra Web, a la izquierda se nos muestra el administrador de sitios (en él se nos mostrarán los archivos que componen nuestro proyecto) y, por último a su derecha la zona de edición.

Notar que desde las pestañas inferiores tenemos la posibilidad de trabajar en varios modos: Normal, Etiquetas HTML, Código fuente o Vista Preliminar.



Práctica: Mi primera Web con NVU

En esta práctica vamos a usar un texto que podéis encontrar en http://wiki.gleducar.org.ar/wiki/index.php/Razones_para_usar_software_libre_en_educaci%C3%B3n se trata del texto:

Razones para usar software libre en educación

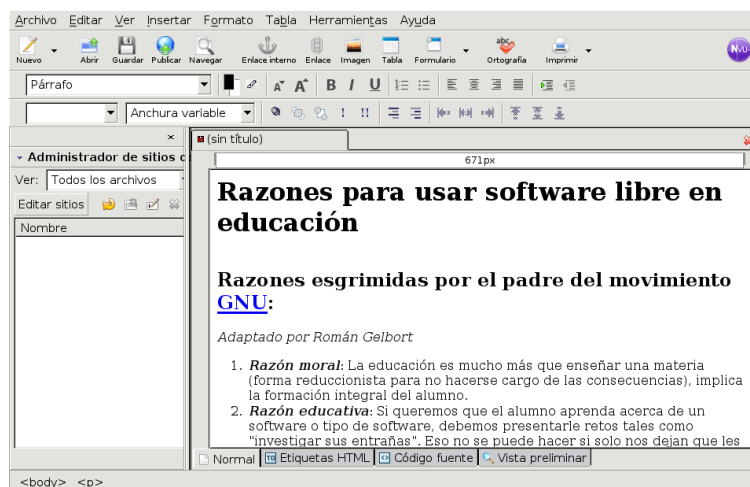
Razones esgrimidas por el padre del movimiento GNU:

Adaptado por Román Gelbort


1. **Razón moral:** La educación es mucho más que enseñar una materia (forma reduccionista para no hacerse cargo de las consecuencias), implica la formación integral del alumno.
2. **Razón educativa:** Si queremos que el alumno aprenda acerca de un software o tipo de software, debemos presentarle retos tales como "investigar sus entrañas". Eso no se puede hacer si solo nos dejan que les enseñemos a apretar botones, se requiere un mayor compromiso.
3. **Razón libertaria:** No hay mejor forma de enseñar las bondades de la vida en libertad que la de ejercerla. Si mis alumnos tienen que sufrir que un software libre no está tan desarrollado como el homólogo privativo, sufrirán por su escaso desarrollo o aprenderán a entrar en la comunidad de desarrollo y verán como el programa se vuelve eficiente gracias a ellos. Es la diferencia entre libertad y libertinaje... una tiene en cuenta la responsabilidad y la otra no.
4. **Razón económica:** Esta es la menos importante de todas las razones... aunque la más vista desde antes de entender al software libre. Un sistema operativo libre, como Linux, es mucho más barato que su homólogo privativo. No porque no se paguen licencias, sino porque no tenemos que pasarnos la vida arreglándolo para que siga funcionando.

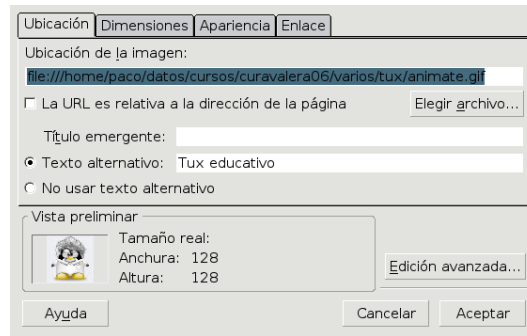
Ejemplo: En un colegio, en el que trabaja un miembro del proyecto (aunque estos ejemplos abundan), compraron 3 PCs con Windows XP y se conectaron a Internet. A las 2 semanas... ya se empezaron a colgar; antes del mes hubo que empezar a reinstalar el sistema por problemas con los virus que pasaron por encima del antivirus "actualizado". En Totoras (Santa Fe) hace muchos meses que tienen instalado y funcionando Linux sobre LTSP en 15 computadoras Pentium MMX, sobre un servidor único... ¿qué es colgarse? ¿qué es reinstalar? ¿qué son los virus?

Lo primero que haremos será seleccionar ese texto y copiarlo a nuestra página. Como podremos observar, al copiarlo desde una Web se mantiene el mismo formato que en la página original. Si deseamos cambiarle algo, sólo hemos de usar los iconos de las barras de herramientas o las barras de menús.

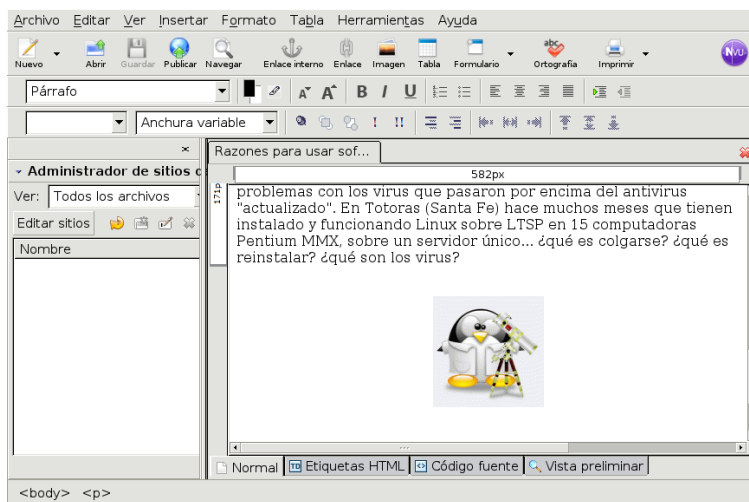


Introduzcamos un gráfico (el que ha servido de logo del curso). Se trata del fichero `animate.gif` que podéis obtener desde la Web del curso. Lo bajamos a nuestro ordenador y después, pulsando sobre el

icono  se selecciona.



Es importante marcar la casilla **La URL es relativa a la dirección de la página** así como que no se nos olvide el **Texto alternativo**. En la captura que sigue, además se observa que lo he centrado.



Debajo del gráfico voy a añadir un hipervínculo que indique de dónde he sacado la página Web. Para eso voy al menú **Insertar**→**Enlace**, se abre una ventana en la que escribo el texto (en este caso el texto va a ser la propia dirección web) en el campo **Texto del enlace** y la URL a la que se debe enlazar en el campo **Ubicación del enlace** (copiando y pegando la dirección que acabamos de escribir).



Texto del enlace
http://wiki.gleducar.org.ar/wiki/index.php/Razones_para_u...

Ubicación del enlace
Introduzca una dirección web, un archivo local, una dirección de correo electrónico o seleccione un enlace interno o una cabecera de la lista desplegable:
http://wiki.gleducar.org.ar/wiki/index.php/Razones_para_usar_software_libre

Lo anterior es una dirección de correo electrónico
 La URL es relativa a la dirección de la página

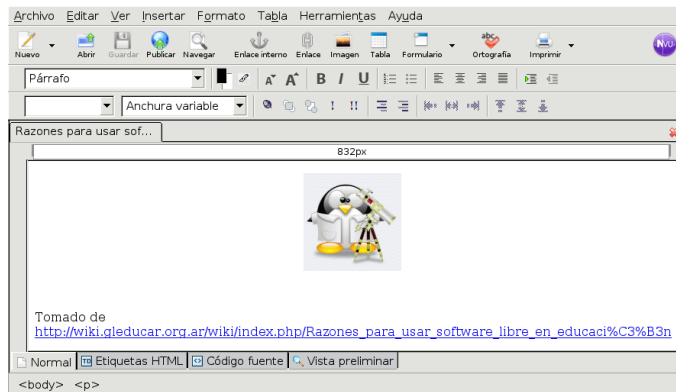
▲ Menos propiedades

Etiquetas Technorati
 Ésta es una etiqueta Technorati

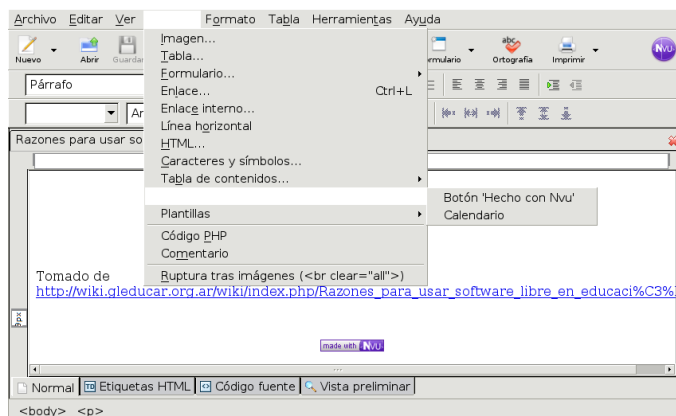
Destino
 El enlace se abrirá
en una nueva ventana

Amistad (XFN)
 La ubicación del enlace es otra dirección mía
Amistad amigo conocido contacto ninguno
Física conocido personalmente
Profesional colaborador colega
Geográfica co-residente vecino ninguno
Familia hijo padre hermano cónyuge pariente ninguno
Romántica musa amor compromiso novio/a

Al pulsar sobre el botón **Aceptar** aparecerá el hiperenlace en la página que estamos creando.

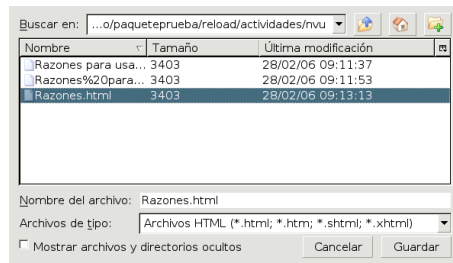


Por último insertemos un botón  que nos indique que está realizado con NVU. Lo hacemos desde el menú **Insertar**→**Componentes reutilizables**.





Ya solo nos falta salvar nuestro proyecto pulsando sobre **Archivo**→**Guardar** o desde el icono **Guardar** de la barra de herramientas.



Para saber cómo queda podéis revisar el fichero de nombre `Razones.html` desde Moodle.

7.2.2. Más editores de HTML

Tenemos multitud de editores de HTML para Linux (quanta, asWedit, screen, ...). En este apartado me centraré sólo en comentar dos de los más usados para este menester, se trata de Mozilla Composer y bluefish.

Mozilla Composer

Se trata del editor de páginas web de Mozilla. Desde el navegador Mozilla con **Ventana**→**Composer**, o desde el escritorio con

 **Aplicaciones** ▷ **Internet** ▷ **Mozilla** ▷ **Mozilla Editor**, accedemos a un sencillo editor de páginas web que es una evolución mejorada del Composer de Netscape.



Mi primera Página Web

Escrita con Mozilla Composer



El editor que trae permite seleccionar el texto que deseamos formatear y acceder a continuación a las etiquetas HTML desde un menú, en lugar de teclearlas el usuario.

No creo conveniente detenerme en comentar su uso, ya que existen multitud de cursos¹¹ que explican con detalle qué se puede hacer con él.

Bluefish

La página principal es <http://bluefish.openoffice.nl/>. Es con el que trabajo habitualmente. Me parece sencillo, potente y me da todo lo que necesito para realizar mis páginas Web. En contra frente a los anteriores: es necesario conocer el lenguaje HTML para poder sacarle todo el partido.

Lo encontramos en:

 **Aplicaciones** ▷ **Internet** ▷ **Bluefish**

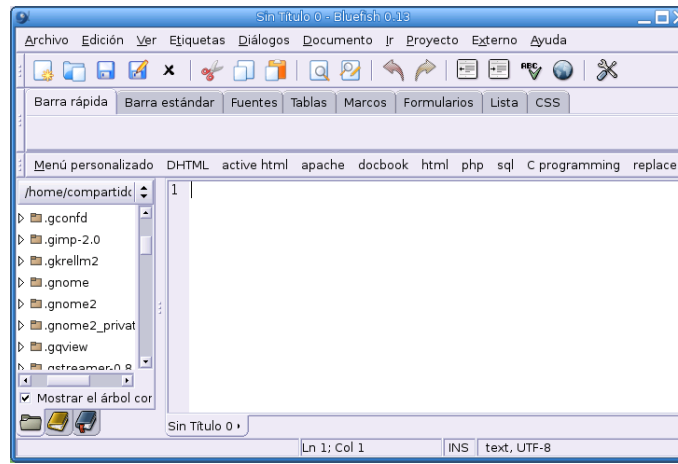
 **Aplicaciones** ▷ **Menú Debian** ▷ **Aplicaciones** ▷ **Editores** ▷ **Bluefish**

O, como es habitual, desde una xterm:

¹¹La ayuda en línea del programa es excelente.



```
$ bluefish &
```



Cuando se abra el programa tendremos un entorno para escribir HTML que permite insertar las etiquetas directamente y que nos facilita el trabajo con menús y ventanas para gestionar los atributos de éstas. Además, podemos ver cómo quedaría nuestra página si pulsamos en el menú **Externo** y seleccionamos **Externo Mozilla**¹². Merece la pena probarlo.

A destacar: Multidocumento con pestañas (al estilo del gedit), resaltado de la sintaxis, soporte para lenguajes de script (p.e. PHP), previsualización en el navegador, corrección ortográfica a través de ispell, etc.

Podemos encontrar un manual en *inglés* en <http://bluefish.openoffice.nl/manual/>

7.3. Integrando nuestros apuntes.

7.3.1. Introducción

La extensión del uso de ordenadores e internet en el ámbito educativo (*e-learning*), ha abierto una posibilidad muy interesante para la enseñanza: el uso de “plataformas educativas”.

LMS es un acrónimo de *Learning Management System*¹³. Hablamos de un LMS (en la nomenclatura de los centros TIC lo podemos trasladar por el de “plataforma”) cuando nos referimos a un software “que facilita el desarrollo del aprendizaje distribuido a partir de información de muy diversa índole, utilizando los recursos de comunicación propios de Internet, al tiempo que soportan el aprendizaje colaborativo en cualquier lugar y en cualquier momento” <http://prometeo3.us.es/publico/jsp/introPlataformas.jsp?mn=4>

Un ejemplo claro de LMS es en el que estamos desarrollando esta actividad formativa, se trata de un LMS de código abierto denominado Moodle. En general se trata de una aplicación que se instala en un servidor Web y permite que alumnos, profesores y demás miembros de la comunidad “virtual” puedan, a través de ella, realizar actividades relacionadas con actividades formativas: se trata pues de un lugar en que se deben poder descargar contenidos, que permita mecanismos de comunicación entre profesores y alumnos (foros, chat, ...), que incorpore sistemas de evaluación, ... Evidentemente, para que esto pueda ser así, debe disponer de mecanismos de administración que permitan la gestión de los cursos, así como de la matriculación de alumnos, asignación de profesores, ...

Bien, admitamos que la administración es específica de nuestra plataforma (en general como profesores esto no debería preocuparnos ya que no debería ser nuestro cometido), que nos habituamos a distintas

¹²O en el icono correspondiente

¹³Sistema de Gestión de Aprendizaje



formas de comunicación y evaluación. Lo que sí que no deberíamos estar dispuestos a asumir es que el trabajo realizado para desarrollar contenidos para un sistema de gestión de aprendizaje no se pueda usar en otro. Este problema se está intentado resolver desde hace algún tiempo creando varios estándares de paquetes que puedan ser usados por los distintos LMSs existentes. Entre esos estándares sobresale el estándar SCORM¹⁴ (*Sharable Content Object Reference Model*¹⁵)

SCORM es un estándar que se centra en los contenidos, en el que se establecen una serie de especificaciones para la creación de objetos de aprendizaje reutilizables en las distintas plataformas. De esa manera, si creamos un contenido que se ajuste al modelo SCORM, no deberíamos tener ninguna dificultad en poder usarlo en cualquier LMS que lo soporte.

Pero ¿qué podemos entender por eso de objetos de aprendizaje?, pues todo lo que hemos elaborado en el curso (y mucho más). Es decir, usando SCORM podremos integrar todas las actividades realizadas hasta ahora integrándolas en un solo paquete que, además, al estar basado en un estándar podrá ser usado en cualquier plataforma que se precie.

Internamente los paquetes SCORM son unos ficheros comprimidos en formato zip, que además de contener las actividades, deben contener un fichero de manifiesto (`imsmanifest.xml` está en formato XML y consta de cuatro secciones: Metadatos, Organizaciones, Recursos y SubManifiestos) en él se establecen además de los contenidos, el orden en que deben ir trabajándose para lograr los objetivos, tal y como fueron diseñados por la persona que creó la actividad.



- Un excelente documento para saber más sobre todo esto es una tesina de MIGUEL REBOLLO PEDRUELO de título: *El estándar SCORM para EaD* y que está a vuestra disposición en el Moodle del curso.

- Sobre **e-learning** una serie de Web interesantes:

<http://es.wikipedia.org/wiki/E-learning>

y en ella se enlaza a dos que creo merece la pena visitar se trata de

<http://prometeo3.us.es/publico/index.jsp>

<http://www.elearningworkshops.com/index.php>

Pero, por mi parte, dejaré la teoría, y veamos todo esto de forma práctica.

7.3.2. Editor Reload

Para realizar paquetes reutilizables de aprendizaje existen varias utilidades, pero dentro del software libre la que sobresale es Reload. Reload es un programa realizado en Java de libre distribución y cuya web principal es

<http://www.reload.ac.uk/>

Instalación

Podemos descargar el programa de <http://www.reload.ac.uk/editor.html> o directamente desde el enlace que hay en la web del curso.

El fichero necesario es

`Setup_ReloadEditor202_lnx.bin`

Una vez en nuestro poder, para instalarlo hay que ejecutar desde una xterm:

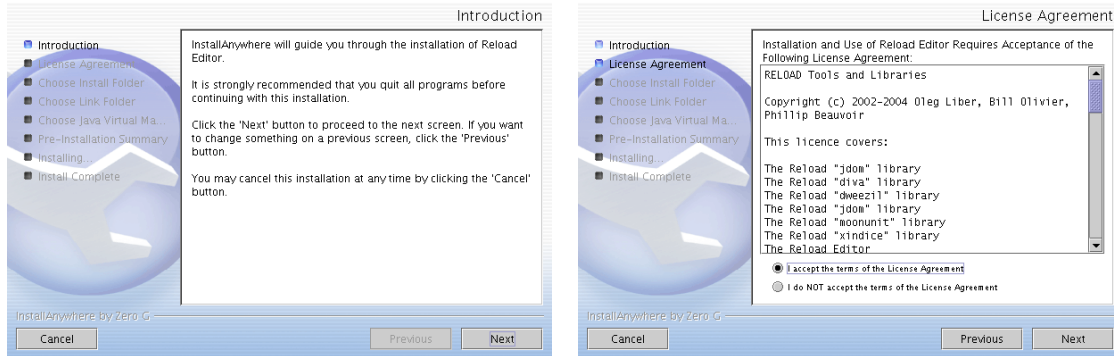
¹⁴La iniciativa ADL (*Advanced Distributed Learning*) se inicia en el año 1997 por parte del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca con el objetivo de crear un estándar para el desarrollo de formación educativa basada en las TIC.

¹⁵Modelo de Referencia de Contenido Compartido

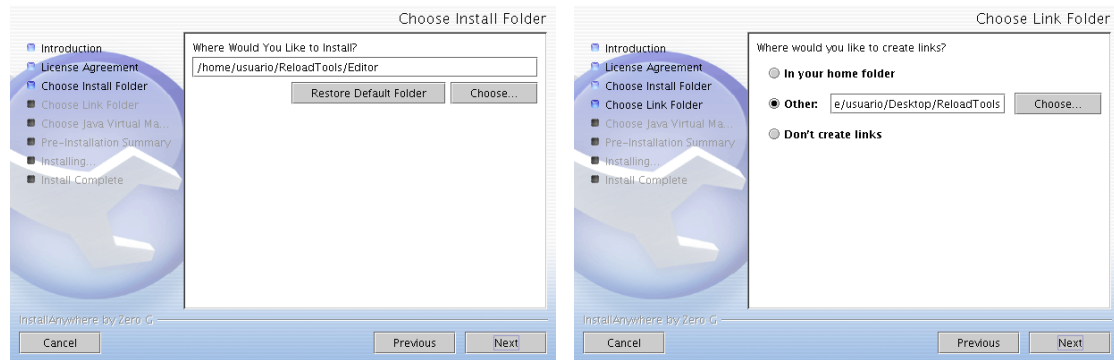


```
$ssh Setup_ReloadEditor202_lnx.bin
```

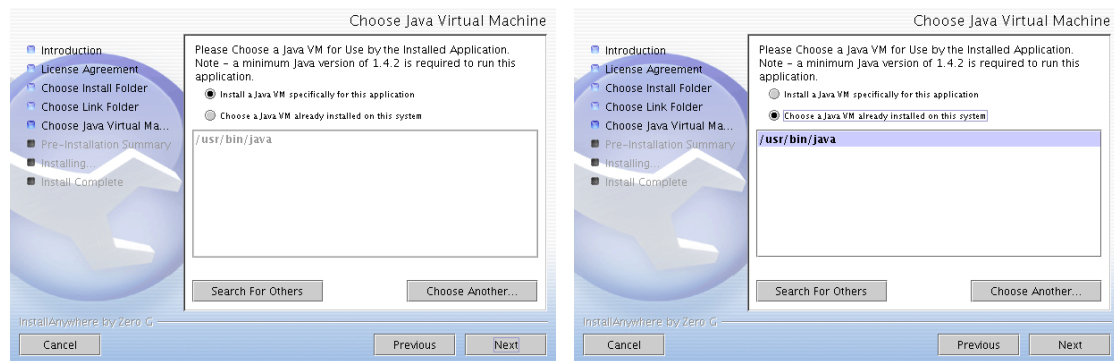
Se iniciará entonces el proceso de instalación, es simple y está muy guiado a través de la secuencia de ventanas que van apareciendo. Pulsamos sobre **Next** en la primera de las ventanas y en la siguiente marcamos la casilla de aceptación de la licencia.



En las ventanas donde nos pregunta por la carpeta para instalar y para crear los enlaces dejamos los valores por defecto.

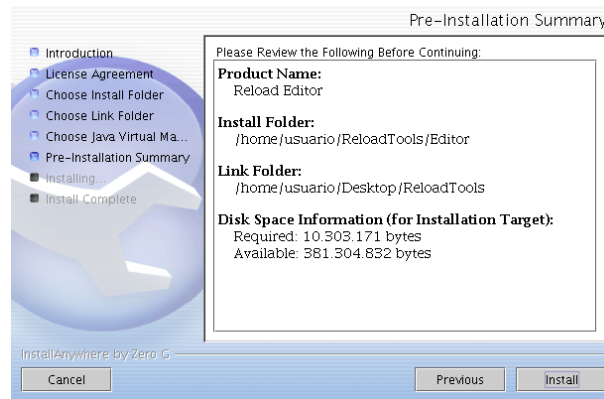


Seguimos adelante y aparece una ventana en la que, por defecto, está marcada la casilla de instalar Java. Parto de la idea de que tenemos Java instalado, así que marco la opción de usar el que ya hay disponible en el sistema¹⁶.

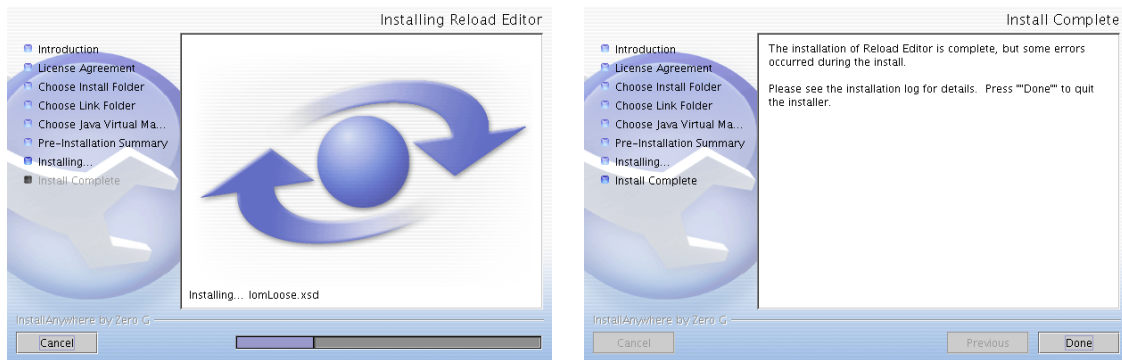


En la siguiente ventana nos informa de los datos de configuración para la instalación de la aplicación así como del espacio necesario.

¹⁶Si no es así, dejaríamos la opción por defecto.



Tras pulsar sobre el botón **Install** comienza la instalación (tardará un poquito)




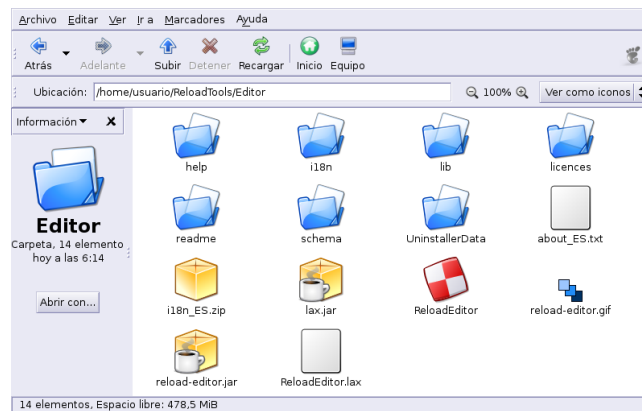
Castellanizar Reload El programa se instala en inglés, para que el entorno esté en castellano hemos de bajarnos el archivo `i18n_ES.zip` (lo he puesto en Moodle). Una vez que lo hemos bajado hay que copiarlo en la ruta adecuada (con el comando `cp`), situarnos en el directorio donde lo hemos copiado (`cd`) y descomprimirlo (`unzip`). La secuencia de órdenes que tendremos que escribir en el terminal se corresponde con las tres primeras líneas:

```
usuario@guadalinux:~$ cp i18n_ES.zip ReloadTools/Editor/
usuario@guadalinux:~$ cd ReloadTools/Editor/
usuario@guadalinux:~ReloadTools/Editor/$ unzip i18n_ES.zip
Archive: i18n_ES.zip
replace i18n/uk/ac/reload/dweezil/messages.properties? [y]es, [n]o, [A]ll,
[N]one, [r]ename: A
inflating: i18n/uk/ac/reload/dweezil/messages.properties
inflating: i18n/uk/ac/reload/dweezil/messages_es.properties
inflating: i18n/uk/ac/reload/editor/messages.properties
inflating: i18n/uk/ac/reload/editor/messages_es.properties
inflating: i18n/uk/ac/reload/moonunit/messages.properties
inflating: i18n/uk/ac/reload/moonunit/messages_es.properties
inflating: about_ES.txt
```

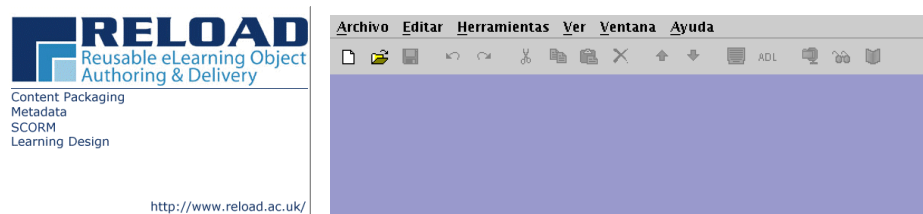
Sólo una aclaración: al escribir la letra A le obligamos a sobrescribir los ficheros que ya existen en el sistema.

Uso

Para ejecutarlo, basta con pulsar sobre el icono del escritorio  o bien, con Nautilus, nos situamos en el directorio adecuado y se pulsa sobre `ReloadEditor`



Se abre el programa y aparece la pantalla inicial, lista para empezar a trabajar con él.



En esta pantalla inicial, como es habitual, tenemos la barra de menú y la barra de herramientas. Su uso no presenta mayor dificultad y ya debería estar en español. Para salir, como siempre **Archivo**→**Salir**

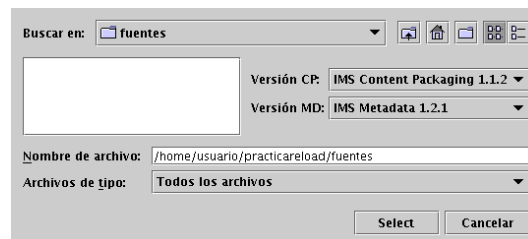
La mejor manera de ver su funcionamiento es, sin duda, guiar su uso de forma

Práctica

Para la realización de esta práctica, antes vamos a preparar la zona de trabajo. Así que crearemos¹⁷ un directorio en donde pondremos los ficheros que la van a constituir¹⁸, en él pondremos nuestro trabajo:

```
$ mkdir -p ~/practicareload/fuentes
```

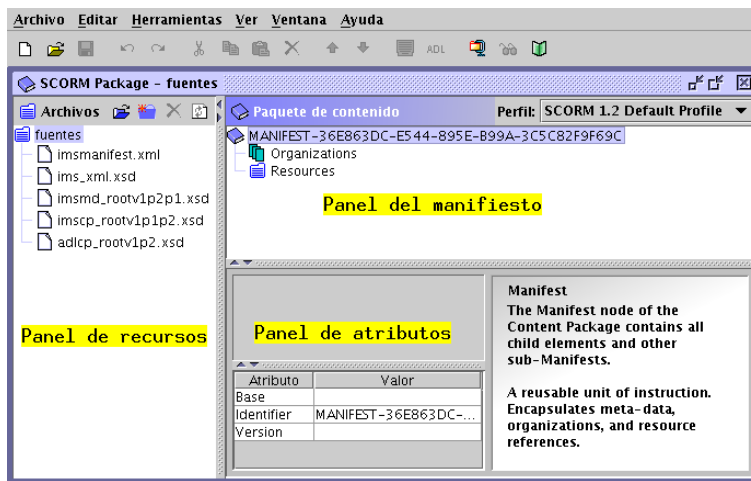
Abrimos el editor y pulsamos sobre **Archivo**→**Nuevo**→**Paquete SCORM 1.2** (📄). Se nos abre una ventana que nos permitirá seleccionar el directorio de trabajo en el que vamos a desarrollar nuestro proyecto.



Una vez que llegamos hasta la ruta deseada, pulsamos sobre el botón **Select**. Se crearán de forma automática una serie de archivos, de ellos el más significativo es el que contiene el manifiesto (`imsmanifest.xml`) del paquete SCORM que acabamos de crear, se trata del archivo de datos principales.

¹⁷También podemos optar por crear el directorio de trabajo desde el propio editor.

¹⁸En ese directorio podemos copiar y pegar todos los ficheros que vamos a incluir en nuestro proyecto.



Además de la barra de menús y la de herramientas (su uso es casi inmediato), podemos ver que la pantalla se divide en tres grandes zonas:


Panel de recursos además del fichero que contiene el manifiesto y los del esquema XML, en él se sitúan los archivos que vamos a usar en nuestro paquete.

Panel del manifiesto en esta zona se define el paquete: manifiesto, metadatos y recursos.

Panel de atributos nos da información de los distintos elementos que constituyen el paquete. Desde él podemos cambiar determinados atributos a esos elementos.

Ya tenemos la estructura básica de nuestro proyecto.

Vamos a añadirle el material que va a contener. El material lo tenemos disperso por el ordenador después de bajarlo/trabajarlo en el curso y mejor si lo situamos en el directorio creado. Para eso crearemos una carpeta¹⁹ (que contendrá las “fuentes”) en las que incluiremos a su vez una serie de subcarpetas que

contendrán los ficheros necesarios para su realización. Así que usando el botón  del panel de recursos crearé el directorio `material` y dentro de él los directorios:

```
HotPotatoes
Jclic
WINK
impress
pdftohtml
LyX
nvu
```

Dentro de cada uno de estos directorios vamos a situar los siguientes archivos:

`HotPotatoes`: los htmls de la práctica del tema 2

`Jclic`: Miproyecto en formato Jclic y el fichero `index.html`

`WINK`: práctica sobre wink (html y swf)

`impress`: la presentación del tema 5 (fichero en formato impress, html y gráfico)

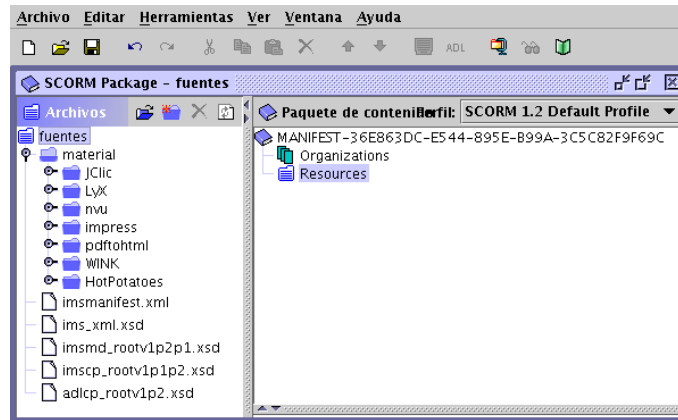
`pdftohtml`: el tema 4 del curso en formato pdf y el html que se obtiene al pasarle el comando `pdftohtml`.

`LyX`: la práctica sobre LyX en formato pdf y la de L^AT_EX-Beamer (apuntes y presentación)

`nvu`: la página que se ha hecho en este tema (html y gif animado)

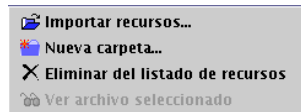
He puesto en Moodle todos los ficheros necesarios con la estructura de directorios anterior y ya comprimidos, el fichero se llama `practica-reload.tgz`. De todas formas creo que es mejor si lo hacéis vosotros mismos para ir viendo cómo funciona el programa.

¹⁹No es obligatorio hacerlo así ya que podemos ir incluyendo el material abriendo Nautilus y usando copiar y pegar.



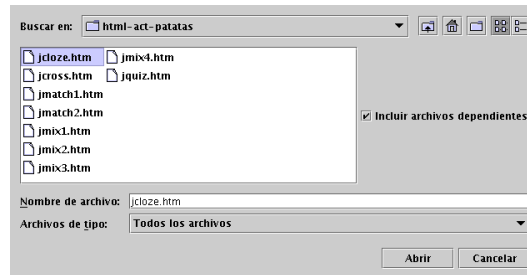
A tener en cuenta:

- Para que nos cree los directorios en el lugar adecuado antes hemos de situarnos en el directorio padre. Si no es así los crea en el raíz desde el que estamos trabajando.
- Si pulsamos sobre un directorio o recurso con el botón derecho del ratón, podemos



- Cuando una carpeta no está vacía se marcará con el gráfico

Añadamos las actividades de HotPotatoes. Para ello me sitúo en el directorio de destino, pulso sobre el icono del panel de recursos y busco la ruta en donde tengo situados los ficheros creados en la práctica sobre HotPotatoes



Que no se nos olvide marcar la casilla **Incluir archivos dependientes**, de esa forma él solo se encargará de añadir los ficheros necesarios para ese recurso (por ejemplo gráficos, otros ficheros hiperenlazados, sonidos, ...). Repetimos el proceso para cada uno de los materiales a incluir.

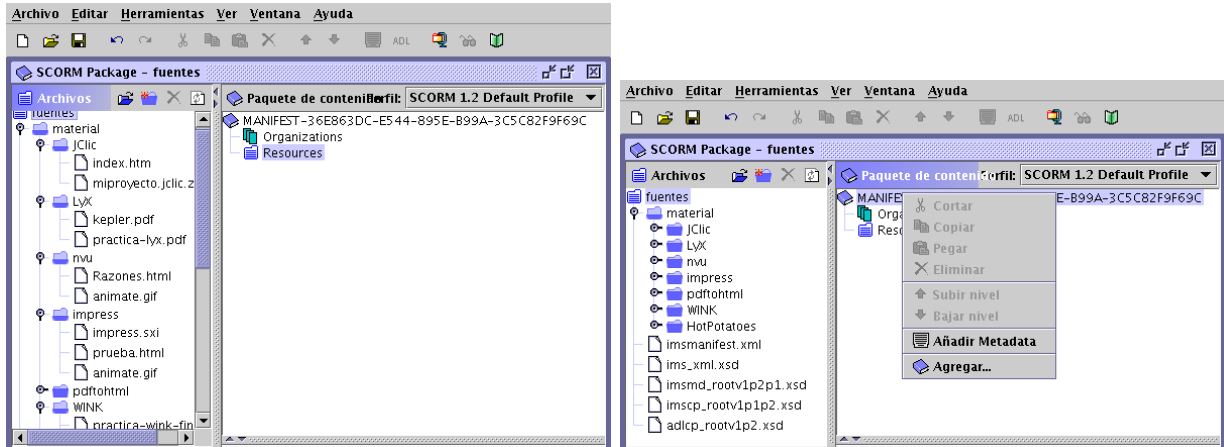
De nuevo recordar (y ya van tres veces, ¡qué pesado!) que podemos usar este método, copiar y pegar usando Nautilus, usar el comando cp, ... ¿Cuál es mejor? Prefiero usar el propio editor cuando no sé qué gráficos o enlaces necesito incluir, pero si son varios ficheros sin gráficos (como en este caso), prefiero copiarlos directamente o usar Nautilus.


Si tras iniciar un proyecto añadimos algún recurso copiándolo a la carpeta adecuada tendremos que reiniciar () el editor para que lo detecte.

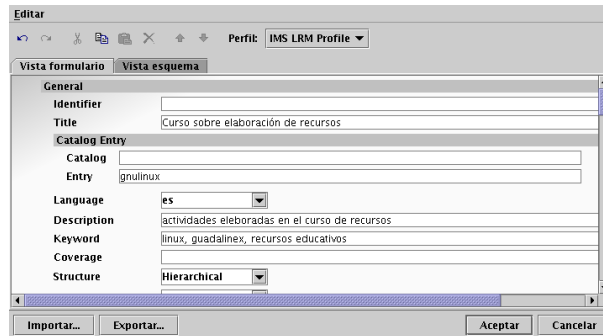


Tras todo este trabajo, deberíamos tener en cada directorio los ficheros que necesitamos para la realización de la práctica.

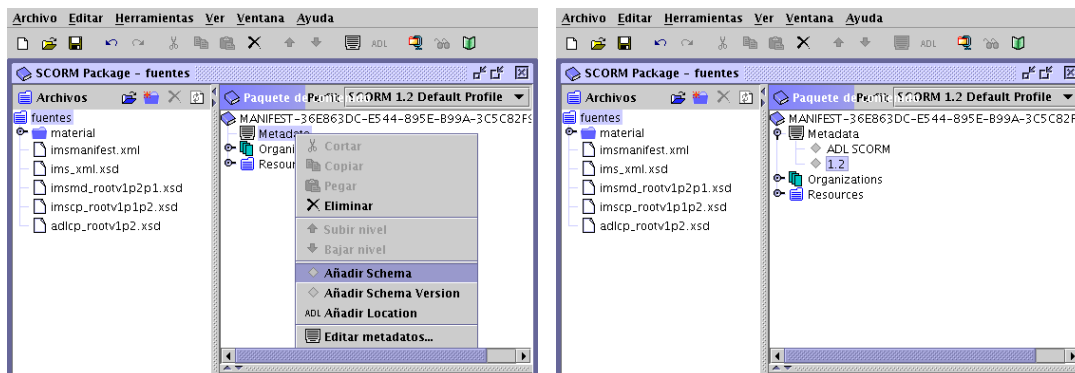
Con nuestro material ya preparado, pongámonos a desarrollar “nuestros apuntes”. Nos situamos dentro del panel del manifiesto sobre MANIFIESTO, pulsamos el botón derecho del ratón y seleccionamos **Añadir Metadata**.



Nos crea un icono con el nombre Metadata, nos situamos sobre él y volvemos a pulsar el botón derecho del ratón, optando por **Editar Metadatos** () y añadimos los metadatos asociados a nuestro proyecto (no es necesario rellenarlos ahora, podemos hacerlo después). Los metadatos contienen información acerca del contenido del paquete que facilitará la búsqueda y su uso por otros usuarios.



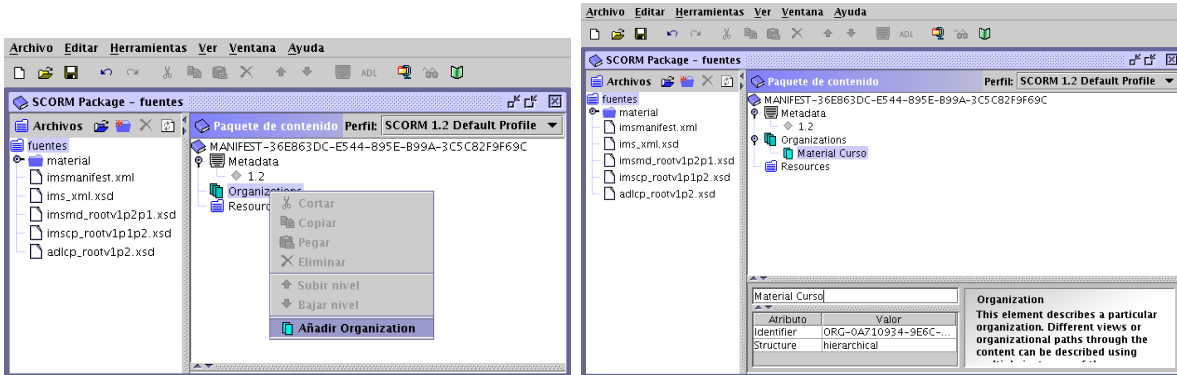
Aceptamos y, de nuevo en el panel del manifiesto, pulsamos sobre Metadata con el botón derecho del ratón y, añadimos²⁰ que se trata de un paquete ADSL SCORM y después la versión (**Añadir Schema y Añadir Schema Versión**).



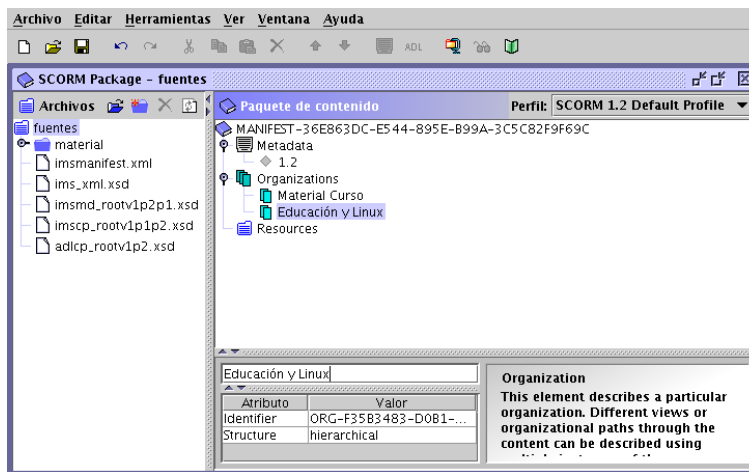
²⁰No es imprescindible.



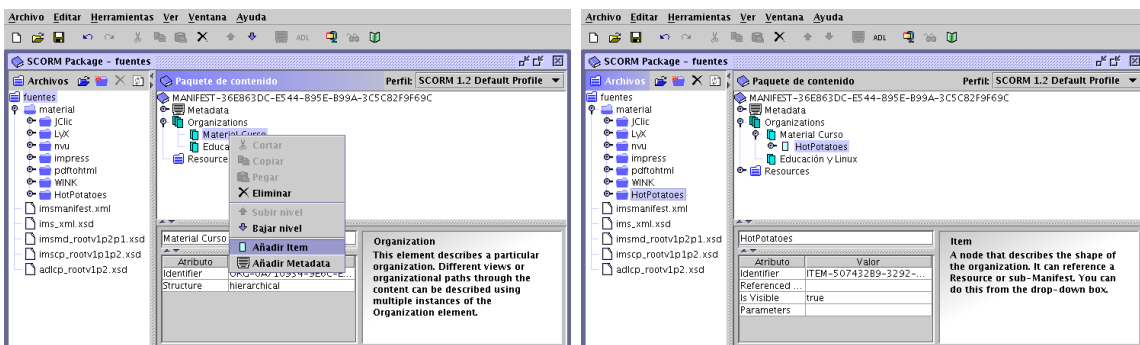
Por defecto el paquete de contenidos no tiene ninguna organización de los mismos, para “estructurar” las actividades es necesario crear una Organización. Nos situamos sobre **Organizations**, hacemos clic con el botón derecho del ratón y marcamos **Añadir Organization**. En la parte inferior de la ventana cambiamos el nombre por defecto por el que se adecue a nuestros intereses.



Para esta práctica he creado dos organizaciones, *Material Curso* y *Educación y Linux*

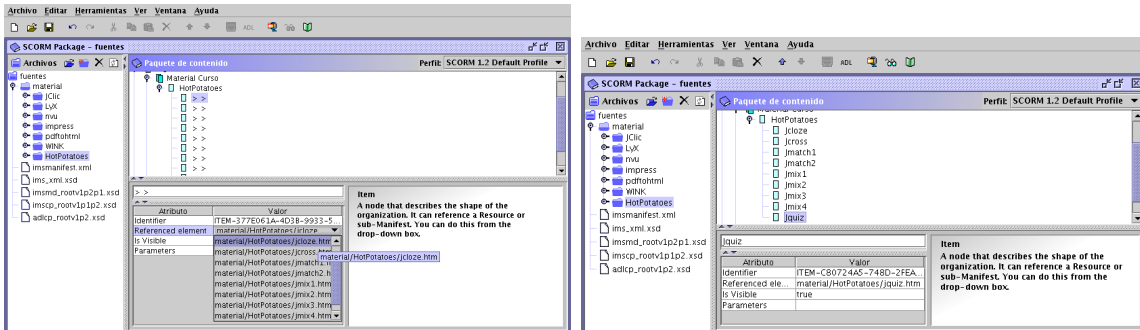



Ahora, dentro de cada organización vamos poniendo los distintos contenidos o recursos que corresponden a cada una (la secuencia con que se añadan será la que aparecerá después en el menú, aunque se puede cambiar después si es necesario). En primer lugar voy a añadir a *Material Curso* los archivos de HotPotatoes.

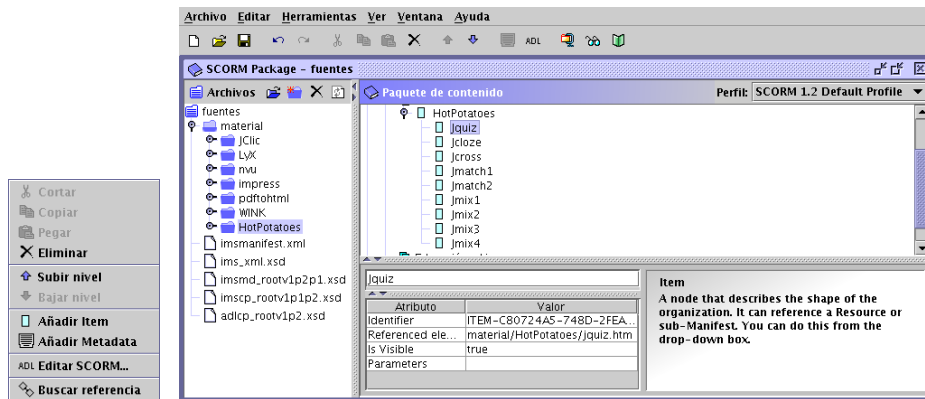




Se puede añadir un ítem desde el menú que se obtiene al situarnos sobre una organización y pulsar el botón derecho del ratón o simplemente arrastrando y soltando cada elemento desde el panel de recursos (si se trata de un directorio él sólo creará todos los ítems y añadirá los ficheros contenidos en ese directorio). En el caso de HotPotatoes, al arrastrar y soltar no toma los nombres correctamente, por lo que luego he tenido que escribir el nombre correcto desde el panel de atributos. Notar que en el campo de **Referenced Element** aparece el nombre del fichero²¹. Si nos equivocamos no hay problema, con él marcado y la tecla **Supr** (o con deshacer) lo eliminamos.



Si deseamos reorganizar los distintos ítems podemos moverlos, para eso, pulsamos sobre el botón derecho del ratón o sobre el botón  de la barra de herramientas. De esta forma he puesto la actividad de Jquiz la primera

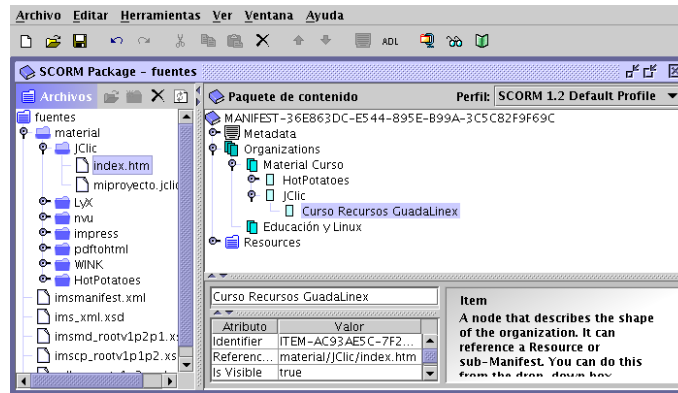



Añadiré ahora un nuevo ítem, pero éste de forma manual. Le pondré de nombre JClic y arrastrando situaré sobre él el fichero `index.html` que hay en la carpeta correspondiente a JClic, situada en el panel de recursos dentro de material. Al arrastra el fichero al lugar correspondiente el programa pondrá de forma automática el nombre (Curso Recursos Guadalinux).

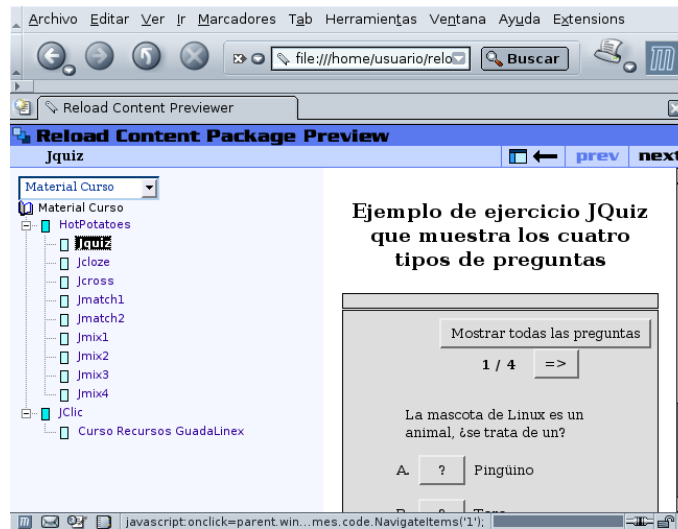
²¹ Si seleccionamos un elemento de la carpeta **Resources** situada en el panel del manifiesto, veremos un último campo (SCORM Type) del panel de atributos. Este último campo puede tomar dos valores:

Assets si toma este valor, se trata de un recurso cuyo objetivo no es que sea navegable desde el índice de contenidos. Por ejemplo un gráfico hiperenlazado desde un página Web.

SCO es de este tipo si se trata de recursos que representan un objeto de contenido que va ser accesible desde el índice de contenidos.

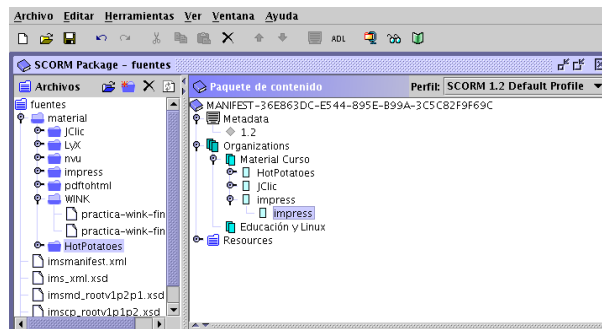


Veamos cómo va el trabajo pulsando sobre el icono  de vista previa

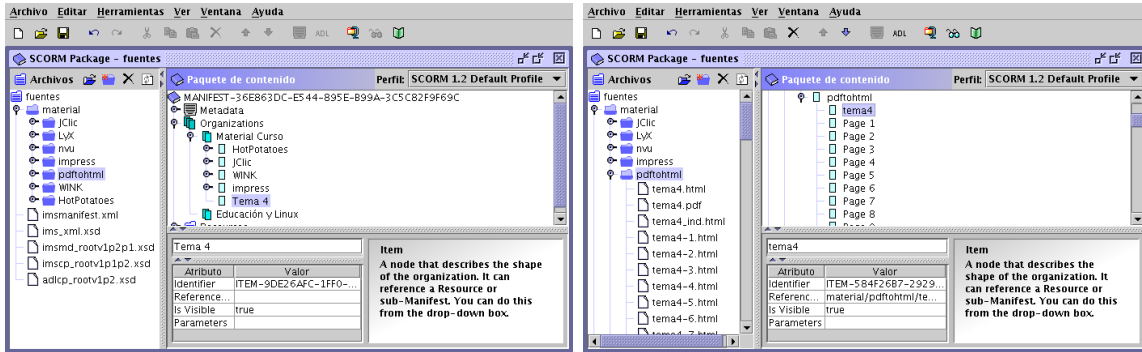



Continuemos con nuestro paquete. Le llega el momento a WINK, creo para él un nuevo ítem y arrastro hasta él el fichero practica-wink-fin.html.

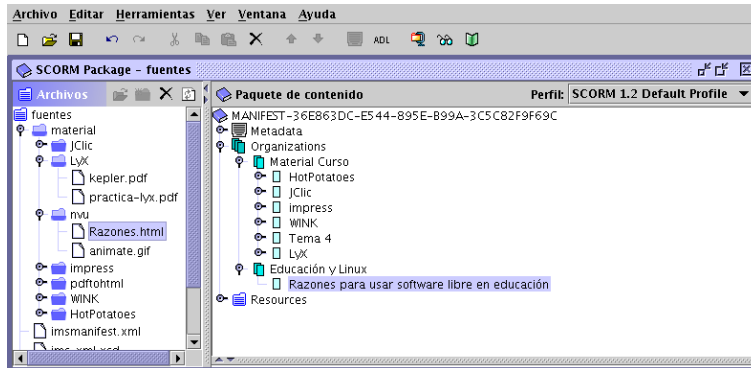
Sin embargo con impress arrastro el directorio entero. Como me copia todos los ficheros del directorio, elimino los que no me interesan que aparezcan en los menús: prueba.html y animage.gif






Añado otro ítem para el pdf tohtml y le cambio el nombre a Tema 4 desde el panel de atributos. Hay dos ficheros que no quiero que aparezcan, así que en este caso, los marco y pulso sobre la tecla suprimir. Se trata de tema.html y Outline y los gráficos. Es decir, dejo sólo los ficheros tema4.pdf y Page 1 a Page 35.

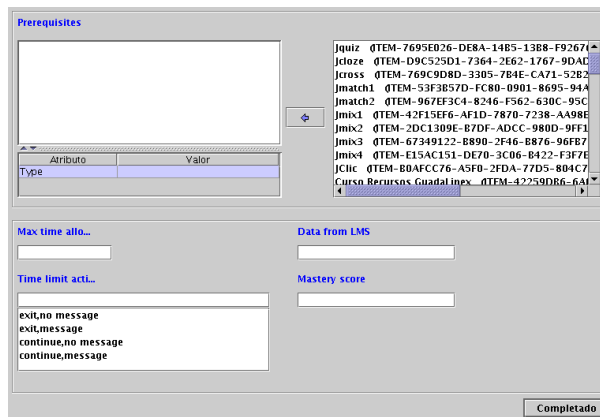


Por último arrastro la carpeta LyX, como contiene sólo dos pdfs, no tengo que retocar nada. Ya casi, ahora arrastro el fichero Razones .html hasta el organizador *Educación y Linux* y guardamos nuestro trabajo (.



De nuevo es buena idea ver cómo nos ha quedado pulsando sobre vista previa . Ahora es el momento de ajustar las entradas del menú de la izquierda a nuestro gusto, por ejemplo, cambiando o disminuyendo la longitud de los textos.


 El formato SCORM permite hacer más cosas que se escapan de lo que pretendo en este tema. Así, si con un organizador en vídeo inverso, marcamos sobre el icono  se nos abre una ventana



que nos permite establecer:



Prerequisites tareas o trabajos que hay que tener superados antes de poder seguir con la actividad.



Los seleccionamos en el recuadro de la derecha y pulsando sobre  los pasamos al de la izquierda.

Max time allo... tiempo máximo para esta actividad (en minutos)


Data from LMS datos que se envían desde la plataforma en que se ejecuta el paquete SCORM

Time limit acti... acción a realizar si se supera ese límite de tiempo: salir sin avisar, salir avisando, continuar sin avisar o continuar avisando.

Mastery score puntuación máxima (de 0 a 100)

Cuando a un ítem se le han añadido metadatos, el icono se remarca con un punto en el centro  .

Llega el momento de:

1. **Archivo** →  **Crear el paquete de contenidos...** Con esta opción crearemos el paquete SCORM. Se trata de un fichero zip que contiene las actividades (recursos u objetos de aprendizaje) así como información sobre la organización y estructura de los mismos (manifiesto). Pero para que podamos usarlo nuestro LMS tendrá que tener el soporte adecuado (Moodle lo tiene)



Este paquete es el que debemos usar si deseamos usar la potencia del formato SCORM²². Desde Moodle²³, hacer uso de él es un "juego de niños".

2. **Archivo** →  **Guardar el archivo como vista previa...** Si lo que deseamos es usarlo sólo como "página web" esta es nuestra opción.

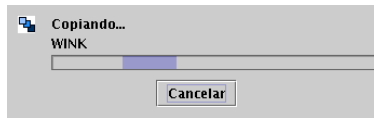
Me gusta crear un directorio en el que lo guardaré, por ejemplo, uno de nombre `recursos`



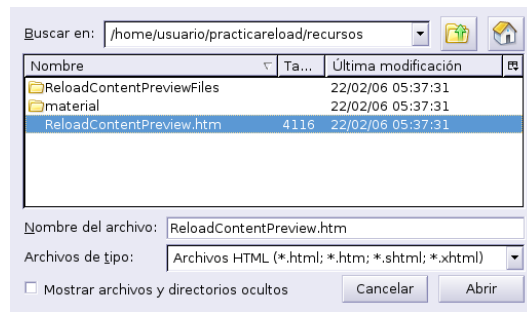
Cuando esté dentro de él pulso sobre el botón **Select**

²²Si deseamos comprobar cómo queda nuestro paquete, en <http://www.reload.ac.uk/scormplayer.html> tenemos a nuestra disposición el fichero `Setup_ReloadSCORMPlayer121_lnx.bin` (se instala de igual forma que el editor). Una vez instalado dispondremos en nuestro sistema de un visualizador de paquetes en formato SCORM.

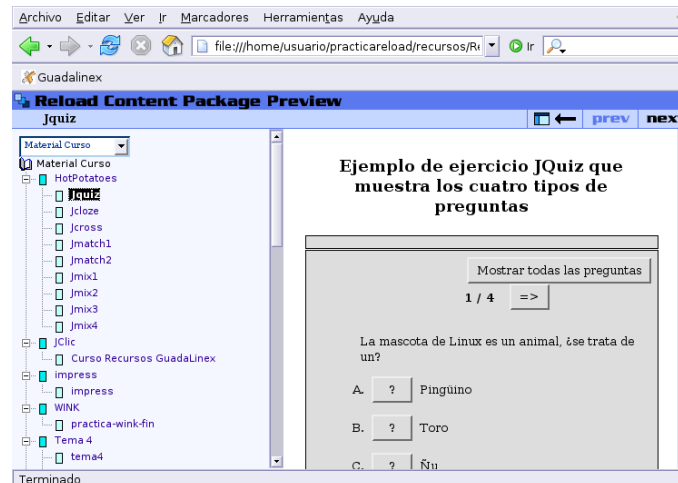
²³En la documentación de e-ducativa aparece que está soportado, pero no puedo testarlo.



y cuando finalice el proceso de copia, podremos ver su contenido desde cualquier navegador Web, sólo tenemos que optar por visualizar el fichero



Ya tenemos nuestra flamente Web creada.

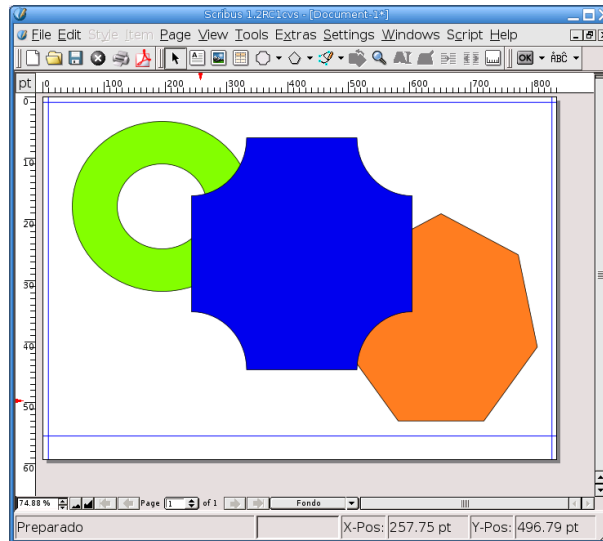


En la web del curso podéis comprobar el resultado final en ambos casos.

7.4. Ampliación: Scribus

Scribus es un paquete de Auto-Edición (DTP: *Desktop Publishing*) para sistemas Linux al estilo de programas comerciales como el Abode PageMaker, QuarkXPress, o Calamus SL. Es una de las alternativas para hacer Autoedición en linux ya que cuenta con multitud de funciones y posibilidades.

Es ideal para crear folletos, trípticos, carteles, revistas y en general cualquier documento de este estilo que se os ocurra. Una vez generado y guardado el documento cabe la posibilidad de modificarlo rápidamente con cualquier editor de texto, en lugar de utilizar formatos propietarios, binarios e indocumentados como otros programas, es muy sencillo obtener el documento PDF para su posterior impresión.



Lo encontramos en  **Aplicaciones** > **Menú Debian** > **Aplicaciones** > **Gráficos** > **Scribus**
O desde una xterm:

```
$ scribus &
```

Su web <http://www.scribus.net/>

No voy a explicar su uso, pero creo que merece la pena que lo tengáis como referencia si optáis por la autoedición. Para aquellas personas interesadas en este tema, he puesto tres artículos de la revista Linux-Magazine en Moodle donde se explica su uso y posibilidades.

Ω