



Del Arte de imprimir o la Biblia de 42 líneas: aportaciones de un estudio crítico

Luz María Rangel Alanís

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tdx.cat) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tdx.cat) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tdx.cat) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

**Del arte de imprimir
o la Biblia de 42 líneas:
aportaciones de un estudio crítico**

Tesis doctoral presentada por:

Luz María Rangel Alanís

Dirigida por:

Dr. Enric Tormo Ballester
Universidad de Barcelona

Dr. Aureli Alabert Romero
Universidad Autónoma de Barcelona

Programa de doctorado:

Las revoluciones tipográficas
Bienio 2000-2002

Universidad de Barcelona
Facultad de Bellas Artes
Departamento de Diseño e Imagen

Abril, 2011.

Capítulo

6

Dum venio, attende lectioni,
exhortationi, et doctrinae.
Noli negligere gratiam, quae in te est,
quae data est tibi per prophetiam, cum impositione
mannum presbiterii.

1 Ad Timotheum 4:13-14

...

Entre tanto que voy, ocúpate de la lectura,
la exhortación y la enseñanza.
No descuides el don que hay en ti,
que te fue dado mediante profecía con la imposición
de las manos del presbiterio.

1 Timoteo 4:13-14

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.1 Introducción

El propósito principal es obtener un catálogo de tipos de la Biblia de 42 líneas. La selección de letras se efectuará a partir de dos parámetros: tipográfico y de diseño. Este estudio nos ayudará a verificar si las afirmaciones hasta ahora *históricamente ciertas* de si la Biblia esta conformada por un total de 299 tipos¹ son verdaderas.

Trabajaremos con el microfilm del tomo del Nuevo Testamento de la Biblia de 42 líneas² que tiene en su acervo la Universidad de Sevilla, pues al hacer los acercamientos necesarios con el aparato de medición no se observaban deformaciones en el contorno. A la vista de la letras impresas se puede observar que en las minúsculas encontramos una o más variantes de la misma letra. Tomaremos en cuenta además la existencia de mayúsculas y otros casos como abreviaturas, ligaduras y signos de puntuación. Esta información deberá quedar reflejada en las listas del catálogo.

Para obtener dicho catálogo se desarrolla proceso de digitalización, vectorización y simulación compositiva de caracteres cuya metodología se explicará brevemente a continuación. Los apartados siguientes se implementan de la siguiente forma: el Apartado 6.2 detalla el método de trabajo con la máquina de medición sin contacto QVA-200 de Mitutoyo³, además de los programas QVPak y FormPak que generan los ficheros de uso .DAT. En el Apartado 6.3 se describe cómo se realiza la programación del lector y analizador modular de los datos para la reconstrucción vectorial de los tipos de la Biblia cuyo resultado es el código fuente ventaQVdat_011.ps. Este código a su vez genera los archivos .DAT.B42 que se usan a modo de pseudoformato tipográfico en el simulador para la composición de líneas de ginyB42.ps que produce un archivo PDF; el detalle de la programación del código fuente se realiza en el Apartado 6.4. El Apartado 6.5 muestra los resultados de una tercera programación, la interfaz Web de usuario. La aplicación de todo este proceso nos lleva a poner en marcha las aplicaciones en un sistema de servidor WWW; el Apartado 6.6 nos puntualiza los requisitos mínimo y su operativa de uso. Finalmente el Apartado 6.7 esta dedicado a describir la implementación del proyecto de código libre <http://gutenberginy.sourceforge.net> que engloba todos estos desarrollos.

6.1.1 Metodología de trabajo

a) Laboratorio

El aparato de medición QVA-200 de Mitutoyo (Fig. 115) se basa en la detección de contornos y puede trabajar con una resolución de captación de 0.0001 mm. El contorno está definido por el cambio de intensidad de la luz que tiene dos parámetros de orientación en el escáner, *ascendente* de oscuro a claro y *descendente* de claro a oscuro. Se puede escoger entre varios sistemas de coordenadas:

- a) Sistema de coordenadas de máquina (SCM) es un sistema *fijo*, que puede trabajar con milímetros o pulgadas.
- b) Sistema de coordenadas de pieza (SCP) es un sistema *flexible*.
- c) Sistema de coordenadas de vídeo (SCV) se basa en *píxels*.

Para obtener nuestros contornos trabajaremos con el SCM y para la captura de imágenes con el SCV.

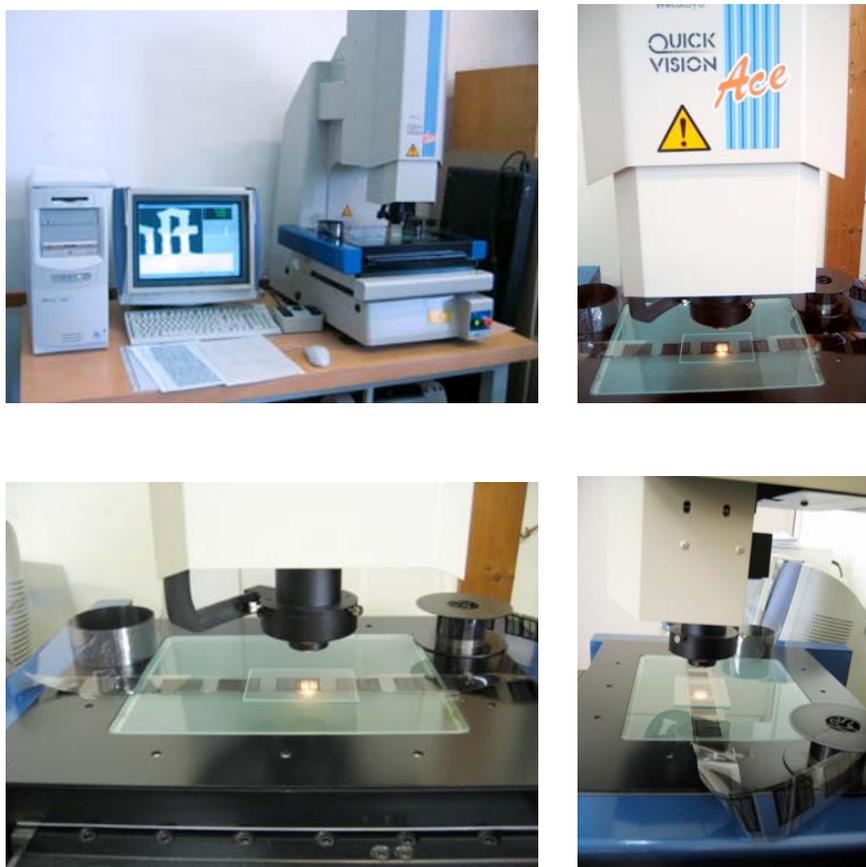


Fig. 115 Imágenes del aparato de medición Quick Vision durante el escáner del microfilm de la B42.

El escaneo de cada letra determina su *contorno*. Un contorno está formado por puntos y según se trabaje podemos unir los puntos para formar una curva cerrada compuesta de líneas rectas. Algunas letras poseen blancos interiores, por lo que el contorno puede estar formado, en términos topológicos, por varias componentes conexas.

Usando la aplicación *QVPak* obtendremos el escaneado de los contornos, quedando cada lectura guardada en un archivo *.TXT*. En la aplicación *Formpak* se acoplan las lecturas de los contornos y podemos guardarlos como ficheros de diversos formatos: *.DAT*, *.FPK*, *.MES*, *.DXF*, *.DE2*, etc. El que nos interesa es el fichero *.DAT*, que guarda los datos agrupando separadamente cada contorno y donde cada punto es referido como una coordenada.

b) Programación de la lectura y análisis modular de los datos para la reconstrucción vectorial de los tipos de la B42

Para el correcto control del primer paso de captura de datos, resulta imprescindible construir una aplicación capaz de manejar las miles de coordenadas (x, y) de lectura del microfilm, de forma automatizada y con todo rigor, para luego transformarlas en vectores. Todo ello con los mismos datos y recursos utilizados para el estudio de contornos (*Cluster Analysis*) expuesto en el capítulo 7.

Estos vectores, a su vez, necesitan ser fijados en un entorno de código y datos que les permita comportarse como un tipo de letra digital, de forma lo suficientemente flexible y precisa para ser tratada más adelante por un sistema de composición.

Es para esta labor que se escribió la aplicación *ventaQVdat_011.ps* que permite transformar un simple fichero de datos numéricos *.DAT* (proveniente del Quick Vision) en un archivo tipográfico (un tipo de letra) preparado para componer, dando como resultado dos ficheros: un *.B42* como formato del tipo de letra digital, y un *.PDF*, con el análisis gráfico y modular del carácter generado.

La operativa de uso también ha resultado importante: ejecución multiplataforma (Linux, Unix, MacOSX, Windows), en local o remota en un servidor, bajo línea de comando o enlazable en un entorno Web, sin instalación para el usuario y manejable con un simple navegador de Internet.

c) Programación del simulador para la composición de líneas con los tipos de la B42

La composición de líneas de la B42 habría podido ser tratada con muchas de las aplicaciones comerciales de diseño y diagramación que hay en el mercado, pero ninguna de ellas nos habría asegurado un entorno lo suficientemente estable y transversal de trabajo, ajustado a las necesidades específicas con las que esta investigación pretendía trabajar, con el máximo rigor, y sin concesión alguna a distracciones gráficas o de interfaz.

Necesitábamos asegurar un trabajo de composición con una herramienta capaz de manejar los archivos *.B42* dentro de un marco muy concreto de posibilidades en el tratamiento de los caracteres de la escritura y los espacios, siempre según las reglas y condicionantes que siguieron Gutenberg y Schöffer.

Fue para esta labor que se escribió la aplicación *ginyB42.ps*, que permite componer una sola línea de la Biblia (y solo una), bajo unas determinadas condiciones y comportamientos (y sólo estos), dando como resultado final un fichero en formato *.PDF* con la frase compuesta en una escala de trabajo que permite analizar con mucha precisión los pormenores del ensayo.

Como en la herramienta anterior, mantenemos las prestaciones de la operativa de uso: ejecución multiplataforma, en local o remota en un servidor, bajo línea de comando o enlazable en un entorno Web.

d) Programación del interfaz Web de usuario

Para poder dar la máxima usabilidad al componedor, ha sido imprescindible hacerlo por medio de una interfaz basada en los estándares Web y a través de la ventana de un navegador. Para ello el usuario podrá olvidarse por completo de la plataforma donde trabaje (Windows, MacOSX, Linux, Unix, iOS o Android), así como de cualquier protocolo de instalación e incluso de la potencia de su máquina, pues, como un simple Terminal, esta quedará relegada al control del servidor. El dispositivo (ordenador, tableta o teléfono), sólo necesitará acceder a una conexión a Internet que pueda soportar un navegador mínimamente actualizado.

e) Instalación y puesta en marcha de las aplicaciones en un sistema de servidor WWW

Exento de excepcionalidad, este entorno de trabajo (también llamado *cloud computing*⁴) es reproducible en una amplia gama de servidores de red e Internet. Para simplificar la vida al usuario, se han centrado los esfuerzos en las prestaciones de la máquina remota (el Servidor), que recibirá la petición de trabajo, procesará la información ejecutando el software escrito para esta investigación y luego la devolverá en forma de resultado en una ventana del navegador de este mismo usuario.

Por consiguiente, la instalación y configuración se ha concentrado en una sola máquina y una sola vez – la que actúa de Servidor en este proyecto en la Universidad de Barcelona –, y hemos documentado el proceso para que pueda ser reproducida también en otras máquinas, con mucha independencia del sistema operativo usado, tanto a escala global (Internet) como a escala local (red).

f) Proyecto de código libre

Para divulgar y dar continuidad a las aplicaciones desarrolladas en esta investigación, se ha optado por usar el portal de publicación de proyectos de código libre *SourceForge.net*. Este 'site' de referencia nos asegura los recursos que a medio plazo se necesitan para estabilizar los contenidos desarrollados para las herramientas de software y toda su documentación relacionada, ofreciendo visibilidad en la red para compartirlo.

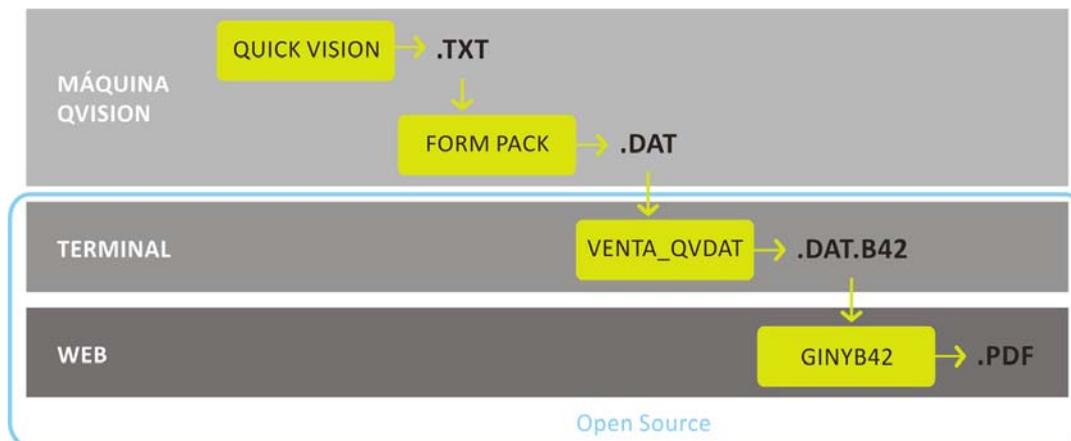


Fig. 116 Ámbitos de desarrollo de las aplicaciones.

Notas

1. Zedler, Gottfried. *Die Sogenannte Gutenbergbibel, XX*. Gutenberg-Gesellschaft, Mainz, 1929.
2. Gutenberg, Johannes. *Biblia Latina*. Moguntiae, (c. 1454- agosto, 1456). Alemania. Depositada en el Fondo Antiguo de la Biblioteca de la Universidad de Sevilla.
3. *Machines de mesure par analyse d'image 3D*. Mitutoyo [en línea]. France. © 2007. Disponible en: <http://www.mitutoyo.fr/index.php>
4. CLOUD COMPUTING. Es un término que se define como una tecnología que ofrece servicios a través de la plataforma de Internet. Los usuarios de este servicio tienen acceso de forma gratuita o de pago todo depende del servicio que se necesite usar. El término es una tendencia que responde a múltiples características integradas. Uno de los ejemplos de esta "nube" es el servicio que presta Google Apps que incorpora desde un navegador hasta el almacenamiento de datos en sus servidores. Los programas deben estar en los servidores en línea y puedas acceder a los servicios y la información a través de Internet. Una de las principales diferencias del Cloud Computing es que no hay necesidad de conocer la infraestructura detrás de esta, pasa a ser "una nube" donde las aplicaciones y servicios pueden fácilmente crecer (escalar), funcionar rápido y casi nunca fallan, sin conocer los detalles del funcionamiento de esta 'nube'. Maestros del Web. "Cloud Computing: nueva era de desarrollo" [en línea]. © 2008. Disponible en: <http://www.maestrosdelweb.com/editorial/cloud-computing-nueva-era-de-desarrollo/>

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.2 Laboratorio

6.2.1 Método de uso de la máquina de medición sin contacto: Quick Vision de Mitutoyo

Dado que la detección de un contorno está definido por la variación de intensidad de la luz, el primer requisito que aplicamos para elegir la letra a trabajar es que el contorno se lea de principio a fin en una sola lectura; así mismo cada contorno que integre la letra deberá leerse de esta forma.

Vista la imagen de la página en el microfilm fijamos unos parámetros para homogeneizar la lectura de los contornos: iluminación; longitud, intervalo y sentido del barrido; tipo de contorno y punto de inicio de la lectura en la letra.

Al escanear cada letra debíamos asegurar su posición, es por ello que el nombre puede contener 3 o 4 números y una letra, según el momento en que se añadió nuestro catálogo. El nombre de cada fichero refleja la ubicación física de la letra dentro de la página, según la pauta *página-columna-renglón-númerodecarácter-letra*. Por ejemplo si el nombre del fichero es *1-1-4-6-a.DAT*, esto significa que el carácter en cuestión se encuentra en la página 1, en la columna 1, en la línea 4, y es el sexto carácter, o sea una *a* (véase Tabla 36). En un inicio todas las letras pertenecían a la misma página de trabajo por lo que no se contemplo incluir el número de página. El nombre del fichero sólo tiene 3 dígitos y pertenece a la primera página del Libro de Mateo. En la Biblia de la Universidad de Sevilla la encontraremos como la página 8. Sin embargo, como ya se explicó en el Apartado 4.1.4, al verificar que existían más ejemplos de letras (véase Fig. 95) tuvimos la necesidad obtener más datos mediante una muestra (véase Tabla 28). No obstante si por casualidad encontrábamos alguna letra fuera de la muestra que completara nuestro catálogo también era tomada en cuenta, ya que el nombre del fichero nos daría su ubicación exacta.

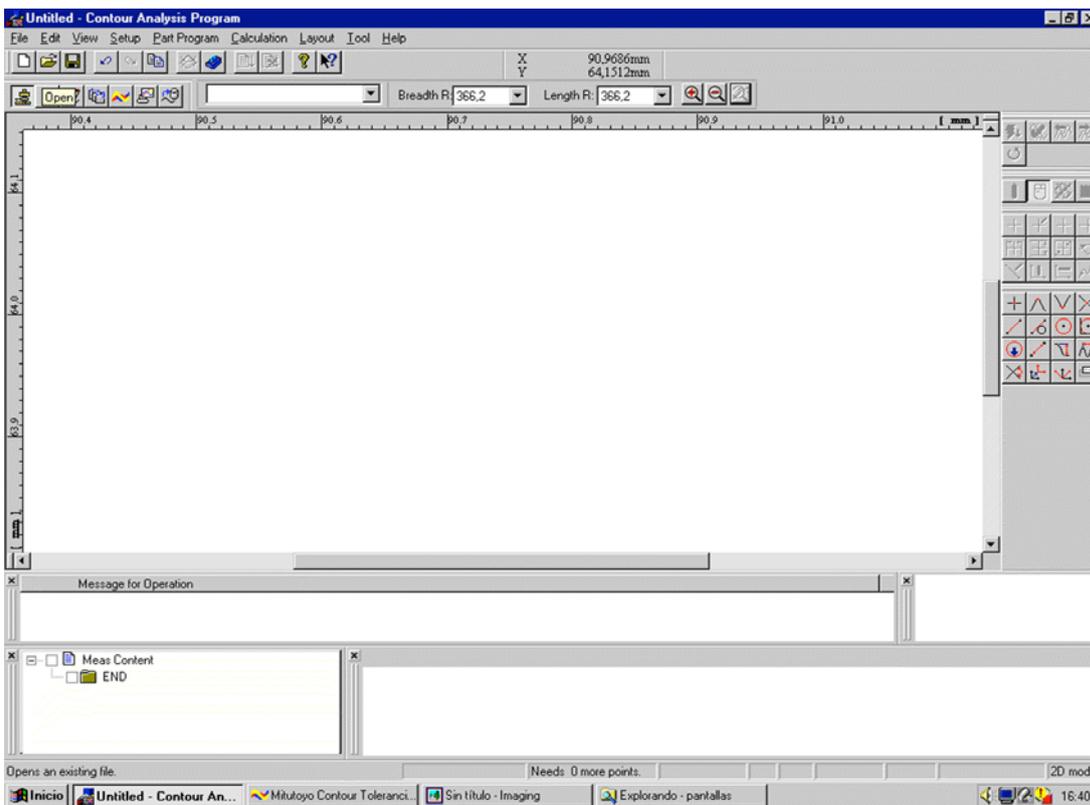
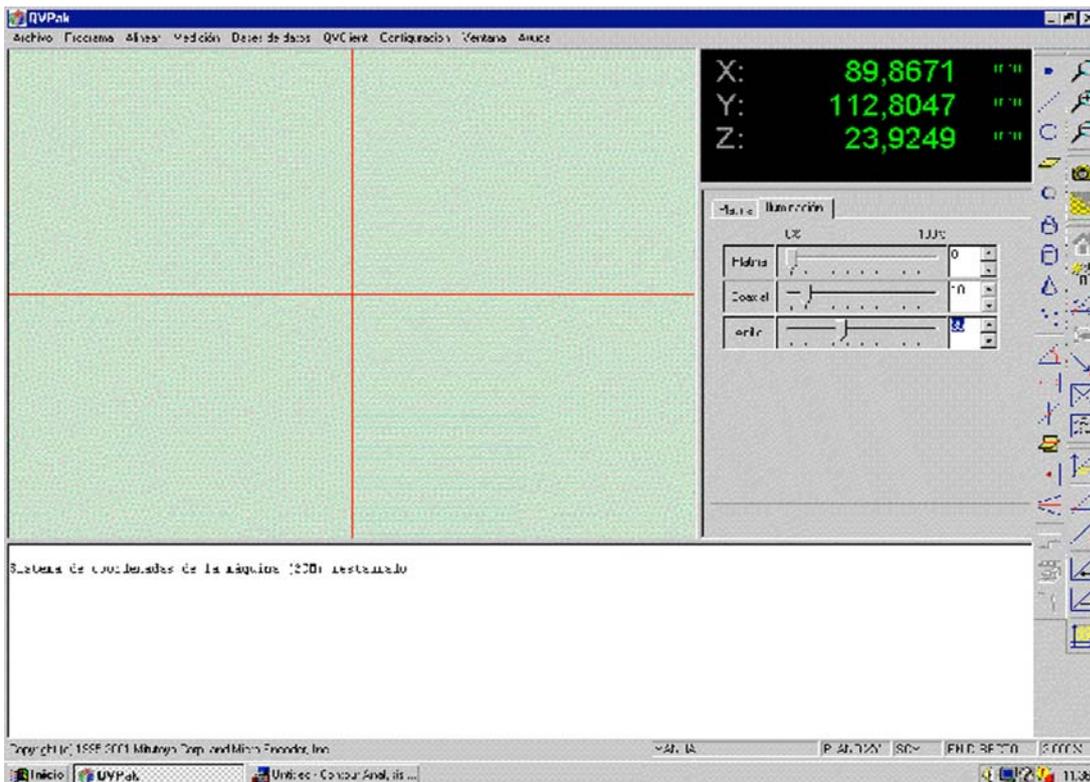


Fig. 117 Pantalla principal de las aplicaciones QVPak y Formpak

6.2.2 Generación de ficheros .DAT con 1, 2 y 3 contornos

Para la obtención de los contornos debimos trabajar con el microfilm de la B42 bajo el siguiente protocolo:

En primer lugar se elige a priori cada carácter y la aplicación *QVPak* se hace un escaneado de los contornos que lo integren. Cada lectura quedará guardada en un archivo .TXT y con la nomenclatura del nombre que ya se ha explicado.

En la aplicación *Formpak* acoplamos el o los archivos .TXT (uno por cada contorno o *path*), dependiendo del carácter que se trate; por ejemplo una “i” tiene un contorno (Fig. 118), una “p” dos y una “a” tres contornos (en la Tabla 37 se puede verse el contenido del .TXT). Cada grupo de contornos se guarda con el formato propio de la aplicación .FPK, una vez integrada la letra podemos hacer uso de ella dentro de la aplicación misma, en otra aplicación o guardarla, según el trabajo a desarrollar.

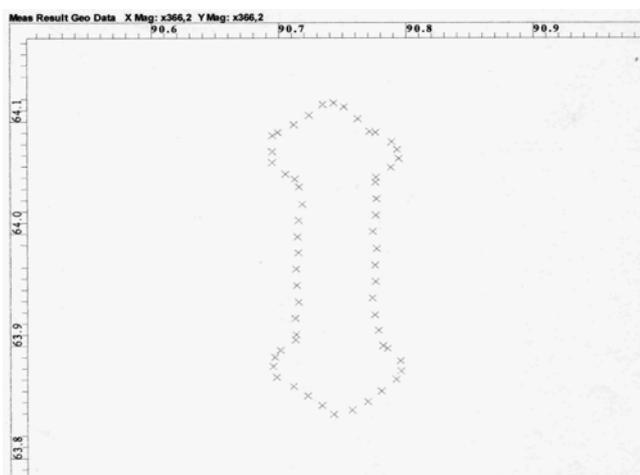


Fig. 118 Aspecto en pantalla de la “nube de puntos” que conforman un contorno, fichero .TXT en Formpak.

En nuestro caso nos interesa guardar los ficheros como .DAT, que agrupa los datos por contorno y donde cada punto es referido como una coordenada (véase Tabla 38). También se pueden obtener ficheros .DXF para usar en programas vectoriales comerciales como Autocad, Corel Draw o Illustrator.

Los datos de los archivos .DAT serán modificados para ampliar la letra proporcionalmente a cualquier tamaño, por lo que se establece una medida de ampliación – que nombraremos de ahora en adelante *factor maestro (fMd)* o *factor multiplicador de escala* –, y que debe cumplir el requisito de ser cómoda para las cuestiones tipográficas además de contemplar la salida de impresión. Para ello establecemos una cuadrícula de módulos basada en el concepto del cuadratín. L

Nuestra unidad de medida será nombrada como *unidad matriz*, que tendrá como base el punto tipográfico (cícero Didot y pica anglosajona), además tomaremos como referencia el cuerpo 18 (cícero Didot) de la B42 por lo cual la equivalencia principal es está:

$$270 \text{ mm} = 720 \text{ puntos de cícero Didot} = 40 \text{ cuadratines del cuerpo 18 Didot}$$

La tabla de ampliación y equivalencias se muestra a continuación:

		Milímetros	Puntos pica ⁵	Puntos de cícero Didot	Cuadratines cuerpo 18 (Didot)
Unidad matriz	1/5 SubMódulo	6	17.01	16	0.88
	1/2 SubMódulo	15	42.52	40	2.22
	1 SubMódulo	30	85.05	80	4.44
	2 SubMódulo	60	170.10	160	8.88
	3 SubMódulo	90	255.15	240	13.33
	4 SubMódulo	120	340.20	320	17.77
Altura x = módulo m	5 SubMódulo	150	425.25	400	22.20
	6 SubMódulo	180	510.30	480	26.66
	7 SubMódulo	210	595.35	560	31.11
Módulo tipográfico	8 SubMódulo	240	680.40	640	35.55
Módulo caligráfico	9 SubMódulo	270	765.45	720	40

Estas equivalencias también pueden servir:

$$\begin{aligned}
 1 \text{ mm} &= 2.835 \text{ (factor para convertir milímetros a puntos de pica)} \\
 1 \text{ unidad matriz} &= 6 \text{ mm} \times 2.835 = 17.01 \text{ puntos de pica} \\
 72 \text{ puntos de pica} &= 1 \text{ pulgada}
 \end{aligned}$$

La visualización de la modulación a partir de la *unidad matriz* se muestra en la Figura 119.

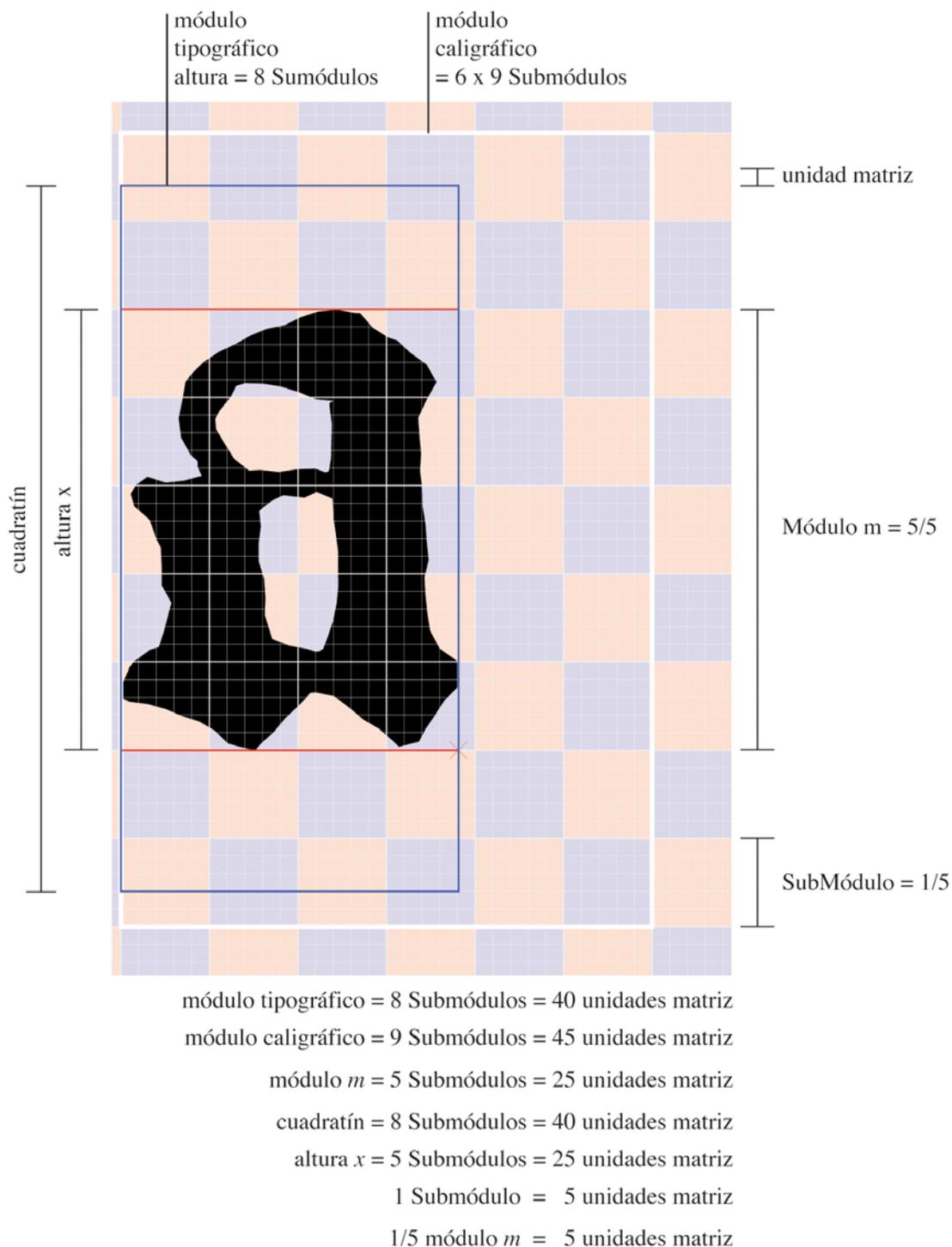


Fig. 119 Unidades de medida para el desarrollo de la programación.

Notas

5. El Sistema Internacional de Unidades (SI) admite, en su Resolución 10 de la 22^a Conferencia General de Pesos y Medidas de 2003, el uso del “punto” y la “coma” como separación decimal. En España y como en el resto de Europa, excepto las Islas Británicas, esta establecido el uso de la coma. En esta parte de la investigación haremos uso del “punto decimal” proveniente de la tradición anglosajona por la sencilla razón de que es lo más utilizado en los lenguajes de programación.

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.3 Programación del lector y analizador modular de los datos para la reconstrucción vectorial de los tipos de la Biblia de 42 Líneas

6.3.1 Objetivos de la aplicación

Todo lo referente a este capítulo es relativo a *ventaQVdat_011.ps*

1. Leer con precisión los .DAT generados por Quick Vision

Las coordenadas x , y leídas por la máquina de medición sin contacto (*Quick Vision*), son números de punto flotante de hasta seis decimales. Para evitar desviaciones, debidas a los redondeos frecuentes que se producen en los intérpretes, estos datos se convierten previamente a enteros (de hasta 7 dígitos) antes de ser tratados por nuestro algoritmo.

Véase en la Tabla 38 el aspecto interno y tipo de datos de un fichero .DAT generado por *Quick Vision* del carácter de la Biblia *minuscualigada_sslarga*.

Véase en la Tabla 39 el aspecto interno y tipo de datos del diccionario TraMS (objeto de programación), con las coordenadas clasificadas por contornos ($/o/1$) y ejes ($/xCOO/yCOO$).

2. Adaptar la coordenadas leídas a un sistema tipográfico modular representable gráficamente

Aproximarnos a la solución gráfica del proyecto de diseño que permitió la generación de cada uno de los tipos móviles de la Biblia, es lo que hay detrás de la representación gráfica resultante de la ejecución de esta herramienta, que podemos ver en la Tabla 40, donde podrán distinguirse claramente cada uno de los elementos fundacionales de un sistema tipográfico (módulo, línea de base, caja del carácter, origen del carácter siguiente, etc). Los tamaños del fichero .PDF resultante, para facilitar su impresión en papel, se han adaptado al DIN A4.

3. Escribir con precisión un tipo de letra digital componible

En paralelo a la representación gráfica anterior, esta aplicación escribe en un fichero (.B42) todos datos del carácter que son necesarios para hacer funcionar un componedor dinámico de

textos. Podemos ver los datos del carácter *minuscualigada_sslarga* en este pseudoformato⁶, ordenados para mejorar su lectura en la Tabla 41.

6.3.2 Análisis de cada uno de los parámetros que condicionan el resultado

· Fichero .DAT del Quick Vision que se analiza

Parámetros [A] y [C] del menú de cabecera del código fuente.

Tipo de dato: *cadena de texto* con el *path* del fichero.

Los ficheros .DAT son ante todo ficheros de texto puro, *ascii*, que se leen para clasificar los contornos y coordenadas del carácter para su posterior análisis. Existe una validación previa de este particular formato que genera *Quick Vision* (.DAT es solo una extensión genérica), en base a un marcador de inicio de datos por contorno (*mm*) y la organización de los puntos de coordenadas, uno por línea, separados por una coma.

· Deducción del factor *fMd*

Parámetros [1] [2] [A] y [B] del menú de cabecera del código fuente.

Tipo de datos: *Booleano*, *cadena de texto* con el *path* del fichero y *datos numéricos*.

El factor multiplicador de escala (*fMd*) es un parámetro fundamental de este algoritmo, que sirve para encajar perfectamente el tamaño del carácter dentro del sistema modular tipográfico que se ha establecido. Si es necesario, debido a desviaciones en la medición u otros desajustes, éste parámetro deberá calibrarse para cada carácter.

Podemos decirle al programa que lo deduzca o (si ya lo hemos establecido) que lo lea directamente como parámetro. El factor multiplicador de escala (*fMd*) deberá empujarse las coordenadas del carácter a una escala que le permita trabajar con la llamada *unidad matriz* (la que construye todo el sistema modular de nuestra aplicación), que tiene un valor de *17.01* puntos de pica. El sistema modular se ha calculado para que los glifos puedan trabajarse cómodamente dentro de un formato DIN A4.

En el caso de los caracteres de la Biblia, hemos empezado deduciendo el *fMd* de los glifos ... a e m i n o r s t u v ... pues sabíamos de antemano que todos ellos median exactamente una altura de 5 SubMódulos (altura *x*), o lo que es lo mismo: 25 *unidades matriz* (donde 1 *unidad matriz* = 1/5 parte de un SubMódulo). Esta primera medición nos ha permitido establecer la altura del *Módulo Tipográfico*⁷ con un valor fijo de 40 unidades matriz (8 SubMódulos) y el *Módulo Caligráfico*⁸ de referencia, que lo inscribe, con un valor fijo de 6x9 SubMódulos (30 x 45 *unidades matriz*). El resto de caracteres se han ido ajustando en base a este sistema, hasta completar la tabla definitiva de valores para todo el catálogo (véase Tabla 42).

· Representación del carácter con relleno o perfilado de color

Parámetro [D] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Podemos plasmar el glifo de dos maneras: relleno al 100% de color (y sin perfil), o con solo un perfil fileteado de 0.7 puntos de grosor (y vacío por dentro). El color siempre será un negro intenso al 100% de color CMYK.

· *Posicionado de la línea de base*

Parámetro [E] del menú de cabecera del código fuente.

Tipo de dato: *número entero*.

Siempre medido en un número entero de *unidades matriz* verticales (hacia arriba) a partir de la base del *Módulo Tipográfico* (representado como un rectángulo azul). El carácter reposará en esta línea roja al componerse, y el glifo se construirá también a partir de aquí.

· *Representación del arlequinado modular*

Parámetro [F] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Para mejorar visualmente determinados aspectos de la retícula modular, podemos optar por activar (o no) el coloreado alternativo (salmón y gris) de los SubMódulos. Con ello mejoramos la visualización de los filetes finos (en gris) que marcan las *unidades matriz*, los filetes medios (en gris) que marcan los SubMódulos, y finalmente el filete más grueso (en blanco) que marca el *Módulo Caligráfico*.

· *Representación de la retícula modular*

Parámetro [G] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Permite visualizar las *unidades matriz*, los SubMódulos y el *Módulo Caligráfico*. Cuando su representación no va acompañada por el arlequinado modular, el rectángulo fileteado que marca el *Módulo Caligráfico* se pinta en gris.

· *Análisis de cobertura del carácter*

Parámetro [H] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Siempre en relación al *Módulo Tipográfico*, permite valorar el porcentaje de mancha que representa el glifo dentro de este. Gráficamente, genera una página adicional en el .PDF de salida, donde el formato de página es idéntico al *Módulo Tipográfico* (siempre representado como un rectángulo azul). La operación de barrido, punto a punto, se realiza a una resolución de 72 ppp.

· *Escritura del fichero .B42*

Parámetro [I] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Los mismos datos que se utilizan para representar gráficamente el carácter en .PDF, son escritos en un fichero *ascii*, bajo sintaxis *PostScript* y en un pseudoformato que podrá ser leído más tarde por el componedor.

· *Nombre interno del carácter*

Parámetro [J] del menú de cabecera del código fuente.

Tipo de dato: *cadena de texto*.

Obedece al nombre que describe el carácter de forma sucinta. Siempre sin espacios en blanco y haciendo uso de mayúsculas y minúsculas si es necesario. También pueden utilizarse números y signos.

Este valor textual permitiría que el componedor pudiera llamar el carácter por su nombre, en vez de por su código, como ahora. Por el momento sólo se usa a título informativo como, por ejemplo, en el etiquetado flotante que se activa en el momento de la composición de un renglón o en las tablas de descripción del catálogo de tipos en HTML y PDF. Existe otro nombre interno único, que se almacena automáticamente, ligado al nombre del fichero .DAT leído.

· *Código interno del carácter*

Parámetro [K] del menú de cabecera del código fuente.

Tipo de dato: *cadena de texto*.

Se escribe con un solo entero⁹ en base ocho (*octal*), para facilitar y asegurar su correcta descripción en cualquier contexto. Este número de tres cifras no siempre equivale a un carácter accesible en el teclado (razón de más para usarlo).

Podremos ver en la Tabla 43 las equivalencias de este código *octal* con su carácter decimal (no siempre fácil de escribir) y, en caso de existir, con el código Unicode (no siempre numéricamente equivalente).

· *Posición de origen para el siguiente carácter*

Parámetro [M] del menú de cabecera del código fuente.

Tipo de datos: *Booleano y numérico*.

Descrita en *unidades matriz*, es el valor del ancho del carácter, que no tiene por que coincidir con el ancho del glifo. Solo será así cuando el valor lógico sea verdadero. A partir de este punto se compondrá el carácter siguiente (véase Tabla 43).

6.3.3 Asignación de nombres y códigos a los signos de la escritura

· *Asignación de nombres*

El nombre tiene como base el usado por *unicode*. Se escribirá en minúsculas, si hay una mayúscula indicaran que se trata de una letra con la característica de la caja alta. No se usaran espacios. No se usarán acentos ni caracteres fuera de los *ascii* universal. Si existe un segundo tramo del nombre, precedido por un guión bajo, nos indicará algún tipo de modificación de la forma original, como por ejemplo en el caso de los caracteres que tienen más de una versión (glifo).

· *Asignación de códigos*

Para simplificar la notación numérica (dado que sólo vamos a manejar poco más de un centenar de caracteres) nos hemos basado en la tabla *ISO/IEC 8859-1 (Latin 1* o también llamada *WinAnsi*), la más utilizada en los teclados de nuestra área geográfica. Los signos de la Biblia que sabíamos que no podíamos encontrar en nuestras letras (segundas versiones, diacríticos y abreviaturas especiales, etc), les asignamos un código de carácter (tabla Latin 1) inexistente en la Biblia, como: el asterisco, corchetes, arroba, etc.

La notación numérica se ha basado en el *octal* (base ocho), pues permite utilizarse en un entorno Web y es coherente con la estrategia usada en los formatos tipográficos escritos en *PostScript (Type1, OpenType, etc)*. No siempre nuestra tabla de códigos coincide con la de *Unicode* (salvando las distancias del doble *byte*, pues aquí trabajamos con uno solo), entre otras cosas debido a que hay algunos caracteres históricos que no tienen asignación en esta moderna clasificación (véase Tabla 43).

6.3.4 Análisis del resultado

· *Elementos que configuran la representación gráfica en PDF de un carácter de la Biblia*

Para mayor detalle véase Tabla 40.

Módulo caligráfico

El filete rectangular blanco (o gris cuando no se representa ninguna retícula) nos sitúa el área modular siempre fija de 6x9 SubMódulos.

Módulo tipográfico

El filete rectangular azul nos sitúa esta área modular, variable en anchura, siempre fija de ocho SubMódulos de altura.

Línea de base

Viene representada por la línea roja inferior, donde se apoyaran los caracteres que se compongan.

Altura x

La línea roja horizontal superior nos marca esta distancia que parte de la línea de base, también roja.

Módulo m

Este espacio conceptual es el teórico cuadrado cuyo valor de lado sería la distancia entre las horizontales rojas de la línea de base y la altura x, es decir 5 SubMódulos.

SubMódulo¹⁰

Manchado por el arlequinado en color (salmón y gris), de cinco por cinco unidades matriz, es representado por la retícula del filete, de mediano grosor, en gris.

Unidad Matriz¹¹

Cada uno de los cuadrados representados por la retícula del filete mas fino, en gris.

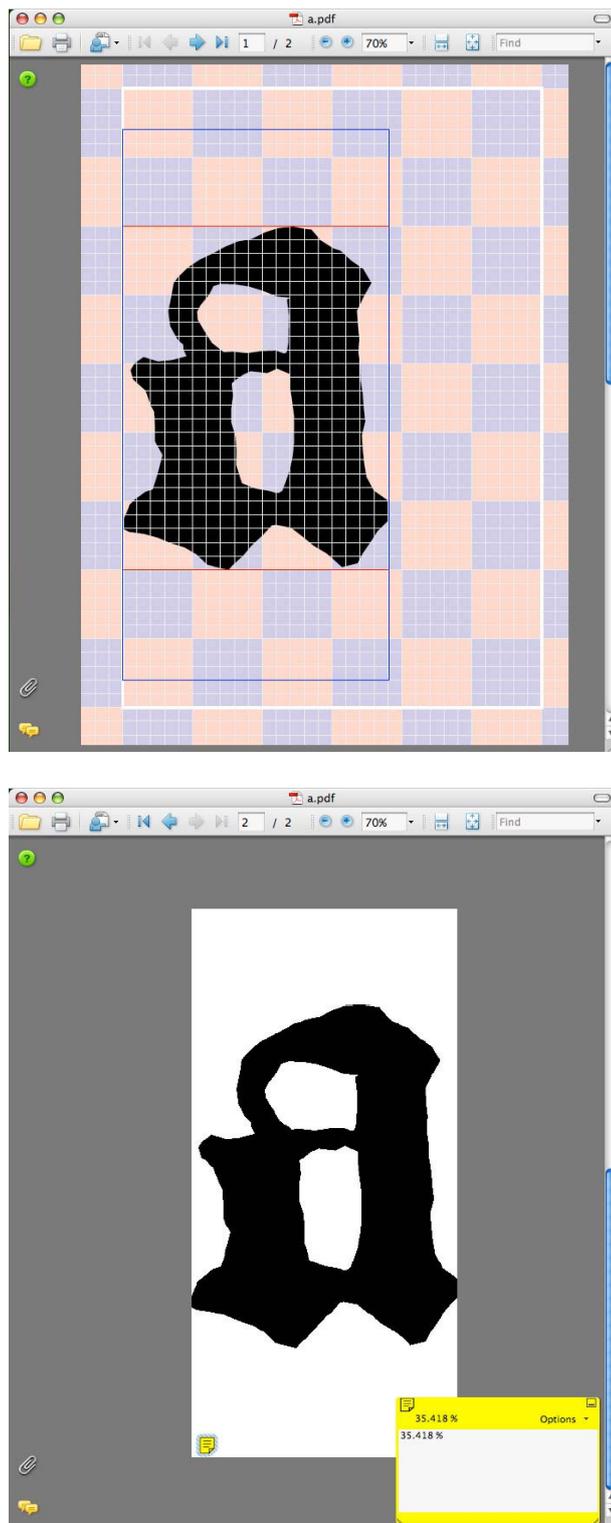


Fig. 120 Visualización del .PDF que produce la aplicación *ventaQVdat_011.ps*

Glifo vectorial

Relleno o fileteado de color negro, el glifo del carácter es construido a base de pequeñas rectas que unen los puntos leídos en el proceso de digitalización sin contacto.

Glifo de imagen

Se genera (en una segunda página del fichero de dimensión igual al Módulo Tipográfico) como resultado del análisis del porcentaje de cobertura. Cada píxel en negro es un punto que el barrido ha detectado dentro del área del glifo.

Anotación con el porcentaje de cobertura

La información en porcentaje con el resultado del análisis de cobertura acompaña, a modo de *sticky note*, la parte inferior de la imagen del glifo.

· Elementos que configuran los datos del pseudoformato .B42

Describimos, uno a uno, los datos que configuran el fichero de este pseudoformato tipográfico adaptado a las necesidades de nuestra investigación. Hay que hacer notar que, en los datos numéricos, se usan dos tipos de escalas, siempre en unidades de punto tipográfico de pica:

La escala de página: son valores adaptados por *fMd* a la dimensión ajustada a DINA4. Esta es la misma escala con la que trabajará el componedor de líneas.

La escala original de carácter: son valores puros, directos del .DAT, fruto del proceso de digitalización.

Caixa

Este *array* lleva dos valores numéricos en escala de página, con el ancho y alto del Módulo Tipográfico.

character

Este *array* lleva dos valores de *string*, con el código *octal* y el nombre interno del carácter.

DAT

Esta *string* lleva el nombre del fichero .DAT generado en el proceso de digitalización, origen de todas las deducciones espacio-modulares del carácter.

fMd

Esta *string* lleva el valor numérico clave (en nuestro caso siempre un número real), para escalar las coordenadas del .DAT a la dimensión adaptada a un DINA4, es decir la escala de página. En este mismo pseudoformato, determina valores como: *Caixa*, *OglifS* y *Modul:5*.

glif

Este *array* es la descripción congelada (operadores atrapados como literal) de los puntos que describen, línea a línea (moveto/*lineto*), cada uno de los contornos (*closepath*) que dibujan el glifo del carácter. El parámetro *setbbox* nos describe el rectángulo (*boundingbox*) que inscribe exactamente el glifo. Todo ello en escala original del carácter.

Modul:5

Este valor numérico (número real) es la unidad matriz.

OglifS

Este *array*, de un solo valor numérico, describe en escala de página la posición horizontal, desde el origen del carácter, al inicio del carácter siguiente. Su valor suele coincidir con el segundo valor del *array* del parámetro *Caixa* (ancho del Módulo Tipográfico).

Origen

Este *array* lleva dos valores numéricos, en escala original del carácter, para corregir la posición de todas las coordenadas del parámetro *glif*, de forma que se ajusten al punto de origen o,o.

UliniaAscendent

Este valor numérico entero, nos describe la cantidad de *unidades matriz* que, en vertical, sitúan la *Altura x* desde la base del Módulo Tipográfico.

UliniaDbase

Este valor numérico entero, nos describe la cantidad de *unidades matriz* que, en vertical, sitúan la *Línea de Base* desde la base del Módulo Tipográfico. Si restamos los valores de *UliniaAscendent* y *UliniaDbase* siempre tendrá que darnos 25 unidades matriz, que es el valor asignado al *Módulo m*.

6.3.5 Código fuente de ventaQVdat_011.ps

· Edición y desarrollo

Esta aplicación, escrita desde cero, se ha trabajado con diversos editores de código y en diversas plataformas. En su última etapa, el sistema de trabajo se estabilizó utilizando el editor *vi* y *Ghostsript* por línea de comando, tanto de forma local como en remoto, vía protocolo *SSH*. También se hicieron pruebas con *Adobe Acrobat Distiller* para testear la compatibilidad en un intérprete comercial de referencia. Cabe destacar que en el código hay dos algoritmos importantes reutilizados en otro proyecto de código abierto (<http://femfumcali2copi.sourceforge.net/>): *espiadic_fura2* y *espiapaq_fura3*. Estos tienen el papel fundamental de poder reescribir datos, como objetos tipo diccionario (*dictionary*) o paquete (*array*), con cualquier nivel de recursividad.

· El código PS en HTML

Para facilitar más aún la tarea de dar el máximo acceso y divulgación a las fuentes, sobretodo a los que no son desarrolladores o no están familiarizados con las herramientas que se usan en un proyecto de código libre, hemos etiquetado todo el código *PostScript* en *HTML*, facilitando la lectura con el coloreado de lo que son comentarios clave de la aplicación (en rojo), comentarios en general (en gris) y el código PS (en negro). Así mismo las líneas han sido numeradas, con etiquetado flotante (*tooltip*), de forma idéntica que en los ficheros originales.

Esta es la URL de acceso: http://gutenberginy.sourceforge.net/ventaQVdat_011.ps.html (Fig.121).

Este tipo de presentación no resta efectividad real al contenido, pues con un simple copiar/pegar podemos tener el código perfectamente interpretable en un fichero.

Otra importante ayuda que ofrece esta página, es el traductor *Google Translator* que hay en el desplegable de la cabecera. Con una simple selección de idioma, nos serán traducidos los comentarios del texto original en catalán.

· Repositorio

En la última etapa del trabajo y para facilitar el acceso a los desarrolladores, se ha usado el programa de control de versiones SVN, habilitando un repositorio de código fuente en una de las infraestructuras donde reside el proyecto de *Free Software*:

http://gutenberginy.svn.sourceforge.net/viewvc/gutenberginy/ventaQVdat_011.ps?view=log

No hay que olvidar que para usar esta herramienta es imprescindible poseer archivos .DAT del *Quick Vision*. Para ello también hay almacenados en el repositorio (*B42QuickVisionDAT.zip*) toda la colección del catálogo tipográfico en este formato.

```

% !
## ventaQVdat_011.ps v 0.0.11 llegeix els fitxers de dades .DAT generats pel QuickVision
## amb qualsevol nombre de contorns (path), per redibuixar-ne el resultat, inscrit en format A4
## en unes condicions determinades i en base a una mesura mestre (fMd) que determinem cada vegada
## en funció de les variables YBbox i Umodul. També pot explorar el % de cobertura del caràcter,
## en funció de la seva caixa (Mòdul Tipogràfic), generant-ne una imatge de comprovació.
## Escriu a disc el caràcter explorat, en un pseudoformat tipogràfic (.B42), que ens servirà
## per compondre dinàmicament una línia de la Bíblia de 42 línies amb l'eina ginyB42.ps

## Copyright (c) 2006-2011
## Concepte: Luz Ma. Rangel Alanís <quadrati@luzrangel.com>
## Codi: Marc Antoni Malagarriga i Picas <marcantoni.malagarriga.picas@ub.edu>
## Projecte de Programari Lliure: http://sourceforge.net/projects/gutenberginy/
## Repositori http://gutenberginy.svn.sourceforge.net/viewvc/gutenberginy/

## Aquest programa és programari lliure: podeu redistribuir-lo i/o modificar-lo
## sota els termes de la Llicència Pública General de GNU publicada per la Free
## Software Foundation, ja sigui la versió 3 de la Llicència, o (a la seva elecció)
## qualsevol versió posterior.
##
## Aquest programa es distribueix amb l'esperança que sigui útil, però SENSE CAP GARANTIA;
## ni tan sols la garantia implícita MERCANTIL o d'APTITUD PER A UN OBJECTIU PARTICULAR.
## Consulteu els detalls de la Llicència Pública General de GNU per a més informació.
##
## Haurieu de rebre una còpia de la Llicència Pública General de GNU junt amb aquest
## programa. En cas contrari, consulteu <http://www.gnu.org/licenses/>

```

Fig. 121 Aspecto en pantalla del código ventaQVdat_011.ps

Notas

6. Con el término *pseudoformato* nos referimos a que, a pesar de que hemos almacenado los datos del tipo de letra en un lenguaje estándar (*PostScript*), no hemos seguido ninguna de las convenciones establecidas en los formatos conocidos de tipografía digital (*type font formats*).
7. Representado gráficamente en el PDF de salida por un rectángulo azul fileteado, véase la imagen en la Tabla 40.
8. Representado gráficamente en el PDF de salida por un rectángulo blanco fileteado, véase la imagen en la Tabla 40.
9. Todos los caracteres de la Biblia se trabajan en un solo octeto (*byte*). Los 256 índices posibles de la tabla de codificación específica albergaran poco más de un centenar de tipos digitales (109).
10. El carácter del espacio ancho (coloreado en magenta) que se escribe con la barra espaciadora (espacio en blanco) y que equivale al código \040 en *octal* (32 en decimal), es también equivalente a la medida de un SubMódulo.
11. El carácter del espacio fino (coloreado en cian) que se escribe con el carácter *almohadilla* (#) o *hashmark*, y que equivale al código \043 en *octal* (35 en decimal), es también equivalente a la medida de 2.5 Unidad Matriz.

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.4 Programación del simulador para la composición de líneas con los tipos de la Biblia de 42 Líneas

6.4.1 Objetivos de esta aplicación

Todo lo referente a este capítulo es relativo a *ginyB42.ps*

1. Analizar el catálogo de tipos B42.

A diferencia de lo que comúnmente sucede en los formatos tipográficos digitales, los .B42 son ficheros que contienen un solo carácter a la vez (tantos caracteres de la Biblia, tantos .B42).

En primer lugar, este aplicativo presupone que los B42 residen en su mismo directorio, luego en caso de no encontrar ningún fichero con la extensión .B42 (extensión sensible a caja), el algoritmo se detendrá, informando de que no ha encontrado ningún carácter disponible.

En segundo lugar, si componemos una línea de texto con algún carácter que no tenemos a disposición en formato .B42, el algoritmo nos informará de ello y también se detendrá.

En tercer y último lugar, si en el análisis individual de cada .B42 se produce algún error en la lectura del código interno del carácter, el algoritmo también nos informará de ello, pero la aplicación continuará su trabajo, descartando el carácter leído para la composición.

2. Componer un solo renglón con tipos de letra B42, en base a criterios de espaciado y longitud de línea.

La composición del renglón de texto (y sólo uno a la vez), puede hacerse con tres posibles métodos, los mismos que regían el estricto proceso de composición:

- Componer rellenando el espacio de la Línea Maestra establecida, jugando con la prosa, sólo aplicable a los espacios entre caracteres.
- Componer sin límite ni ajustado de Línea Maestra ninguna y sin prosa.

- Componer con prosa fija, de 24 puntos, sin ajustado de línea ninguna, pues será la longitud de ésta la que se tome como nueva Línea Maestra disponible.

6.4.2 Análisis de cada uno de los parámetros que condicionan el resultado

- *Fichero de datos de cabecera capGinyB42.dat cuando ejecutamos en entorno Servidor*

Parámetro [0] del menú de cabecera del código fuente.

Tipo de dato: *cadena de texto* con el *path* al fichero *capGinyB42.dat*.

Al trabajar en un servidor Web, el responsable de escribir los datos dinámicos en la cabecera del código del algoritmo es el interfaz de usuario, vía navegador. Los datos escritos afectan a los parámetros [1], [2] y [3].

- *Método de trabajo escogido para el componedor*

Parámetro [1] del menú de cabecera del código fuente.

Tipo de dato: *número entero*.

Escogiendo entre cero, uno y dos, establecemos uno de los tres métodos de trabajo:

o : Compone con prosa relativa a los cuadratines del ancho de la Línea Maestra. De esta forma el valor de la prosa, que sólo se aplica a los espacios entre glifos, hace que la línea se ajuste exactamente al ancho Maestro establecido en el campo de cuadratines de la Línea Maestra.

1 : Compone sin prosa, de forma que ignora el valor del ancho Maestro del campo de cuadratines de la Línea Maestra.

2 : Compone con prosa fija (24 puntos) y establece el nuevo ancho de la Línea Maestra. De esta forma el valor resultante se salvará, para la próxima ejecución, como ancho Maestro dentro del campo de cuadratines de la Línea Maestra.

- *Línea de texto a componer*

Parámetro [2] del menú de cabecera del código fuente.

Tipo de dato: *cadena de texto*.

Potencialmente ilimitada (a la escala en que trabajamos los formatos de página suelen ser de gran formato), esta línea de texto tiene un campo máximo visualizable (por interfaz) de 90 caracteres. Así mismo, fijando de antemano el ancho de línea, se aconseja trabajar a un máximo 60 cuadratines (alrededor de 5 metros, como ancho máximo teóricamente admisible en un sistema de impresión).

En esta cadena escribiremos en notación decimal (la normal desde el teclado), como de costumbre, pero también nos será necesario escribir ciertos caracteres en notación *octal* (\###), para poder acceder a todo el rango del catálogo tipográfico de la Biblia. El mejor ejemplo de ello es la cadena de texto que contiene una de las frases maestras estudiadas (véase Apartado 5.2 y/o Fig. 112): (\21oiudas a\257ut #g\133\242u\21oit #phar\133s e\075 zar\046 de#)

· *Número de cuadratines de la Línea Maestra*

Parámetro [3] del menú de cabecera del código fuente.

Tipo de dato: *numérico*.

Esta variable nos condiciona la medida del renglón cuando trabajamos con el parámetro [1] y valor 0 (cero). De otro modo (y sólo cuando trabajemos vía Web) podrá almacenar el valor establecido como maestro, parámetro [1] y valor 2, para la ejecución posterior.

· *Trabajars con unidades matriz o con porcentajes*

Parámetro [4] del menú de cabecera del código fuente.

Tipo de dato: *Booleano*.

Esta variable se ha puesto fuera del alcance del interfaz Web (no así de la línea de comandos), de forma que en este contexto siempre trabajará con *unidades matriz*.

· *Rango de tolerancia de error según las unidades escogidas en [4]*

Parámetro [5] del menú de cabecera del código fuente.

Tipo de dato: *numérico*.

Esta variable se ha puesto fuera del alcance del interfaz Web (no así de la línea de comandos), de forma que en este contexto siempre trabajará con *tolerancia cero*.

6.4.3 Análisis del resultado

· *Elementos que configuran la representación gráfica en PDF de un renglón de la Biblia*

Formato de página

Según el método de trabajo escogido, el formato de página se adapta (o no) a la cadena de texto escrito. Hay que tener muy en cuenta que, por la escala de trabajo usada, las medidas finales de la página superarán ampliamente los formatos DIN usados en impresoras de sobremesa (A4, A3, A2) y en el supuesto que se quiera reproducir al 100% una línea real de la Biblia, deberemos acudir a impresoras de gran formato, o por el contrario, redimensionar el original, adaptando a la página en las opciones de impresión. El renglón, siempre que sea posible, llevará un margen en los cuatro costados de 100 puntos tipográficos en su escala real.

Glifo vectorial

Fileteado de color violáceo y relleno en gris, el glifo se construirá idénticamente a como ha sido leído en el proceso de digitalización, a base de pequeñas rectas.

Línea de base

Viene representada por la línea roja inferior, donde se apoyarán los caracteres que se compongan.

Altura x

La línea roja superior nos marca esta distancia tipográfica que parte de la línea de base.

Módulo m

Este espacio conceptual es el teórico cuadrado cuyo valor de lado sería la distancia entre las horizontales rojas de la línea de base y la altura x.

Módulo tipográfico

El filete rectangular azul nos sitúa el área modular del mismo nombre, variable en anchura pero siempre fija en el cuerpo o altura. La escala real de trabajo multiplicará el cuerpo original de la Gótica Textur de la Biblia por 41.212121 aproximadamente, dando como resultado un cuerpo de la B42 de unos 680 puntos de pica.

Cuadratín¹²

Su valor puede condicionar el formato de página y podemos verlo parcialmente imaginándonos el cuadrado perfecto que representaríamos con la altura (azul) del Módulo Tipográfico.

Dial y tolerancia de error para el ancho de Línea Maestra

Según el método de trabajo escogido, podremos divisar el Dial de Línea Maestra al final (por la derecha) de la página o en cualquier punto del renglón. En cualquier caso, siempre medirá el valor del ancho de la Línea Maestra en ese momento. De color negro y de espesor cero (en efecto, cero, para que siempre se aprecie como la más fina de las líneas representadas), la línea vertical del Dial puede llevar asociada un área de tolerancia que (no la vamos a poder apreciar en la ejecución Web porque no se ha activado), será representada con una banda amarillenta y transparente, siempre que el intérprete soporte transparencias; de lo contrario solo irá fileteada.

Etiquetado con el .DAT original y nombre interno del carácter

Este etiquetado flotante (*tooltip*) e interactivo (sensible al paso del ratón), activa todo el área del filete azul (Módulo Tipográfico), para señalar de cada carácter su nombre interno y el .DAT original del que es tributario. Es importante añadir que no todos los visualizadores de PDF son sensibles a este efecto del tacto del ratón (por ejemplo AdobeReader sí lo detecta).

DocInfo

Internamente, en las propiedades del documento PDF, se almacenaran unos metadatos que facilitarán su indexación y búsqueda. Aparte de los relativos al título, tema y autoría, existe un campo de palabras clave muy extenso en 3 idiomas.

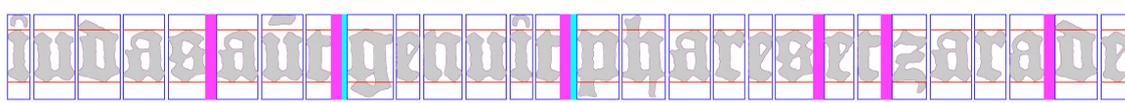


Fig. 122 Aspecto de los elementos que configuran la representación gráfica del PDF de un renglón de la Biblia con ginyB42.ps

6.4.4 Código fuente de ginyB42.ps

· Edición y desarrollo

También escrita desde cero, esta aplicación se ha trabajado con diversos editores de código y en diversas plataformas. El sistema de trabajo se estabilizó, en su última etapa, utilizando el editor *vi* y *Ghostscript* como tándem de desarrollo, tanto localmente como directamente al servidor, vía protocolo *SSH*. También se aseguró su compatibilidad con el intérprete comercial de referencia *Adobe Acrobat Distiller*.

· El código PS en HTML

Para los que no son desarrolladores o no estén familiarizados con las herramientas usadas en un proyecto de código libre, podrán acceder al código *PostScript* a través de una página *HTML*, facilitando la lectura del contenido con un coloreado para distinguir comentarios clave de la aplicación (en rojo), comentarios en general (en gris) y el código PS propiamente dicho (en negro). La numeración de líneas con etiquetado flotante (*tooltip*), facilitará la localización paralela de contenido en los ficheros originales.

Esta es la URL de acceso: <http://guttenberginy.sourceforge.net/ginyB42.ps.html> (Fig.123)

El contenido es fácilmente reutilizable, pues basta un simple copiar/pegar a un fichero para que se convierta de nuevo en interpretable.

Otra importante ayuda que ofrece esta página, es el traductor *Google Translator* que hay en el desplegable de la cabecera. Con una simple selección de idioma, nos serán traducidos los comentarios del texto original en catalán.

· Repositorio

En la última etapa del trabajo y para facilitar el acceso a los desarrolladores, se ha usado la herramienta de control de versiones *SVN*, habilitando un repositorio del código fuente en la infraestructura de *SF.net* donde reside el proyecto: <http://guttenberginy.svn.sourceforge.net/viewvc/guttenberginy/ginyB42.ps?view=log>

Para usar el componedor es imprescindible poseer los archivos *.B42* generados por *ventaQVdat_011.ps*. Para ello también hay almacenados en el repositorio (*B42.zip*) toda la colección del catálogo tipográfico en este formato.

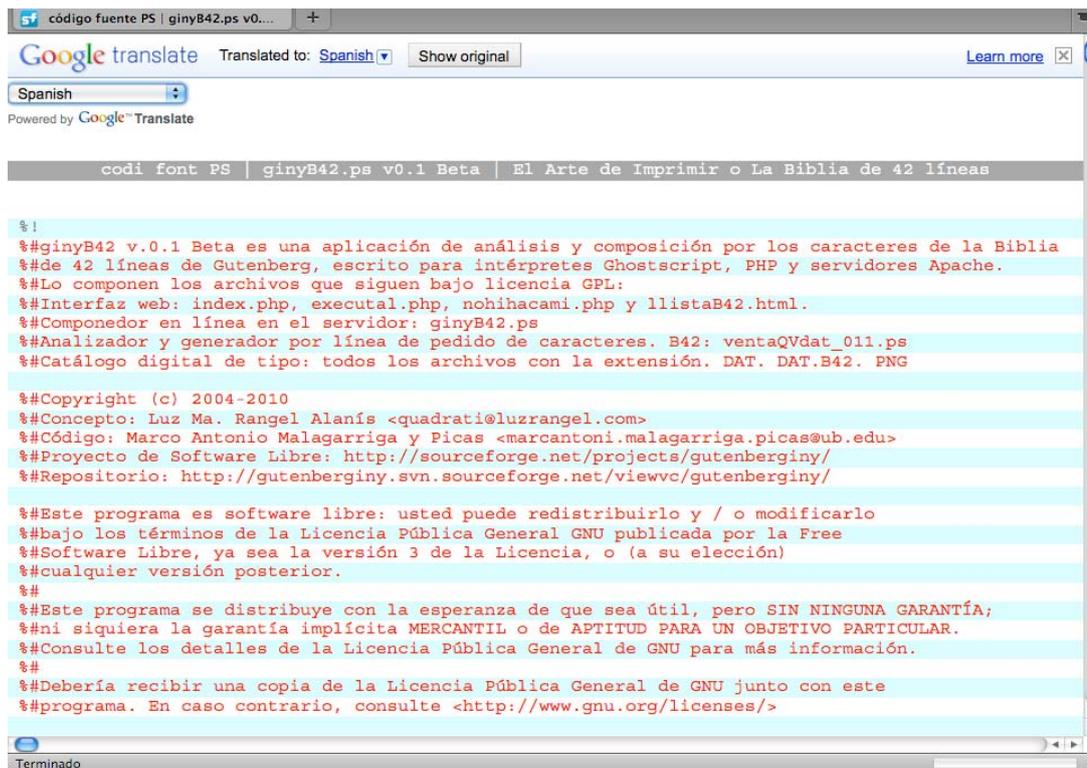


Fig. 123 Aspecto en pantalla del código ginyB42.ps

Notas

12. Diccionario de la lengua española © 2005 Espasa-Calpe. CUADRATÍN. m. impr. Blanco o espacio cuya anchura y altura es igual a la del cuerpo de letra empleado.

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.5 Programación del interfaz Web de usuario

Todo lo referente a este capítulo es relativo a los ficheros: *index.php*, *index_es.php*, *index_en.php*, *executal.php*, *executal_es.php*, *executal_en.php*, *nohahacami.php*, *iudasautgenuitpharesetzarade*, *llistaB42.html*, *llistaB42.ps*, *llistaB42.pdf*

6.5.1 Objetivos de la aplicación

1. Usabilidad del componedor ginyB42, vía navegador de Internet y sin instalación alguna

Utilizar una aplicación vía *línea de comando* a través de un Terminal de ordenador, es muy eficiente y rápido, pero poco amable y accesible al usuario en general. En cambio, hacerlo vía navegador de Internet, en una URL¹³ concreta y sin ningún tipo de protocolo de acceso, supone el grado máximo de usabilidad¹⁴ posible. No hay instalación para el usuario, y si el interfaz está bien diseñado, la eficiencia y universalidad de uso de la herramienta está garantizada. Esto es lo que hemos pretendido lograr en el simple interfaz construido para el componedor *ginyB42*, en tres idiomas:

<http://www2.ub.edu/cursusductus/hagaselaluz/B42/index.php>

http://www2.ub.edu/cursusductus/hagaselaluz/B42/index_es.php

http://www2.ub.edu/cursusductus/hagaselaluz/B42/index_en.php

El uso de esta aplicación se resume en: un campo numérico para el valor del cuadratín, un campo de datos de texto para escribir el renglón, tres botones exclusivos (*radio buttons*) para seleccionar como componer la línea y finalmente un botón de ejecución del trabajo. Complementan el interfaz los vínculos al catálogo de tipos B42 (.HTML y .PDF), el cambio de idioma, la licencia de uso y finalmente los logotipos de las entidades vinculadas al proyecto.

Como hemos dicho antes, no hay instalación para el usuario, pero si la hay para su puesta en marcha en el servidor de Internet (una sola vez y en una sola máquina). Bajo la misma filosofía de cliente/servidor, también es posible instalar localmente (ordenador personal)

ginyB42 en un entorno *localhost*, en perfecta simetría de funcionamiento que si estuviera en Internet. Detalladamente y en el Apartado 6.6 hacemos referencia de nuevo a ello.

2. Acceso al catálogo de tipos B42 en HTML y PDF

Visualizar los ciento nueve caracteres de la Biblia (más dos espaciados), saber como se llaman y saber como se escriben, es la función de las dos tablas que, en formatos diferentes, ofrecemos desde el interfaz de la aplicación *ginyB42*:

<http://www2.ub.edu/cursusductus/hagaselaluz/B42/llistaB42.html>

<http://www2.ub.edu/cursusductus/hagaselaluz/B42/documents/llistaB42.pdf>

6.5.2 Análisis de cada uno de los parámetros que condicionan el resultado

· “*el ancho actual de la Línea Maestra es de... ##.### ...cuadratines*”

(a) Parámetro vinculado a los ficheros de interfaz:

index.php

index_es.php

index_en.php

executal.php

executal_es.php

executal_en.php

iudasautgenuitpharesetzarade

Tipo de campo: *numérico*.

En primera instancia, siempre lee el contenido del fichero *iudasautgenuitpharesetzarade* que alberga el valor de Línea Maestra en cuadratines, establecido por el parámetro (e) en algún momento. De no ser así (por ejemplo, por no existir), el interfaz nos daría un error, tipo...

Warning: fopen(*iudasautgenuitpharesetzarade*) [*function.fopen*]: failed to open stream: No such file or directory in...

Puede ser modificado para usarse puntualmente (c), o para ser fijado como nuevo valor estable (e) y escrito de nuevo en el fichero *iudasautgenuitpharesetzarade*.

· “*\210iudas a\257ut #g\133\242u\210it #phar\133s e\075 zar\046 de#*”

(b) Parámetro vinculado a los ficheros de interfaz:

index.php

index_es.php

index_en.php

executal.php

executal_es.php

executal_en.php

Tipo de campo: *texto*.

Este campo visualiza hasta 90 caracteres, pero podemos escribir más (nótese que un carácter *octal* nos ocupa cuatro). El texto pasa un pequeño filtro que elimina espacios, tabuladores o retornos de carro y otros cuerpos extraños al inicio de la frase.

Fig. 124 Aspecto de la interfaz para el *ginyB42*

· “Compón con prosa relativa a los cuadratines del ancho de la Línea Maestra”

(c) Parámetro vinculado a los ficheros de interfaz:

index.php
index_es.php
index_en.php
executal.php
executal_es.php
executal_en.php

Tipo de campo: *botón de radio*.

La opción fuerza al componedor a escribir el texto en base al ancho del renglón escrito en el campo (a). La prosa aquí se aplica solo a los espacios entre caracteres y puede darse el caso que obligue a los tipos a “colisionar” entre ellos (prosa negativa), o sea, a solaparse (efecto imposible en el mundo real del componedor con tipos de plomo). Antes que prohibirlo, hemos preferido dejar que se visualice este extraño efecto.

· “Compón sin prosa”

(d) Parámetro vinculado a los ficheros de interfaz:

index.php
index_es.php
index_en.php
executal.php

```
executal_es.php  
executal_en.php
```

Tipo de campo: *botón de radio*.

No condiciona en nada al componedor ni establece nada nuevo. Solo respeta la posición fijada de Línea Maestra, representada por esta línea negra vertical que hemos bautizado como Dial.

· *“Compón con prosa fija [24 puntos] y establece el ancho de la Línea Maestra”*

(e) Parámetro vinculado a los ficheros de interfaz:

```
index.php  
index_es.php  
index_en.php  
executal.php  
executal_es.php  
executal_en.php
```

Tipo de campo: *botón de radio*.

Es la opción que teóricamente se asemeja más al estilo puro del trabajo de Schöffer y Gutenberg. Si con ella componemos la frase maestra de la Biblia (la que ofrece por defecto el interfaz), obtendremos un renglón muy ajustado a la realidad de cualquier original de los que aún se conservan.

6.5.3 Análisis del resultado

Los tres idiomas del interfaz (catalán, español e inglés), se mantienen parcialmente también en la segunda pantalla (negra) de ejecución y resultados (internamente el algoritmo habla catalán). El trabajo va lo suficientemente rápido (es el Servidor de Internet quién trabaja y no nuestra máquina local), como para tolerar la pequeña espera que supondrá llegar al vínculo del .PDF final (en amarillo), que nos descargará el resultado en nuestro ordenador.

Los ejemplos aquí expuestos, se han confeccionado en base a la frase maestra...

```
\210iudas a\257ut #g\133\242u\210it #phar\133s e\075 zar\046 de#
```

· *Elementos que configuran los datos de la pantalla negra de ejecución*

Análisis de los caracteres B42 disponibles

El diseño dinámico del componedor permite que, previamente a empezar su tarea principal, revise y 'cargue' en su juego de caracteres (*encoding*) todos los .B42 encontrados en el mismo directorio de trabajo. Una vez revisado su formato y datos clave, se lista por pantalla (mensajes en gris), uno a uno, el nombre del fichero, el código *octal* y nombre interno. Ejemplo...

```
1-1-18-u.DAT.B42          \165    letraminusculau
```

...finalmente son contados y se emite un mensaje con su totalidad...

```
... hi ha ... 109 ... characters disponibles
```

Número de espacios entre caracteres a recalcular

En este cálculo se ignoran los espacios entre palabras y entre los dos caracteres de espaciado (\o4o y \o43). Así, en la frase maestra, a pesar de tener 28 caracteres, el algoritmo nos valida sólo 21 para repartirse la prosa...

```
21 ... nombre d'espais entre caracters a recalcular
```

Valor en puntos de prosa

O dicho de otro modo: ¿que valdrá la prosa en cada espacio entre caracteres? El cálculo que emite este mensaje sólo se produce si hemos compuesto el renglón con la opción *“Compón con prosa relativa a los cuadratines del ancho de la Línea Maestra”*. Es el resultado de la división del *valor total, en puntos, de prosa para ajustar* (504.003 en el ejemplo) entre el *número de espacios entre caracteres a recalcular* (21 también en el ejemplo). De aquí el número flotante hasta la diezmilésima de punto tipográfico...

```
24.0001 ... valor en punts de prosa
```

Valor fijo en puntos de prosa

No es fruto de ningún cálculo (de ahí su redondez), si no que es asignado de forma fija para cada espacio entre caracteres, por la opción *“Compón con prosa fija [24 puntos] y establece el ancho de la Línea Maestra”*...

```
24 ... valor FIX en punts de prosa
```

Número de caracteres de la frase

Este dato se emite siempre, sea cual fuere el método escogido para componer el renglón. Nunca se incluyen los dos tipos de espaciado...

```
28 ... nombre de caracters de la frase
```

Valor de ancho en puntos de los... ##.### cuadratines de anchura de la Línea Maestra

Este dato se emite siempre, sea cual fuere el método escogido para componer el renglón. El primer valor que precede el mensaje (ancho, en puntos, de la línea escrita), es fundamental para cuando juguemos con la prosa y así poder calcular el *valor total, en puntos, de prosa para ajustar*. El segundo valor, en cuadratines, siempre nos situará el Dial gráfico y podrá ser almacenado o no, en función del método...

```
8786.49 ... valor d'amplada en punts dels 12.9137 quadratins
d'amplada de la Linia Mestre
```

Anchura actual de la línea sin prosa

Este dato se emite siempre, sea cual fuere el método escogido para componer el renglón. Coincide con el valor de *El ancho de línea es de... ###.### puntos | ##.### cuadratines*, si trabajamos con el método *“Compón sin prosa”*...

```
8282.49 ... amplada actual de la linia sense prosa
```

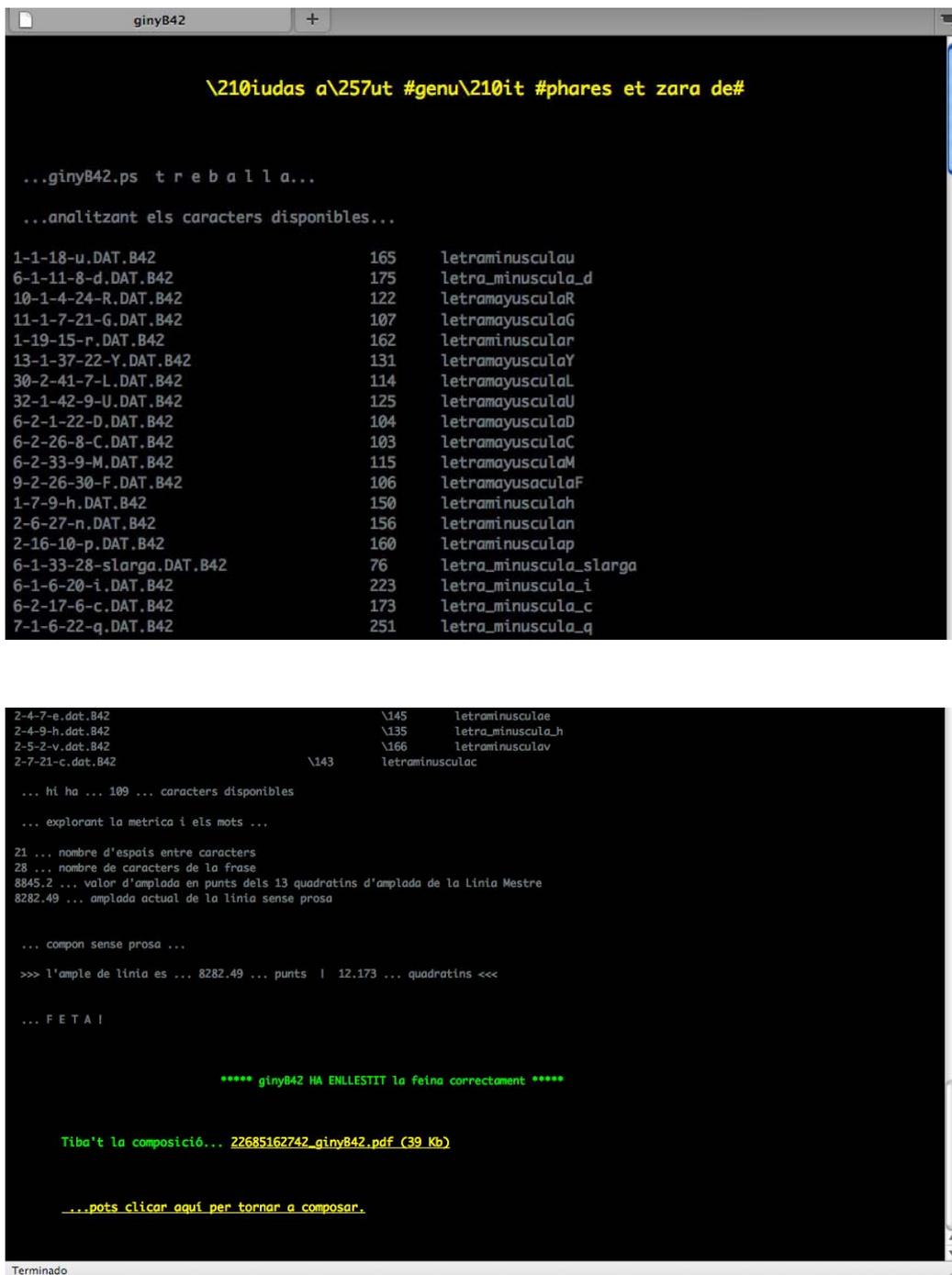


Fig. 125 Visualización de la pantalla negra de ejecución.

Valor total, en puntos, de prosa para ajustar

El cálculo que emite este mensaje sólo se produce si hemos compuesto el renglón con la opción “Compón con prosa relativa a los cuadratines del ancho de la Línea Maestra”. Es el valor resultante de restar la *Anchura actual de la línea sin prosa* de *El ancho de la Línea Maestra fijado*...

504.003 ... valor total, en punts, de prosa per ajustar

Valor total, en puntos, de prosa para añadir
 Al ser producto de dos enteros (*Valor fijo en puntos de prosa por Número de espacios entre caracteres a recalcular*), siempre que optemos por el método “*Compón con prosa fija [24 puntos] y establece el ancho de la Línea Maestra*”, su valor es también un entero...

```
504 ... valor total, en punts, de prosa per afegir
```

El ancho de la Línea Maestra fijado es de... ####.## puntos | ##.#### cuadratines
 Actúa como mensaje de cierre, siempre que optemos por alguno de los dos métodos “*Compón con prosa relativa a los cuadratines del ancho de la Línea Maestra*” o *Compón con prosa fija [24 puntos] y establece el ancho de la Línea Maestra*”...

```
>>> l'ample de Linia Mestre fixat es de ... 8786.49 ... punts |  
12.9137 ... quadratins <<<
```

El ancho de línea es de... ####.## puntos | ##.#### cuadratines
 Actúa como mensaje de cierre, siempre que trabajemos con el método “*Compón sin prosa*”...

```
>>> l'ample de linia es ... 8282.49 ... punts | 12.173 ...  
quadratins <<<
```

· *Elementos que configuran los datos del catálogo de tipos B42*

Catálogo en HTML

La ventana vertical, flotante, que se abre desde el vínculo del interfaz (*caracteres de la Biblia de 42 líneas, disponibles para componer*), nos muestra, uno debajo de otro, toda la colección de los ciento nueve (+2) caracteres que se han editado. Esta tabla se genera dinámicamente, cada vez que activamos una ejecución desde el interfaz Web, a partir de un algoritmo específico de *ginyB42*.

Cuatro son los datos que definen un carácter (al uso) según el encabezado, de izquierda a derecha (Fig. 126):

ASCII

Este resbaladizo acrónimo, señala cuál es el carácter equivalente decimal del glifo de la Biblia. O en otras palabras: cuál es la tecla de nuestro teclado que probablemente hemos de pulsar para representar dicho glifo. Esto último será verdad siempre que trabajemos con un teclado equivalente al juego de caracteres *ISO/IEC 8859-1 (Latin 1* o también llamado *WinAnsi*).

OCTAL

Es la más universal de las notaciones de nuestro componedor. Siempre vamos a encontrar un teclado donde poder escribir con un “\” (*backslash*) y una secuencia de tres números cualquiera entre 0 y 9. Cuatro dígitos para representar a un glifo de la Biblia.

GLIF

Hay una universal tendencia a designar como glifo (*glyph*) a la simple forma gráfica (pixel o vector) que construye un carácter. La forma completamente desnuda de toda la inteligencia que el medio digital puede infundirle (básicamente cifrados de información y algoritmos de comportamiento).

NOM INTERN

Descrito con cierto detalle en el capítulo 6.3.2, el *nombre interno del carácter* nos será de ayuda para describir e identificar aquellas formas tipográficas poco frecuentes, en desuso o cuya forma en los tipos de la Biblia quede alejada visualmente de la forma moderna equivalente.

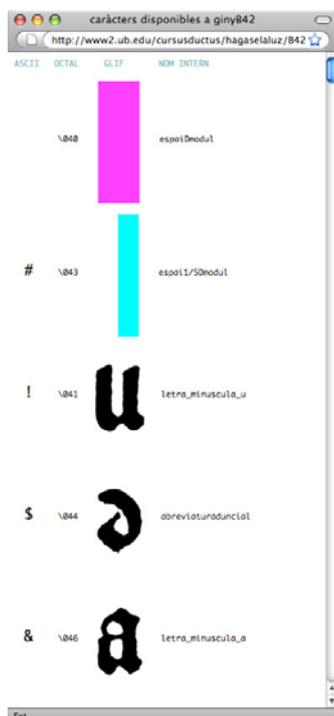


Fig. 126 Ventana flotante del catálogo de tipos en HTML.

Catálogo en PDF

Tabla idéntica a la del formato HTML, sólo que más fácilmente imprimible en formato DIN (véase Tabla 43). Para generarla, se ha escrito una nueva utilidad (*llistaB42.ps*, desvinculada por el momento del proyecto), que *quick&dirty*¹⁵ formatea dinámicamente los datos de las B42. Se apuntó como posibilidad futura vincular este algoritmo a *ginyB42* para que pudiera generar el catálogo en tiempo real, tal y como se hace en HTML.

· Otros elementos vinculados al interfaz

Licencia de Uso

Nos hemos acogido a la licencia matriz del código libre GNU GPL3, de la que cuelga inicialmente una descripción corta pero suficiente:

http://www2.ub.edu/cursusductus/hagaselaluz/B42/ginyB42_GNU_GPL_esp.txt

Vínculo a la página del proyecto SourceForge

El logotipo genérico de este portal de proyectos nos lleva directos a toda la infraestructura del *Gutenberg Bible line composer: ginyB42*

<http://sourceforge.net/projects/gutenberginy/>

Vínculo a la Universitat de Barcelona

http://www.ub.edu/masteroficial/tipografia/index.php?lang=ca_ES

Vínculo a FemFum.com

Este portal de creación de *software* gráfico ha colaborado activamente en esta investigación a través de sus aplicaciones de código libre basadas en PostScript y PDF:

<http://www.femfum.com>

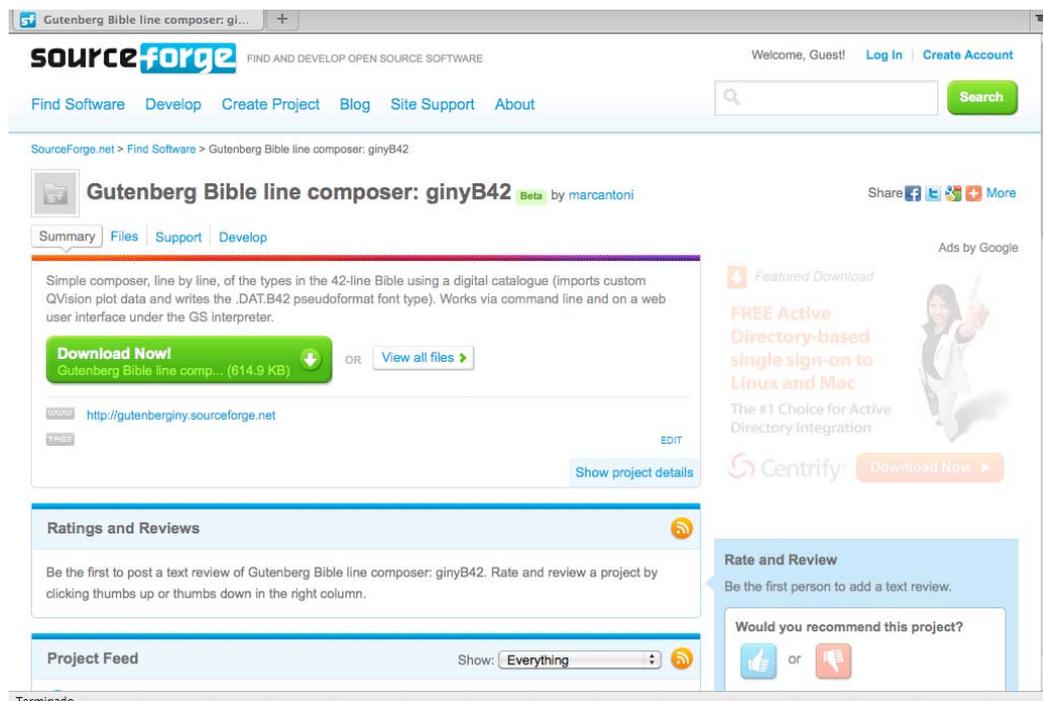


Fig. 127 Visualización de la descarga del proyecto *Gutenberg Bible line composer: ginyB42* en Source Forge.

6.5.4 Código fuente del interfaz de usuario

· Edición y desarrollo

El conglomerado de código necesario para construir un interfaz de usuario para envolver adecuadamente el motor gráfico del componedor, en este caso: HTML, PHP, JavaScript y CSS, por su complejidad (y al mismo tiempo la necesidad de simplificación máxima), nos ha alejado de la idea de utilizar un *software* de diseño Web. Por ello, todo el código se ha escrito enteramente a mano con el editor *vi* y con la ayuda de diversos navegadores para testear (Firefox, Safari, Chrome, MSIE Explorer, Opera), vía *Internet* o en simple *localhost*.

El resultado final ha sido una estructura de ficheros de interfaz muy ligera y dinámica, que hacen el trabajo sin demasiado *glamour* gráfico pero con eficiencia.

· *Repositorio*

En la última etapa del trabajo hemos empaquetado (zip) todo el conjunto de archivos que forman la cadena del interfaz, para servirla bajo el control de versiones SVN, en el repositorio donde residen el resto de ficheros del proyecto: <http://gutenberginy.svn.sourceforge.net/viewvc/gutenberginy/>

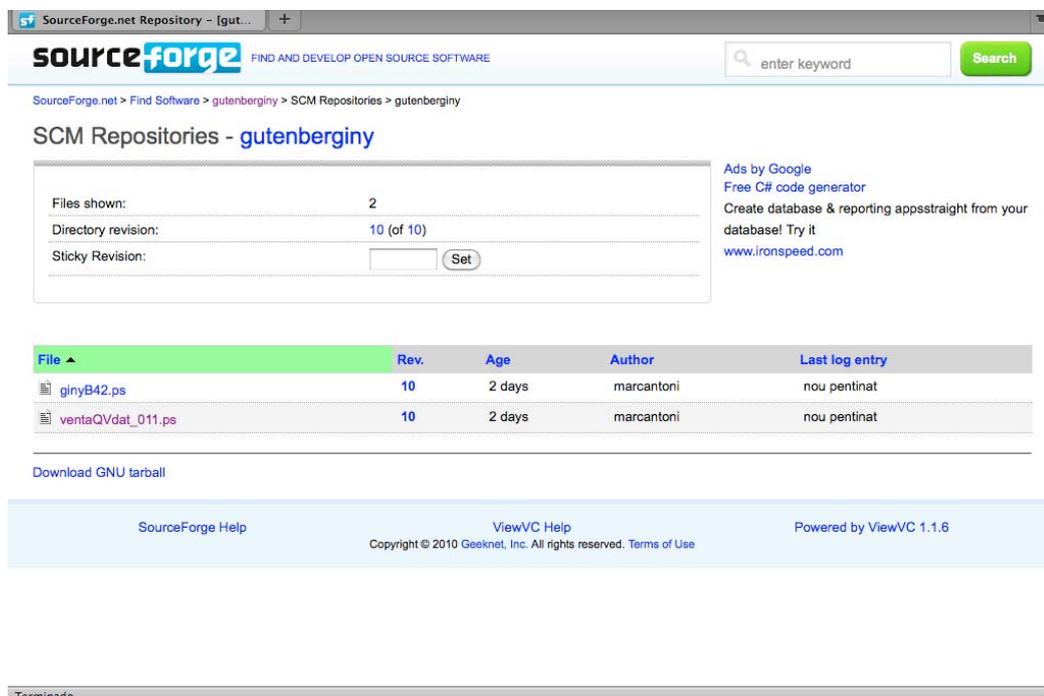


Fig. 128 Visualización del repositorio en Source Forge.

Notas

13. URL, significa Uniform Resource Locator, es decir, localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web en Internet. Wikipedia, la enciclopedia libre [en línea]. Disponible en: <http://es.wikipedia.org/wiki/URL>

14. La usabilidad (del inglés *usability*) es la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. En la interacción persona-ordenador, se refiere a la claridad y la elegancia con que se diseña la interacción con un programa de ordenador o un sitio web. Wikipedia, la enciclopedia libre [en línea]. Disponible en: <http://es.wikipedia.org/wiki/Usabilidad>

15. Rápido y sucio (del inglés *Quick-and-dirty*) es un término usado en referencia a una forma sencilla de solucionar un problema. Este uso es popular entre los hackers, que lo usan para describir una solución primitiva o una implementación que es imperfecta, poco elegante, o incluso inadecuada, pero que resuelve o enmascara el problema, y generalmente es más rápida y sencilla de usar que buscar una solución apropiada. Las soluciones Quick-and-dirty a menudo se centran en un caso concreto de un problema en lugar de arreglar la causa del problema general. Por eso, se usan algunas veces para mantener una parte del software o hardware funcionando temporalmente hasta que se pueda encontrar una solución adecuada. La frase también se usa frecuentemente para describir documentos o tutoriales que sólo dan una visión general de cómo hacer algo, sin entrar en demasiados detalles sobre por qué o cómo funciona. Wikipedia, la enciclopedia libre [en línea]. Disponible en: http://es.wikipedia.org/wiki/R%C3%A1pido_y_sucio

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.6 Instalación y puesta en marcha de las aplicaciones por la línea de comandos de un Terminal y en un sistema de servidor WWW

6.6.1 Requisitos mínimos

· *Para trabajar desde un Terminal por línea de comandos*

Máquina y sistema operativo compatibles con los requisitos de instalación de Ghostscript (GS):

<http://ghostscript.com/doc/9.00/Install.htm>

Ghostscript 8/9 (intérprete de lenguaje PostScript), bajo licencia de código libre (GNU GPL) y en cualquier plataforma: <http://ghostscript.com/>

Software visualizador de PDF:

AdobeReader <http://get.adobe.com/reader/>

GSview <http://pages.cs.wisc.edu/~ghost/gsview/index.htm>

ApplePreview <http://www.apple.com/macosx/features/300.html#preview>

Evince <http://projects.gnome.org/evince/>

XPDF <http://www.foolabs.com/xpdf/>

· *Para trabajar desde un sistema de Servidor WWW*

La máquina y sistema operativo que actúe como servidor local o de Internet deberá ser compatible con los requerimientos de las aplicaciones descritas seguidamente:

Servidor HTTP Apache 1.3.#, bajo licencia de código libre y en cualquier plataforma:

<http://apache.org/>

Intérprete PHP 4/5, bajo licencia de código libre y en cualquier plataforma:

<http://php.net/>

Ghostscript 8/9 (intérprete de lenguaje PostScript), bajo licencia de código libre (GNU GPL) y en cualquier plataforma:

<http://ghostscript.com/>

Alternativamente a Ghostscript, esta aplicación también puede funcionar con el software comercial Adobe Acrobat Distiller (bajo licencia propietaria) y en cualquier plataforma:

<http://www.adobe.com/devnet/acrobat/wwdistiller.html>

<http://www.adobe.com/products/acrdis/>

En este caso, los requisitos de funcionamiento interactivo, por línea de comandos o bajo un servidor de Internet, cambiarían substancialmente.

· *Necesidades del usuario*

Navegador de Internet (MozillaFirefox, AppleSafari, GoogleChrome, MSInternetExplorer, Opera).
Visualizador de PDF (AdobeReader, Gsview, ApplePreview, Evince, XPDF).

6.6.2 Instalación

· *Para trabajar desde un Terminal por línea de comandos*

Linux/Unix

Con toda probabilidad, el intérprete Ghostscript (GS) nos vendrá preinstalado sea cual sea la distribución con la que trabajemos (Debian, Ubuntu, Fedora, Linkat, Suse, RedHat, Mandriva, etc), pero en el remoto caso que no fuera así o quisiéramos actualizarla, podremos instalarla a voluntad, de forma manual o automatizada. Lo mismo sucede en sistemas clásicos de Unix (Sun y otros).

Instalación manual: vía código fuente

<http://ghostscript.com/doc/9.00/Install.htm>

<http://www2.ub.edu/cursusductus/femfum/Mo9o6/instalGSmacOSXlinux.html>

Instalador automatizado: vía *rpm*, *dpkg*, *YaST* u otros comandos (según distribución)

MacOSX

Instalación manual del GS: vía código fuente, ídem a Linux/Unix

<http://www2.ub.edu/cursusductus/femfum/Mo9o6/instalGSmacOSXlinux.html>

Instalador automatizado del GS: vía DMG para GS v8.71 de la LinuxFoundation

<http://www.openprinting.org/download/printdriver/macosx/gplgs-8.71.dmg>

MSWindows

Instalador automatizado del GS: vía exe (última versión hasta la fecha gs90ow32-bit o gs90ow64)

<http://sourceforge.net/projects/ghostscript/files/>

· *Para trabajar desde un sistema de Servidor WWW*

Instalación del intérprete PostScript (GS):

Linux/Unix, MacOSX, MSWindows

En servidores de Internet reales es menos probable que Ghostscript venga preinstalado. Más probablemente aún, en 'hostings' de alquiler con prestaciones bajas no podremos acceder a este servicio, pues suele formar parte de los servicios dedicados y a servidores virtuales.

Si se poseen los privilegios para una instalación remota de Ghostscript al servidor (por ejemplo vía SSH), los mismos recursos descritos en el apartado anterior (Terminal por línea de comandos) nos serán suficientes para todas las plataformas, y también de forma idéntica lo será para trabajar con una máquina en 'localhost'. De lo contrario, será el administrador del sistema quién tendrá que ejecutar esta tarea.

Instalación del servidor HTTP Apache:

Linux/Unix

Obviamente, es el corazón del 'hosting' en servidores reales de Internet para estas plataformas.

Para trabajar en 'localhost', Apache viene de serie en todas las distribuciones.

MacOSX

Es también el corazón del 'hosting' en servidores reales de Internet para estas plataformas.

Apache viene de serie en todos los sistemas X en máquinas locales (ordenadores convencionales) pero hace falta activarlo para usarlo en 'localhost' y para ello nos basta encender un botón en...

System Preferences > Sharing > Personal Web Sharing

Después ya podemos acudir al navegador para comprobar que nuestro servidor local (localhost) ya es activo, escribiendo:

<http://localhost/> ...o... <http://localhost/~miCuentaDeTrabajo>

MSWindows

Sea cual sea la versión de MSWindows con la que trabajemos (2K, XP, Vista, W7), este sistema operativo tiene su propia filosofía de servidor Web (WindowsServer), completamente ajena a la operativa que presentamos aquí. No obstante este sistema operativo también permite la instalación de servidores Apache en distribuciones integradas de código libre:

<http://www.wampserver.com/>

<http://www.apachefriends.org/en/xampp-windows.html>

Instalación del intérprete PHP:

Linux/Unix

En servidores de Internet reales basados en Linux/Unix, inclusive en los alojamientos básicos, probablemente tengamos ya este servicio dado de alta (de lo contrario solo será cuestión de mejorar las prestaciones del 'hosting'). Para trabajar en 'localhost' en una máquina local, PHP viene de serie en todas las distribuciones.

MacOSX

En todos los sistemas X (10.0/10.6), en máquinas locales, PHP viene preinstalado de serie pero hace falta activarlo manualmente manipulando una serie de líneas del fichero de configuración *httpd.conf* del servidor Apache. Esta operación es delicada y hace falta hacerla desde el Terminal con privilegios de administrador. Aun que resulte redundante, a un usuario básico le resulta más fácil instalar de nuevo el PHP en uno de los paquetes integrados de libre distribución para MacOSX:

<http://www.apachefriends.org/en/xampp-macosx.html>

MSWindows

En servidores de Internet basados en Windows, difícilmente la prestación del PHP forma parte del servicio inicial de alojamiento. De todos modos, solo será cuestión de mejorar las prestaciones del 'hosting', en cada caso, para incorporar el uso de PHP o de lo contrario pasarse a un servidor Linux directamente. En máquinas locales, el caso del PHP se resuelve de la misma manera que para instalar Apache, a través de los paquetes integrados que lo incorporan:

<http://www.wampserver.com/>

<http://www.apachefriends.org/en/xampp-windows.html>

6. Reconstrucción vectorial y simulación compositiva de los tipos de la Biblia de 42 líneas

6.7 Proyecto de Código Libre

La filosofía con la que hemos creado las herramientas de este apartado, nos ha permitido beneficiarnos de la estructura que SourceForge¹⁶ ofrece a toda la comunidad de desarrolladores de proyectos abiertos, en todo el mundo.

Este proyecto empezó a desarrollarse mentalmente a principios de 2005 y, en octubre de 2009, con versiones de código de una cierta madurez, fue dado de alta oficialmente en este portal de referencia como *Gutenberg Bible line composer: ginyB42*.

El ánimo de los autores, más allá de esta investigación, es de compartirlo, mejorarlo y evolucionar, si cabe, hacia usos paralelos o derivados, a los planteados inicialmente como objetivo para este desarrollo.

6.7.1 Secciones clave del proyecto

La infraestructura básica de los proyectos *Open Source* o de *Free Software*¹⁷ es parecida en todos ellos y, por ende, los portales que los acogen ofrecen servicios y espacios comunes donde gestionar la información, siempre bajo el binomio conceptual de trabajo remoto *cliente/servidor*.

Los apartados aquí descritos brevemente, han ido evolucionando poco a poco en base a su utilidad real. El tiempo y los futuros usuarios marcarán quizás otras necesidades que por ahora no han sido contempladas aquí (WiKi, enlace a redes sociales, canal IRC, etc).

· *Web Site*

<http://gutenbergginy.sourceforge.net>

Esta página, con ánimo de portal de entrada, fue creada básicamente con motivo de la ponencia “*La B42: los orígenes de la industria gráfica*” presentada en la Conferencia 2009 de la Association Typographique Internationale (AtypI) en Ciudad de México. Nos dará acceso tanto a los recursos relacionados al portal SourceForge, como a todo el material elaborado para la ocasión de dicho congreso tipográfico.

· *Summary*

<http://sourceforge.net/projects/gutenberginy/>

Un condensado resumen de los objetivos del proyecto, tecnologías utilizadas, público potencial al que se dirige y las últimas novedades en sus diversos apartados, haciendo especial hincapié en la más reciente actualización del *software* para descargar.

· *Mailing List*

<https://lists.sourceforge.net/lists/listinfo/gutenberginy-b42eines>
<gutenberginy-b42eines-owner@lists.sourceforge.net>

Es la ventana al exterior del proyecto. La URL nos enlaza con la página para darse de alta a la lista de correo que vincula a toda la comunidad de usuarios y desarrolladores implicados. La dirección de *email* nos comunica directamente con los que administran este proyecto.

· *Help*

<http://sourceforge.net/projects/gutenberginy/forums/forum/984784>

A modo de pequeñas recetas de ayuda, aquí se contestan algunas de las preguntas más frecuentes que podrá hacerse el usuario de estas herramientas.

· *Tracker*

http://sourceforge.net/tracker/?atid=1152794&group_id=271083&func=browse

Se describe con cierto detalle el estado de resolución de errores, omisiones y otros problemas, encontrados en los diversos componentes que configuran el *software* del proyecto.

· *Code (SVN)*

<http://gutenberginy.svn.sourceforge.net/viewvc/gutenberginy/>

Es el llamado repositorio del código fuente, donde hay las últimas versiones de los ficheros clave del desarrollo (listas para descargar vía [Download GNU tarball](#)), administradas a través del aplicativo SVN, el cual permite gestionar colaborativamente el trabajo de diversos programadores que inciden sobre un solo fichero de código.

· *Código PS en HTML*

http://gutenberginy.sourceforge.net/ventaQVdat_011.html

<http://gutenberginy.sourceforge.net/ginyB42.html.html>

Los dos enlaces nos muestran de la manera más simple y accesible (una página html), el código fuente *PostScript* de las dos aplicaciones principales del proyecto, con también la ayuda de un coloreado básico de la sintaxis y la numeración de líneas.

· *Download files*

<http://sourceforge.net/projects/gutenberginy/files/>

Aquí podemos descargar íntegramente la última versión del conjunto de ficheros (programas, interfaz, catálogo de tipos, etc.) que configuran todo el proyecto. De modo que con lo que viene empaquetado en el .ZIP de descarga podremos instalar el aplicativo donde queramos. Exactamente lo mismo que podemos ver funcionando en el siguiente apartado de la *Herramienta OnLine*.

· *Herramienta OnLine*

http://www2.ub.edu/cursusductus/hagaselaluz/B42/index_es.php

http://www2.ub.edu/cursusductus/hagaselaluz/B42/index_en.php

<http://www2.ub.edu/cursusductus/hagaselaluz/B42/>

Es el componedor de líneas de la Biblia, que funciona con un simple navegador. El interfaz (versiones en español, inglés y catalán), nos ofrece también enlaces al catálogo de tipos, ayuda y copia de la licencia GNU GPL3 con la que se distribuye el proyecto.

6.7.2 Alguna idea más para su utilidad futura

A los autores de esta parte del proyecto, uno de los objetivos que nos gustaría llevar a la realidad es poder comprobar la utilidad de estas herramientas más allá de esta Tesis Doctoral. Este deseo ya se ha materializado parcialmente, dada la reutilización que se ha llevado a cabo recientemente, de parte del código del componedor, en otro proyecto tipográfico para la llamada *Open Web*¹⁸.

Otra línea de uso a ofrecer para los amantes del *revival* tipográfico o los estudiosos que rescatan del olvido tipos de letra que, o sólo existen en plomo o sólo existen en papel, es dar cuenta del proceso utilizado en esta investigación como un buen *método expres* para una rápida y precisa puesta en *escena digital* de material tipográfico que sería muy laborioso de otro modo. El flujo de trabajo de digitalizar sin contacto y vectorizar empaquetando como un tipo de letra digital enlazable a un componedor Web, podría demostrarse realmente útil y efectivo en estos casos.

Pero el mismo espíritu abierto del proyecto o el concepto mismo de reutilización del código, hace impredecible cuál será el próximo *mashup* que nos sorprenderá en un futuro. Sea cuál sea el devenir de nuestra aportación, nos encantará que se produzca.

Notas

16. <http://sourceforge.net/about>

17. Estos dos conceptos tienden a confundirse y son substancialmente distintos. Véase éste artículo de Richard Stallman, uno de los fundadores del proyecto GNU e incansable divulgador del conocimiento y cultura libres [en línea]. Disponible en: <http://www.gnu.org/philosophy/open-source-misses-the-point.html>

18. Con motivo del Drumbeat Barcelona Festival 2009, se dio de alta como proyecto en la Mozilla Foundation, una iniciativa basada en un componedor de caracteres tipográficos localizados a través de Google Maps [en línea]. Disponible en: <http://www.drumbeat.org/project/m%C3%A0quina-descriure-de-google-maps>