

## IMAGEN DIGITAL



## El lenguaje PostScript

Gugsms, 2003

### 1. Introducción

PostScript es un "lenguaje de descripción de página", es decir, es un lenguaje de programación que se usa para decirle a una máquina destinada a imprimir cómo y qué debe imprimir.

Un documento PostScript, en realidad, es un pequeño programa que le dice a una máquina qué, cómo y dónde imprimir, paso a paso. Los documentos PostScript se destinan a imprimirse en aparatos PostScript, es decir, en aparatos que tienen un dispositivo interno capaz de descifrar el código que reciben y convertirlo en simples puntos de impresión ("aquí imprimo, aquí no, aquí sí, aquí también, etc...").

Un documento o fichero PostScript, como programa que es, debe atenerse a unas reglas de construcción muy precisas. Aparte de esto, puede contener dentro datos de todo tipo: Textos, Imágenes binarias (es decir, "fotos") y descripciones matemáticas de gráficos (es decir, "dibujos vectoriales").

La idea de PostScript, que parece bastante básica y tonta, no lo es en absoluto. La existencia de este lenguaje (y de otros similares) permite algo que en la actualidad parece básico pero que hace muy pocos años no lo era: La portabilidad de los documentos de una impresora o filmadora a otra.

Antes, si querías filmar un documento en un sistema de trabajo informatizado, debías componerlo o meterlo a través de una terminal de dicho sistema que era "un sistema completo" o "propietario" (*proprietary*). Es decir, que no podías acoplar y usar a la vez piezas y programas de distintos fabricantes. Y de pasar documentos de un sistema a otro para imprimir, mejor casi ni hablamos. Las calidades y resoluciones variaban según los aparatos y había que adaptar los documentos a cada uno.

## Los fundadores de Adobe e inventores del lenguaje PostScript.



John Warnock

Charles Geschke

PostScript fue inventado y desarrollado por la firma californiana Adobe hacia 1984. Apple lo adaptó en 1985 a sus impresoras láser y fue un rotundo éxito, gracias en parte a una aplicación de diseño de páginas llamada PageMaker que fabricaba la compañía Aldus.

Cuando una empresa de gran solvencia en las artes gráficas, Linotype, adoptó PostScript el éxito quedó asegurado.

El hecho es que PostScript no es el único Lenguaje de Descripción de Página (LDP) que existe, pero es el que ha tenido más éxito y el que se ha convertido en un estándar de las artes gráficas.

### 2. Cómo funciona PostScript

Para ser "adaptable", PostScript es un lenguaje de los denominados "interpretados". Es decir, no le habla directamente a la máquina, sino que necesita un procesador (un dispositivo físico o un programa residente en el ordenador) que actúe como intérprete traductor entre el código PostScript "universal" y la máquina. Eso es lo que se llama "intérprete PostScript". También se le denomina RIP por las

siglas inglesas de *Raster Image Processor*, lo que traducido es algo así como "Procesador de peinado o rastrillado de imagen".

De hecho, el lenguaje PostScript es tan "universal" que su forma más sencilla son simples instrucciones escritas como textos (siguiendo, eso sí, una sintaxis muy rigurosa). Básicamente un fichero PostScript contiene instrucciones que, traducidas al lenguaje humano, dicen cosas del tipo "traza un círculo de 3 cm. de radio, dibuja una línea de 4,5 cm. en un ángulo de 45 grados, etc..." Por eso es portable, porque depende del RIP o interprete PostScript comunicarle a la máquina en su código particular cómo tomar esas instrucciones básicas y ejecutarlas una detrás de otra hasta imprimir la página.

Hasta no hace mucho un RIP profesional de preimpresión solía ser un aparato carísimo con un *software* muy especializado cuyo único trabajo era ir *tragándose* los ficheros PostScript para enviar los resultados interpretados como secuencias de bits ("rastrillados", del inglés *rasterized*) al dispositivo de impresión (una impresora, filmadora, etc...). Hoy día hay excelentes RIP *hardware* y *software* capaces de dar paso a ficheros realmente grandes.

Porque esa es una de las ventajas - desventajas de PostScript: Al ser "interpretado" (para ser razonablemente universal), no puede ser muy condensado, con lo que los ficheros son "grandes" de por sí. Y, lo que es más, su interpretación puede ser una tarea titánica (sin exagerar) capaz de bloquear al RIP más aguerrido y preparado. No es extraño que ficheros de 8 Mb., al ser enviados a una cola de impresora, generen archivos de 40 Mb. o más.

Además, pese a ser un lenguaje básicamente "vectorial" (es decir, que realiza descripciones de operaciones matemáticas para que se ejecuten), puede incrustar "dentro" de sus ficheros objetos de mapas de bits (es decir, imágenes formadas por series ordenadas de bits muy grandes). Eso quiere decir que un fichero con pocos Kbs de instrucciones pueden llevar incrustados ficheros de megas y megas de tamaño.

Y todo eso se lo tiene que tragar y traducir el RIP (¡y hay quien espera que los dispositivos nunca se bloqueen!).

### 3. Los niveles de PostScript

A lo largo de los años y debido a su éxito, PostScript ha ido sufriendo adaptaciones para mantenerse al día. Hasta hoy (2001), esas adaptaciones se han conocido como "niveles" (aunque adobe parece haber dejado de lado esta nomenclatura a partir del nivel 3). Así, el primer PostScript se conoció posteriormente como "Nivel 1". En 1994 Adobe lanzó el "Nivel 2" del PostScript, y en 1998 lanzó el Nivel 3. Cada uno de ellos tiene unas características principales, que són:

1. **PostScript Nivel 1** Capacidad de trabajar con objetos vectoriales (descripciones matemáticas), con "fuentes de contorno" (un conjunto limitado de descripciones de letras a base de trazados matemáticos), con lineaturas de semitonos profesionales, capacidad de generar puntos de semitonos de distintas formas (línea, cuadrado, círculo, elipse, etc...), capacidad de gestionar hasta 256 tonos de gris distintos en una impresión. Independencia total del dispositivo (lo que permite imprimir aprovechando al máximo la resolución de éste). Portabilidad entre aparatos. Disponibilidad pública del código y de su sintaxis (lo que permite a cualquiera escribir un programa capaz de generar ficheros PostScript) y algunas cosas más.
2. **PostScript Nivel 2** Una gestión de memoria extremadamente mejorada. Capacidad de realizar la separación del color dentro del RIP (*in-RIP separation*). Capacidad de recibir imágenes comprimidas en determinados formatos. Soporte de fuentes asiáticas (más amplias en bytes que las europeas). Mejora de los controladores de impresora. Mejora muy notable (aunque algo tardía) de los algoritmos de tramado de semitonos. El nivel 2 de PostScript aun no ha sido adoptado del todo por muchos fabricantes de *software* y *hardware*.
3. **PostScript Nivel 3** El uso de 12 bits para describir las lineaturas de semitono, lo que permite superar la barrera de 256 tonos y llegar hasta 4.096 tonos de un mismo color. Un control extremadamente mejorado de las separaciones de color que permite realizar sin problemas separaciones de color que superen la cuatricromía convencional (hexacromía, duotonos, imágenes multicanal, etc...). Soporte del formato PDF desarrollado para el programa Acrobat. Una gestión del color mucho más avanzada. Y, por último, ciertas características de impresión relacionadas con Internet que no han sido muy desarrolladas. El hecho es que muchos RIPs aún distan de ser capaces de gestionar una mínima parte de las especificaciones de este Nivel 3. Lo mismo ocurre con programas de tratamiento de imagen (¿de qué vale enviar a un dispositivo que reconoce más de 256 tonos por canal una imagen RGB o CMYK cuya descripción de canales se limita a 8 bits por canal?).

Si envías un fichero PostScript con instrucciones de un nivel a una máquina o programa con un RIP de un nivel inferior, los resultados serán o impredecibles o nulos. Es por eso muy importante que te asegures que tu impresor tiene los dispositivos capaces de procesar aquello que pides. Habla con él y consultale si hay algún problema con los ficheros EPS. En caso de duda, atente a niveles inferiores o envíale una prueba.

### 4. Los ficheros PostScript

Muchas personas se hacen un lío con los ficheros PostScript, sin embargo la cosa no es, en lo básico, muy complicada. Son ficheros que se atienen a la estructura del lenguaje PostScript para, usualmente, describir datos de imagen a un dispositivo de impresión "de calidad".

Todo fichero PostScript tendrá un encabezamiento que lo identificará como fichero PostScript: `!PS-Adobe-3.0` y terminará con una marca de final de fichero del tipo `%EOF`, `%EPS grestore`, `%EndPhotoshop`, o algo similar. No lo dudes, cualquier fichero que

abras con un editor de texto y veas que contiene ambas cosas es un fichero PostScript (aunque las apariencias puedan decir otra cosa).

Los ficheros Postscript pueden ser multipágina o de una sólo página. Éstos últimos se suelen llamar ficheros "EPS" (pronúncialo "e-pe-ése"), del inglés *Encapsulated PostScript*, ("PostScript Encapsulado").

La diferencia básica es que los ficheros EPS son ficheros que se van a tratar como una única imagen (especialmente en otro programa) y que, por ello, contienen una delimitación de tamaño (que va marcada mediante el uso de la orden "BoundingBox" y la restricción de no usar la de "showpage").

la diferencia entre los ficheros PostScript a secas y los ficheros EPS es, pues, que son los primeros no encapsulados y pueden ser multipágina (y no los debes llamar, en buena ley, "EPS"). Suelen tener la extensión "ps". Hecha esta salvedad, por comodidad uso aquí casi siempre el término EPS

Los ficheros PostScript (del tipo que sean) pueden contener, además de masas de texto, información gráfica de dos tipos principales: Objetos vectoriales y mapas de imágenes. En claro: Dibujos de programas como Illustrator y fotos de programas como Photoshop. La complicación surge de que un dibujo vectorial puede contener a su vez uno o más "mapas de imágenes" (fotos) incrustadas en su interior, y un mapa de imágenes puede contener uno o más objetos vectoriales (trazados, en este caso). Es un poco como esas muñecas rusas que se contienen unas dentro de otras.

La clave que te ayudará a distinguir unos de otros es la siguiente. Si intentas abrir un dibujo vectorial EPS (originario de Illustrator, Freehand o similares) en un programa de imagen (como Photoshop), éste te pedirá que le digas una resolución y un modo de color para abrirlo. (Ten cuidado, una vez que lo hagas, si lo guardas así, habrás perdido la información vectorial y dado una resolución permanente al fichero).

Si, por el contrario intentas abrir una foto EPS (mapa de imágenes) en un programa vectorial como Illustrator, lo que verás será la previsualización de dicho fichero y no verás trazados, sino sólo una caja cruzada por un aspa. No habrá "formas" que editar, sólo la caja que enmarca el fichero.

Si has llegado hasta aquí, entenderás que aunque para crear ficheros PostScript podría usar un simple editor de textos, hoy tendrías que ser una persona muy peculiar para hacerlo así

Al ser un estándar público (aunque bajo el copyright de una empresa particular), hay muchos programas capaces de manipular ficheros PostScript y su enfoque varía. Los hay gratuitos, los hay para Linux, Macintosh, Windows y otras plataformas (Solaris, etc...)

## Opciones al guardar un fichero PostScript

Éstas son las opciones que pueden existir al guardar un fichero PostScript o EPS. No todas existen en todos los programas (la mayoría sólo tienen unas pocas) y los ficheros PostScript que no son EPS sólo disponen de alguna de ellas (no hay visualización, por ejemplo). La mayoría sólo tienen unas pocas.

1. *Previsualizar*: Aquí lo que se pregunta es "¿Qué visualización quieres?". Las opciones son básicamente tres: (a) Ninguna, (b) 1 Bit (Blanco y negro) (c) 8 Bit (color), pero con ciertas variantes.

Un fichero EPS puede llevar "incrustado" un pequeño fichero de cabecera de muy baja resolución y calidad para poder trabajar con el fichero en un programa sin que éste tenga que interpretar el código PostScript.

Así, si deseas situar un EPS dentro de otro programa, debes decirle que le dé una visualización. Si no lo haces, sólo verás una caja grisacea con un cartelito en medio que dice "PostScript image" o "Imagen PostScript" y el nombre del fichero (caso de Quark XPress con cualquier archivo EPS); o simplemente una caja aparentemente vacía (caso de Illustrator o Freehand con una fotografía EPS).

Si eliges una visualización, ten en cuenta que ésta será siempre muy mala (comparada con la de un TIFF). Lo que estarás viendo en QuarkXPress o Illustrator será una pequeña representación (previsualización del fichero), no es el fichero en sí. Es de muy baja resolución (72 ppp).

No todos los programas están obligados a trabajar con esa previsualización. En Adobe Illustrator e InDesign se puede elegir si queremos que el programa interprete ("rasterizar": *To rasterize*) el código PostScript (mejor visualización, peor rendimiento) o no. En Photoshop no hay opción ya que el código se rasteriza siempre (salvo en los contados casos de formas vectoriales y de trazados de recorte, como veremos más adelante). En Quark XPress sólo se puede trabajar con esa visualización, salvo que se adquiera una Xtensión como Enhance Preview XT.



(a) Fichero EPS sin previsualización.



(b) Fichero EPS con previsualización de 1 Bit (imagen de mapa de bits tramada).



(c) Fichero EPS con previsualización de 1 Bit (imagen de mapa de bits tramada).



(d) Fichero EPS interpretado ("real" RGB, 24 bits).

Las opciones de formato de esa previsualización (que no del archivo en si) son varias:

- Ninguna*: No habrá representación (pero el fichero estará perfecto, por lo demás).
- Tiff*: La "cabecera" será un pequeño fichero en formato Tiff. Puede ser de imagen de línea (1 bit / Pixel: bitmap de blanco y negro) o de color indexado (8 bits / Pixel: 256 tonos de color). La resolución "por omisión" es 72 ppp. Pero hay programas (como los de Corel) que ofrecen la posibilidad de aumentarla para mejorar algo la visualización (aunque no es una buena idea).
- Pict*: Denominada a veces "Macintosh", esta opción es muy similar a la anterior salvo por el hecho de ser un formato Pict que se almacena aparte. También puede ser de imagen de línea (1 bit / Pixel: bitmap de blanco y negro) o de color indexado (8 bits / Pixel: 256 tonos de color). La resolución es de 72 ppp. Es la opción "por omisión" de Macintosh.
- Jpeg*: La "cabecera" será un pequeño fichero de color en formato jpeg de 72 ppp. Es la opción que mejor previsualización ofrece pero la menos compatible (sólo vale para Macintosh, necesita tener instalado Quick Time y funciona sólo a partir del nivel PostScript 2).
- WMF*: La "cabecera" será un pequeño fichero de color en formato Windows Meta File de 72 ppp. Es una opción existente en programas de Windows (como Corel). No es muy recomendable. CorelDraw! permite elegir esta opción.
- EPSI*: Supuestamente Independiente de la plataforma (Epsi: *Encapsulated PostScript Interchange*), este formato se limita casi exclusivamente a entornos Unix. La previsualización se incrusta como líneas de comentarios PostScript en datos ASCII de 7-bits. En este caso el archivo EPS sólo puede guardarse como ASCII.

La opción que más compatibilidad ofrece es TIFF (vale para Windows, Macintosh y Unix). En los otros casos, la forma que tienen los sistemas Macintosh y Windows de guardar ciertos formatos hace que muchas veces la previsualización se pierda al

pasar documentos EPS de una plataforma a otra.

Para "recuperarla", basta con abrir el archivo en un programa como Photoshop y volver a guardarla.

2. *Codificación*: El archivo se puede codificar de tres maneras principales:

○ *ASCII*: En este caso, los datos van como código ASCII (es decir: texto del más básico). Es la opción con la que los ficheros ocupan más sitio ('bastante', por cierto), pero también es la opción más compatible y menos problemática.

○ *Binaria*: Aquí, los datos se codifican en forma binaria. Ocupan mucho menos que en el caso anterior pero si se envía a imprimir a máquinas antiguas pudiera haber problemas (bastaría en este caso con reabrir el fichero y volverlo a guardar con codificación ASCII).

○ *JPEG*: En esta tercera vía, los datos se comprimen usando el algoritmo JPEG, que es un sistema de compresión "con pérdidas" (*lossy*). Requiere al menos PostScript Nivel 2.

3. *Incluir trama de semitonos*: Lo que hace es "incrustar" una trama de semitonos PostScript concreta (incluidos todos sus parámetros: Lineatura, ángulo de trama, función del punto,...).

Si se marca (y manipula) esta opción, se dejan esos valores "fijos", por lo que salvo que sepas lo qué estás haciendo, mejor deja esta opción sin marcar. Para aquellos con experiencia, se pueden obtener algunos efectos interesantes y resolver algunos problemas, pero es una opción que sólo los impresores deberían tocar (quedas avisado). No sigas a las personas que tienen la manía de marcar esta opción y la siguiente "por si acaso".

4. *Incluir función de transferencia*: Esto permite incrustar "curvas de transferencia" de tinta dentro de un archivo (que el 10% de tinta cian se imprima como 8%, por ejemplo). De forma similar a la anterior, esta es una opción básicamente reservada a los impresores.

5. *Gestión de color PostScript*: Esta opción es técnicamente un poco compleja, ya que implica cierto conocimiento de la Gestión del color PostScript. Una vez más. Si no entiendes verdaderamente esta posibilidad, no la marques, ya que pueden producirse resultados inesperados.

6. *Blancos transparentes*: Esta opción sólo aparece disponible con ficheros de mapa de bits. Si la marcas, al colocar el fichero en otro programa, lo que es blanco se mostrará e imprimirá como transparente.

7. *Incluir datos vectoriales*: En Photoshop, esta opción sólo aparece activa si intentas guardar un fichero EPS conservando capas de texto o las formas vectoriales como tales (sin interpretar (*rasterize*)).

8. *Incrustar fuentes*: Esta opción sólo aparece en algunos programas (como Adobe Illustrator o CorelDraw!). En otros (como Quark XPress) no está directamente disponible (aunque sea técnicamente posible por medio de alguna extensión especializada o imprimiendo a un fichero (*print to file*)).

En cualquier caso es muy necesario incrustar siempre las fuentes. Si no se pudiera, será necesario enviar una copia de ellas junto con el fichero PostScript.

Ten en cuenta que si el fabricante de la tipografía la creó (a veces por error) con restricciones a la incrustación no podrás incrustarlas. La solución, si no es muy largo es convertir el texto en trazados (opción que algunos programas como Corel te permiten elegir al exportar como EPS).

Si el EPS que guardas tiene también trazados y lo guardas con esa opción, debes tener en cuenta que (al menos hasta la versión 7.01) Photoshop no respetará esos vectores al reabrir el archivo: Perderás los trazados de recorte. Así que, si quieres conservar los trazados, no marques esa opción (pero en ese caso, Photoshop interpretará los textos y formas vectoriales, y conservará los trazados).

9. *Interpolación de la imagen*: Esta opción lo que hace es activar que el RIP interpole la imagen al filmar. El algoritmo de interpolación (y su calidad) depende del RIP que la haga y no va especificada dentro del fichero.

Por si no lo has entendido: Al filmar, la máquina se inventará los píxeles que faltan hasta que el archivo tenga la resolución necesaria usando mecanismos y códigos que su fabricante le ha proporcionado. Eso puede ser bueno, si la máquina está bien diseñada en ese sentido y si no tiene que inventarse demasiados datos. Si no es así, entonces es una mala opción. No interesa activarla con imágenes de alta resolución.

10. *Incrustar los ficheros (de mapa de bits) colocados* Esta opción puede tener varias formas de expresarse, pero sólo se presenta en ficheros vectoriales o de texto que llevan imágenes de mapa de bits. Este tema tiene una explicación tan compleja que, si quieres saberla mejor te lees el mensaje que Kay Anderson envió en enero de 2001 al foro que Adobe tiene sobre el programa Illustrator (versión Macintosh). Merece la pena. Te adelanto, de todas maneras, que en Illustrator la opción estándar es

la de no incluirlos, sino "enlazarlos", y adjuntárselos al impresor en una carpeta cuando envíes el archivo para que el dispositivo de impresión pueda localizarlos al ser invocados por el "archivo madre." De hecho, si no lo haces así en algunos casos como los duotonos o las imágenes multicanal tendrás graves problemas.

En otros programas, es mejor siempre incrustarlos, para que formen parte de la pieza general. Aun no habiendo una norma general, el sentido común dice que cuanto más complejo sea el resultado final (¿recuerdas las muñecas rusas que van una dentro de otra?) más fácil es que se corrompa.

## PostScript, EPS, PDF e Illustrator

Algo que confunde a las gente es la diferencia y parentesco que existe entre los ficheros de PostScript, EPS, PDF e Illustrator. Aunque el tema es complicado, se puede intentar resumir de la siguiente manera:

1. *Un fichero PostScript* Es un conjunto de código PostScript que describe generalmente un contenido destinado a la impresión. Puede (y suele) tener más de una página y necesita imprimirse en un dispositivo capaz de interpretar ese código PostScript (es decir: Un RIP).
2. *Un fichero EPS* (PostScript Encapsulado) Es un tipo concreto de fichero PostScript. Es monopágina. Está obligado a contener unos parámetros PostScript determinados que describen su tamaño y no puede contener algunas ordenes del PostScript más general. También necesita un RIP para su interpretación.
3. *Un Fichero PDF* Es, simplificando mucho, el "hijo listo" de PostScript. Para saber más sobre qué es y cómo funciona, mejor consulta estas páginas [formato PDF](#).
4. *Un fichero de Illustrator* Esto tiene su truco, ya que la relación entre Illustrator y los dos formatos anteriores es extremadamente fina y ha ido variando con el tiempo. La mejor forma de decirlo es reproducir el mensaje de Paul Asente (uno de los ingenieros de Adobe Illustrator) en el grupo de noticias `alt.graphics.illustrator` (28 de diciembre de 2002) respondiendo a la pregunta "¿Cuál es la diferencia entre los ficheros \*.AI y los \*.EPS?":

"[el formato de los archivos de Illustrator] Ha cambiado con el tiempo. En este mensaje doy por hecho que cuando escribo 'fichero EPS' me estoy refiriendo a 'EPSs generados por Illustrator'. Otros programas pueden escribir ficheros EPS que no son en absoluto como los ficheros \*.AI. También describiré cómo encaja aquí el formato PDF:

### Hasta Illustrator 6

La parte de contenido de un fichero \*.ai y un \*.eps son idénticas. Los ficheros EPS incluyen definiciones de procedimiento PostScript (llamadas **procsets**) que permiten que una impresora PostScript pueda representar el contenido. Los ficheros EPS pueden incluir opcionalmente una previsualización prefabricada (*pre-rendered*) de modo que los programas que no son capaces de interpretar el código PostScript puedan mostrar una versión de baja resolución de la imagen como referencia. Los ficheros \*.ai dejan fuera los **procsets**, por lo que son menores.

### Illustrator 7 y 8

En la práctica no había diferencias. Los ficheros \*.ai contenían siempre los **procsets** PostScript. El espacio de almacenamiento en los discos es ya el suficiente para que no merezca la pena hacer que la gente se preocupe de en qué formato deben guardar por dejar fuera o no los **procsets**. La única diferencia real es que los ficheros EPS aun contienen las previsualizaciones. Además, Illustrator puede escribir ficheros PDF, pero el viaje contrario no es real: Si reabres el fichero PDF en Illustrator se pierde la estructura del documento, aunque el resultado tenga la misma apariencia.

### Illustrator 9

Los ficheros \*.eps y \*.ai son completamente diferentes. Los \*.ai se basan ahora en el formato PDF y Acrobat ya puede abrir ficheros \*.ai. Los ficheros \*.eps contienen ahora dos partes: Una es una versión reproducible en PostScript de la ilustración. La otra es una versión comprimida del antiguo formato \*.ai. cuando se vuelve a abrir ese fichero, Illustrator ignora la parte PostScript y sólo usa la parte \*.ai. La impresión usa la parte PostScript e ignora la parte \*.ai. [Por su parte] Todos los ficheros \*.ai contienen una representación (*rendition*) PDF de la ilustración en vez de una versión PostScript y además incluyen el antiguo formato \*.ai comprimido. Cuando se vuelve a abrir el fichero, Illustrator ignora la parte PDF y Acrobat ignora la parte \*.ai. Un fichero es básicamente idéntico a uno \*.ai pero se tiene la opción de dejar fuera la sección \*.ai comprimida para hacer que el fichero sea menor con vistas a su distribución por red.

### Illustrator 10

Muy similar a Illustrator 9, pero con la opción añadida de que al guardar un fichero \*.ai se puede omitir la representación PDF de la ilustración. Sigue siendo un fichero PDF, pero si se abre en Acrobat, lo que se ve es un mensaje diciendo que el fichero se guardó sin contenido PDF.

Obviamente, P. Asente no describe lo que puede ir 'dentro' de un fichero, sino sólo su estructura en un sentido general.

## 6. Fuentes y agradecimientos

La mayoría de la información presente en esta página proviene de las siguientes páginas *web* que recomiendo muy encarecidamente (los pongo por orden de lectura recomendada). Desgraciadamente están en inglés:

- [A First Guide to PostScript](#). Amplía esta pequeña base general.
- [The Prepress Panic Page](#). Un sitio realmente bueno y lleno de cosas útiles. Si trabajas en contacto con el PostScript, deberías visitarlo. Tiene un apartado sobre PDF muy interesante (que, por cierto, está casi todo en español en este mismo sitio).
- [La página de Adobe sobre PostScript](#). El hogar de la Fuerza, Luke. Curiosamente no contiene mucha información.

Y si quieres ya saber *mucho más*, están los originales. Los libros de referencia (vaya, también en inglés). Los puedes encontrar, por ejemplo, como PDFs en el sitio web del laboratorio Fermi (aunque están en más sitios por la red):

- [Especificaciones oficiales del PostScript nivel 2 \(El Libro Rojo\)](#) Unos 4Mb en PDF.
- [Especificaciones oficiales del PostScript nivel 3 \(El nuevo Libro Rojo\)](#) Unos 8Mb en PDF.
- [Especificaciones oficiales de las Fuentes PostScript Tipo 1 \(El Libro Negro\)](#) Unos 455 Kb en PDF.
- [Libro de aprendizaje y ejemplos del lenguaje PostScript \(El Libro Verde\)](#) Unos 868Kb en PDF.

Otro libro muy interesante (también PDF, 846 Kb) es *Thinking in PostScript*, de Glenn C. Reid. [Está disponible](#) en su sitio *web*.



© Gustavo Sánchez Muñoz, 2003