

Magic Point: Presentaciones a la carta

Alvaro del Castillo San Félix

acs@barrapunto.com

1. Introducción

MagicPoint es una herramienta de presentaciones para las X11. Es una herramienta que se basa en la idea de tener unos ficheros de textos relativamente sencillos en los que se indican los contenidos y aspecto de la presentación. Dichos ficheros luego son interpretados por MagicPoint el cual los muestra ya como si fuera una presentación.

La creación de presentaciones con MagicPoint es muy rápida, en especial cuando se conoce la sintaxis del lenguaje de MagicPoint, y permite que podamos trabajar sobre ella desde cualquier editor de texto. Podríamos decir que sigue una filosofía muy similar a LaTeX.

Para abrir boca veamos la entrada en una presentación de GNOME Hispano y Gunih que se preparó para la Guadec y que es una de las más lindas que he podido ver.

Figura 1. MagicPoint en acción: Presentación de GH y Gunih para la Guadec



2. Instalación

La instalación es muy sencilla si tenemos una distribución con paquete de MagicPoint. En el caso de Debian 3.0 la versión que tenemos de MagicPoint es:

```
acs@linex:~/devel/web-xml/articulos/magicpoint$ dpkg -l | grep mgp
ii  mgp                1.09a-8           MagicPoint- an X11 based presentation tool
```

Las opciones de las que disponemos al ejecutar MagicPoint desde la línea de comandos son las siguientes.

```
acs@linex:~/devel/web-xml/articulos/magicpoint$ mgp
Usage: mgp [opts] mgpfile
-b <color>: Specify background color
-c <vfcap>: Specify vfontcap file
-d: Demo mode - go through the presentation
-f <font>: Specify default vfont font
-g <geometry>: Set window geometry
-h Display this help message
-l: Do not use outline font on vfont
```

- n: Disables control key input from tty
- o: Do not override the window manager
- p <page>: Start at the specified page
- q Do not beep on errors
- t <timeslot>: Enable presentation timer
- v: Show version number and quit
- w <dir>: Specify a working directory
- x <engine>: Disable specified rendering engine
- B: Ignore background image
- C: Use private colormap
- D <dir>: Generate html pages for the presentation
- F<mode>,<effect>,<value>: Use forwarding caches
- G: Page guide is on
- O: Obey to the window manager
- Q <quality>: Set background image quality(0-100)
- R: Do not perform automatic reload
- S: Do not process directives that forks process (default)
- U: Do process directives that forks process (unsecure mode)
- T <timestampfile>: Update timestampfile on page refresh
- V: Be verbose
- X <gsdevice>: ghostscript device to use

Vemos que podemos hacer cosas realmente interesantes, como añadir temporizadores para que las transparencias avancen de forma automática, transformas las presentaciones a formatos como HTML, probar una presentación para observar errores (-d) o ver de forma automática como se actualiza una presentación según modificamos el fichero con la presentación.

3. La presentación mínima

Para poder ver la sencillez que puede llegar a tener MagicPoint veamos lo que puede ser una presentación mínima en la que tan sólo mostramos el titular de la presentación sobre MagicPoint.

```
%page
Presentación de Magic Point
```

Figura 2. Presentación mínima MagicPoint

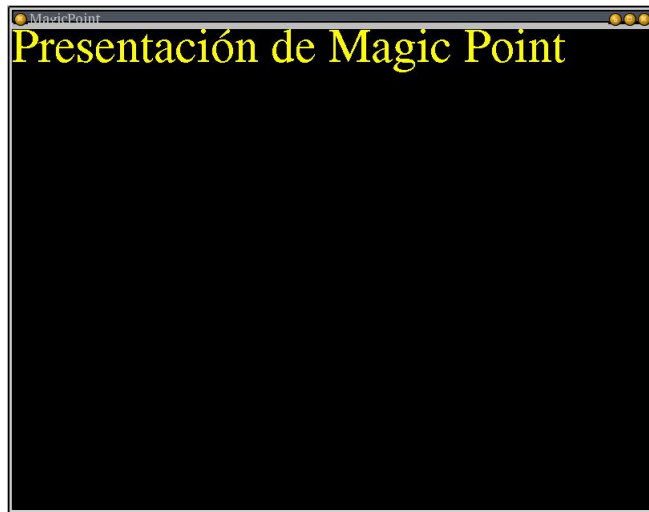


Como podemos deducir el cambio de página entre una transparencia y otra se realiza mediante la orden "page" de MagicPoint. Todas las ordenes de MagicPoint comienzan con un "%" y podemos poner comentarios utilizando el "#". Pero ya podemos ver un primer problema a la hora de visualizar esta presentación: no aparecen el caracter acentuado "ó". Esto es debido a que la fuente por defecto que está usando MagicPoint no tiene soporte para el alfabeto "iso8859-1" que es el incluye entre otras cosas los caracteres acentuado. Para poderlo resolver basta con indicar qur utilice este alfabeto dentro del tipo de letra que estamos utilizando para lo cual utilizamos el comando "%font" de MagicPoint para especificar las propiedades de la fuente.

```
%page
%font "standard", charset "iso8859-1"
Presentación de Magic Point
```

Una vez realizado este pequeño cambio ya tenemos nuestros queridos acentos y eñes disponibles para ser visualizados.

Figura 3. Presentación mínima MagicPoint con soporte de español



4. Formato de las páginas de MagicPoint

Es habitual dentro de las presentaciones que los contenidos se organicen por puntos, que a su vez se dividen en otros puntos. Para este tipo de estructuras de información lo ideal es disponer dentro del programa de presentaciones de niveles en los que poder ir estructurando los datos.

MagicPoint puede simular este modelo de niveles. Para diferenciar un nivel de otro podemos utilizar los tabuladores. Si una nueva línea se encuentra sangrada con un tabulador con respecto a la línea anterior, quiere decir que es un subpunto del punto sobre el que está sangrado (indentado). Es importante resaltar que se necesitan tabuladores de verdad, no espacios en blanco, para que MagicPoint reconozca la estructura del documento. Por ejemplo, dentro de Emacs lo podemos forzar con una línea inicial de la forma:

```
#/* -*- tab-width: 8; indent-tabs-mode: t -*- */
```

Pero esto es sólo una posibilidad. Otra posibilidad bastante interesante es la de poder incluir en cada una de las líneas de una presentación un comando por defecto. Por ejemplo, si queremos que todas las transparencias tengan un fondo de una imagen,

basta con que le digamos a MagicPoint que incluya dicho comando de fondo (bgimage) en la primera línea de todas las transparencias.

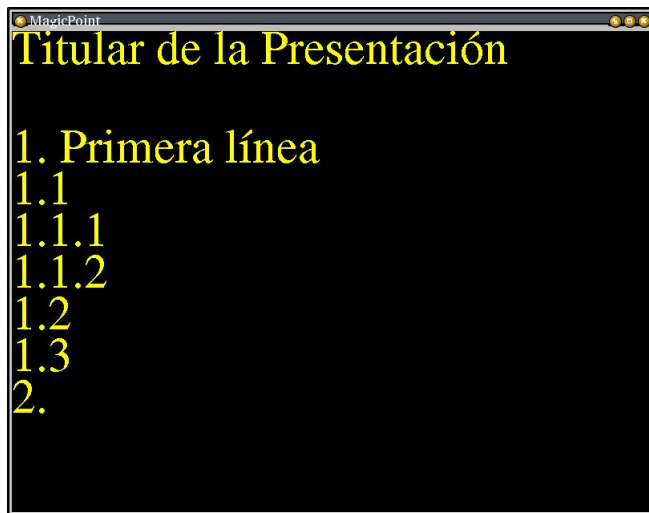
```
%default 1 bimage "fondo.jpg"
```

Hay ciertos comandos que sólo son validos en la primera línea de una transparencia, por lo que será habitual que dentro de "%default 1" encontremos muchos comandos.

```
%default 1 bimage "fondo.jpg",fore "green4"
```

Pero nada mejor que un ejemplo para entenderlo todo.

Figura 4. Organización de una página en MagicPoint



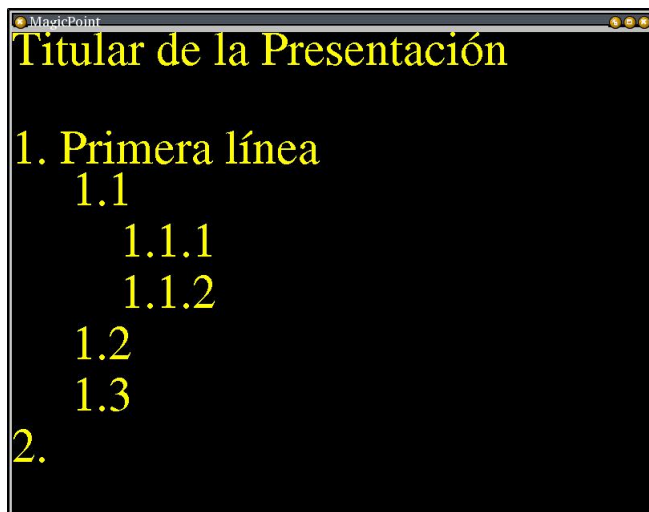
Pues no parece que nos haya hecho mucho caso MagicPoint a la hora de presentar organizada la información en la página. Esto es debido a que por defecto, no se aplican reglas de organización visual de la información. Pero podemos introducir nosotros comandos para que sea así. Los comandos de control de la visualización de la información ya no se hacen por página, si no de forma global, antes de definir cualquier página. Y se aplican por defecto a todas las páginas por igual. Se podría decir que conceptualmente son similares a las hojas de estilo CSS de las páginas HTML.

```
/* -*- tab-width: 4; indent-tabs-mode: t -*- */
```

```
%tab 1 vgap 35, prefix "      "  
%tab 2 vgap 35, prefix "          "  
%tab 3 vgap 35, prefix "              "  
%page  
%font "standard", charset "iso8859-1"  
Titular de la Presentación  
  
1. Primera línea  
1.1  
1.1.1  
1.1.2  
1.2  
1.3  
2.
```

Después de introducir definiciones para 1,2 y 3 tabuladores, y especificar que se debe de poner antes de cada tabulador, el resultado que pasamos a obtener es el siguiente.

Figura 5. Organización tras definir los tabuladores

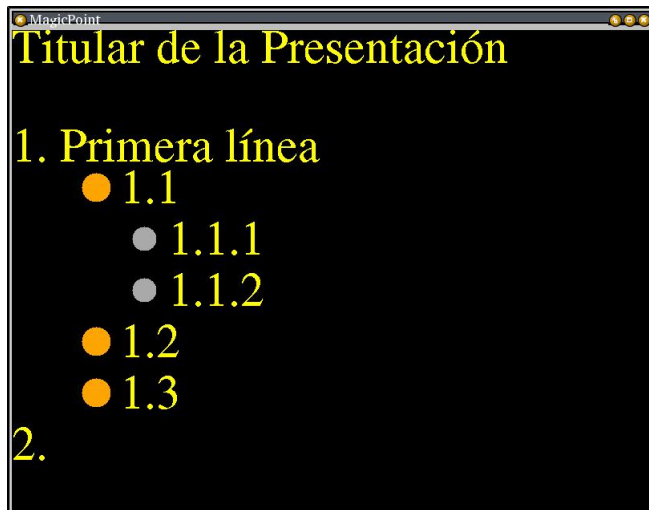


Pero sin duda, lo podemos hacer aún mejor. Antes de cada tabulación, no sólo ponemos espacios en blanco si no que también podemos poner diferentes iconos. Para ello basta con modificar las líneas de definición de tabulación de la siguiente forma:

```
%tab 1 vgap 35, prefix "      ", icon arc "orange" 60
%tab 2 vgap 35, prefix "      ", icon arc "darkgray" 50
%tab 3 vgap 35, prefix "      ", icon delta3 "gray" 40
```

con lo que logramos el siguiente aspecto, mucho más atractivo

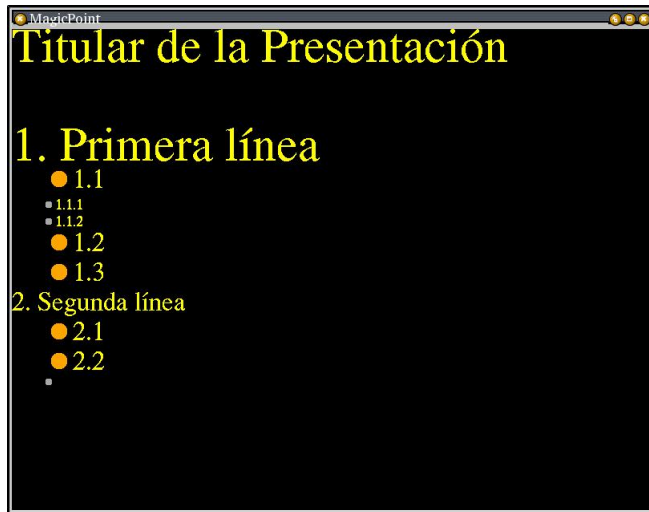
Figura 6. Organización con iconos en los tabuladores



Ya para terminar, lo que podemos echar en falta es que el tamaño de la letra no sea el mismo para todos los niveles de información. Esto lo logramos haciendo que se cambie el tamaño de la letra en función de que tabulación estamos utilizando.

```
%tab 1 size 6, vgap 35, prefix "      ", icon arc "orange" 60
%tab 2 size 4, vgap 35, prefix "      ", icon arc "darkgray" 50
%tab 3 size 3, vgap 35, prefix "      ", icon delta3 "gray" 40
```


Figura 7. Organización con iconos en los tabuladores con tamaños de letra



Hay cuidar que si ponemos un tipo de letra demasiado pequeño, los espacios en blanco pueden no ser suficientes para que el texto quede bien tabulado, como ocurre en este ejemplo.

El problema que tenemos ahora es que el texto sin tabular, se queda con el tamaño de letra del último texto tabulado que hicimos. Y por ejemplo, eso ha provocado que el punto "2" queda con letra que no es la adecuada. Para resolverlo indicamos un tamaño de letra por defecto que es el que se usará cuando no se diga lo contrario [FIXME]

5. Valores por defecto de configuración

Hemos visto que antes de comenzar a escribir las páginas, tenemos una parte de entrada donde se configuran parámetros que afectarán a todas las páginas que vayamos escribiendo en la presentación.

Es precisamente en esta zona inicial donde vamos a necesitar más esfuerzo para poner todo a nuestro gusto. Una vez logrado, podremos escribir las transparencias de la aplicación de una forma muy ágil utilizando esas definiciones. Hasta el momento, hemos utilizado los valores por defecto de MagicPoint para casi todos los posibles elementos de configuración, a excepción del uso de tabuladores.

Vamos ahora a ir viendo que podemos ir cambiando y hacer distintas combinaciones para conocer el juego que nos da MagicPoint. Estas combinaciones las vamos a poder ir guardando dentro de ficheros de configuración de MagicPoint y luego incluirlos con la orden "%include" que nos permiten incrustar el contenido de otro fichero dentro del actual.

En la presentación que llevamos hasta ahora vamos a crear un fichero llamado "estilo.mgp" que va a contener las reglas de estilo generales a toda nuestra presentación. Luego vamos a poder ir viendo como modificando este fichero, podemos tener una presentación muy diferente con los mismos datos. El contenido del fichero de estilo pasa a ser:

```
#!/* -*- tab-width: 4; indent-tabs-mode: t -*- */
%tab 1 size 6, vgap 35, prefix "      ", icon arc "orange" 60
%tab 2 size 5, vgap 35, prefix "      ", icon arc "darkgray" 50
%tab 3 size 2, vgap 35, prefix "      ", icon delta3 "gray" 40
```

mientras que nuestra presentación pasa a ser

```
#!/* -*- tab-width: 4; indent-tabs-mode: t -*- */
%include "estilo.mgp"
%page
%font "standard", charset "iso8859-1"
Titular de la Presentación

1. Primera línea
1.1
1.1.1
1.1.2
1.2
1.3

2. Segunda línea
2.1
2.2
```

Vamos a ir viendo los elementos más habituales comunes a toda la presentación que se definen en este fichero de estilo.

5.1. Fuentes

5.2. Comandos en cada línea de las páginas

5.3. Estilo de los tabuladores

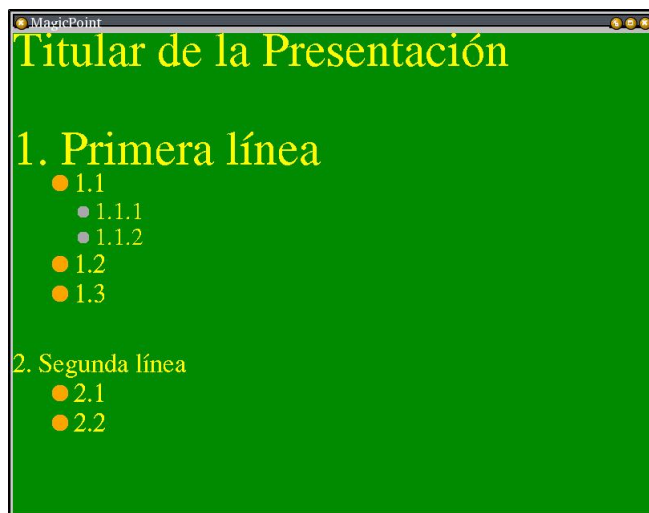
5.4. Fondo de las transparencias

En e fondo podemos poner diferentes motivos aunque siempre recordando que no deben de dificultar la lectura del contenido de las transparencias.

- Color sólido: sin duda es la apuesta más sencilla y con la que menos tiempo invertiremos. Basta con utilizar el comando "back"

```
%default 1 back "cyan"
```

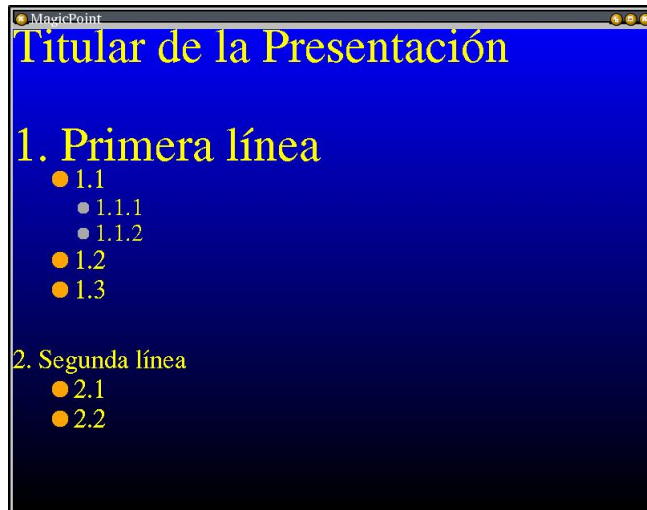
Figura 8. Color de fondo sólido en MagicPoint



- Gradientes de color: se utiliza el comando "bgrad" y permite crear gradientes. El más habitual es de azul a negro y basta con poner "%bgrad" al comienzo de la página

para obtenerlo. Si queremos que nos aparezca como fondo de todas las páginas, basta con añadirlo en la primera línea antes de que comiencen las páginas de la presentación con "%default 1 bgrad".

Figura 9. Gradiente por defecto en MagicPoint

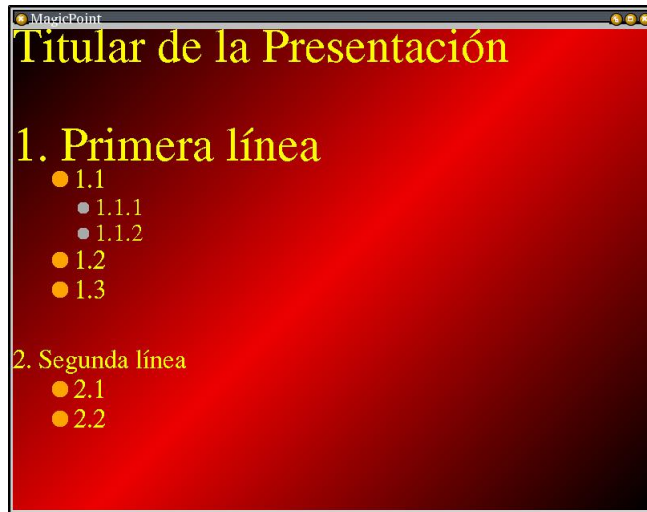


Vemos que el introducir el gradiente por defecto nos proporciona una sensación mucho más estática que el frío color negro, pero debemos de utilizar con cuidado los gradientes y no caer en fondos demasiado llamativos que hagan perder fuerza a los contenidos.

Aunque un poco de juego con las posibilidades de gradientes nos pueden llevar a diseños tan interesantes como el siguiente.

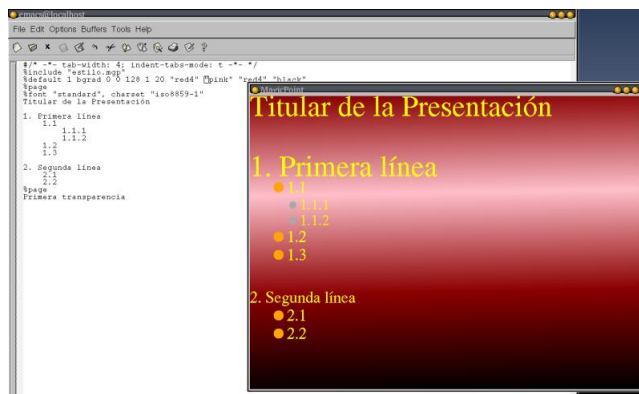
```
%default 1 bgrad 25 25 128 45 1 "black" "red2" "black"
```

Figura 10. Gradiente con rotación a dos colores



La mejor forma de crear el gradiente que más nos agrade es abrir la presentación y un editor con la definición del gradiente, y a la vez que vamos modificando los parámetros del gradiente, se irá actualizando de forma periódica la imagen de MagicPoint, lo que nos permite un diseño del gradiente bastante interactivo.

Figura 11. Edición de gradientes

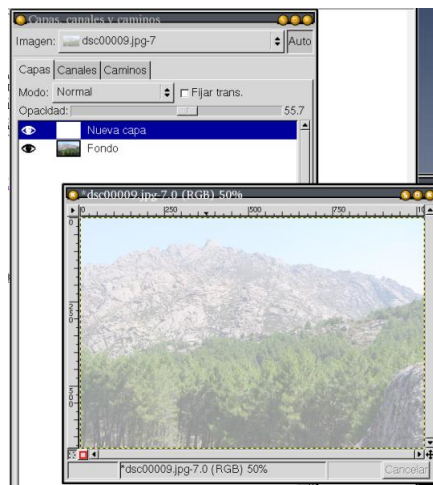


Dentro de la documentación de Magic Point viene un ejemplo muy completo de

como se definen los gradientes y sus resultados, dentro de un fichero llamado "gradation.mgp". Recomendamos al lector interesado que continúe en él la exploración.

- **Imágenes:** esta opción suele ser muy habitual ya que permite por ejemplo incluir de fondo el logo de la compañía que hace la presentación, capturas de la imagen del producto que se presenta o muchas otras cosas. De nuevo, tenemos que ser muy cautelosos de que la imagen de fondo no reste presencia a los contenidos. Una interesante forma de crear fondos difuminados es el abrir una imagen con Gimp, redimensionarla al tamaño que nos interese, luego añadirla una nueva capa blanca y jugar con la transparencia de esta capa para que deje ver más o menos dibujo de fondo. Una vez que lleguemos a la imagen que nos interesa, podemos unir todas las capas y guardarla para ser incluida como fondo.

Figura 12. Creación de una imagen de fondo



Podemos guardar la imagen a diferentes resoluciones para poderla utilizar con distintas visualizaciones de MagicPoint.

```
%default 1 bimage "fondo.jpg"
```

Figura 13. Presentación con imagen de fondo



Vemos que el resultado no es muy interesante pero basta con que cambiemos el color de las letras para obtener este otro resultado.

Figura 14. Presentación con imagen de fondo



6. Fuentes

Una de los puntos críticos de las transparencias es el tipo de fuente que utilizemos. La gran mayoría de la presentación visual van a ser mensajes de texto por lo que las fuentes que utilizemos van a ser críticas de cara a la estética y la comprensión de los contenidos de la presentación.

Una de las primeras bases a tener en cuenta es el utilizar una única tipografía o a lo sumo dos. Si ahora pensamos que tenemos que poner letras en diferentes tamaños, es sencillo llegar a la conclusión de que necesitamos tipografías que escalen bien, es decir, que cuando estemos con tamaños de letra pequeños se lean sin problemas, y cuando usemos tamaños grandes, no se vean demasiado pixeladas. Y un tipo de fuentes que cumple muy bien con este cometido son las fuentes True Type.

Una buena forma de comprobar si tenemos fuentes True Type en el sistema es utilizar el comando "xfonts" que nos muestra todas las fuentes disponibles en el sistema y filtrar aquellas del tipo True Type. De momento no tengo instaladas ninguna en mi sistema pero siguiendo las instrucciones del artículo en GNOME Hispano sobre fuentes True Type (<http://es.gnome.org/documentacion/articulos/truetype-gnome/truetype-gnome/book1.html>), será coser y cantar.

Aprovechando que ya hay algunas fuentes de tipo True Type empaquetadas en Debian, actualmente en la versión inestable, voy a aprovechar para instalar dichos paquetes, y así de paso comprobar que tengo de forma correcta instalada el servidor de fuentes de las X.

```
linex:/home/acs/art/fonts/truetype# apt-get install ttf-freefont
```

Este interesante paquete nos ha instalado fuentes True Type para las familias Mono (Monospaced), Sans y Serif, bajos los nombres FreeMono, FreeSans y FreeSerif. El uso de este tipo de letras veremos que nos va a bastar para nuestras presentaciones, y las fuentes van a quedar realmente bonitas.

Lo primero que tenemos que hacer es indicar en que directorio están las fuentes True Type que vamos a utilizar. Y para ello añadimos dentro de nuestro directorio de inicio el fichero ".mgprc", es decir, creamos el fichero "~/mgprc" en el que ponemos.


```
tfdir "/usr/share/fonts/truetype/freefont/"
```

para poder utilizar nuestras nuevas y estupendas fuentes True Type que acabamos de instalar. Para indicar que queremos utilizar este tipo de fuentes, tenemos que añadir la siguiente línea a la hoja de estilo que hemos estado utilizando hasta el momento.

```
%deffont "standard" tfont "FreeSerif.ttf"
```

Ahora vamos a ver la notable diferencia entre utilizar fuentes True Type, o utilizar la que coge por defecto del sistema en la presentación MagicPoint.

Figura 15. Presentación sin fuentes True Type



Figura 16. Presentación con fuentes True Type



Podemos definir tantas fuentes como queramos pero nunca hay que olvidar que el cambio en el tipo de fuente, despista a las personas observando la presentación, que ya habían asociado el contenido textual a un determinado tipo de fuente. Veamos lo extraño que puede llegar a resultar el utilizar el mezclar tres tipos de fuente.

```
#!/* -*- tab-width: 4; indent-tabs-mode: t -*- */
#include "estilo.mgp"
%default 1 bimage "fondo.jpg",fore "green4"
%deffont "standard" tfont "FreeSerif.ttf"
%deffont "standard1" tfont "FreeMono.ttf"
%deffont "standard2" tfont "FreeSans.ttf"
%page
%font "standard", charset "iso8859-1"
Titular de la Presentación (Serif)

%font "standard1", charset "iso8859-1"
1.Primer línea (Mono)
1.1
1.1.1
1.1.2
1.2 Ve
1.3

%font "standard2", charset "iso8859-1"
```

```
2. Segunda línea (Sans)
2.1
2.2
%page
Primera transparencia
```

Figura 17. Presentación con varias fuentes True Type



7. Imágenes

Ya hemos visto como meter imágenes de fondo, pero aún no hemos visto como podemos meter imágenes dentro del texto de la presentación. Y como vamos a ver, es realmente sencillo.

```
%page
Primera transparencia

%center
%size 4
La lagartija de la pedriza
%image "lagartija.jpg"
```

El resultado de esta transparencia lo podemos ver a continuación.

Figura 18. Imagen centrada y con titular



Pero podemos hacer más cosas a parte de poner la imagen directamente y centrarla en la transparencia. Podemos por ejemplo especificar las dimensiones del tamaño de transparencia para el que fue dimensionada la imagen. De esta forma, si una imagen se ve de forma correcta a 800x600, la veríamos pequeña si pasamos a trabajar a 1024x768. Pero si indicamos lo siguiente:

```
%image "lagartija.jpg" 800x600
```

estaremos indicando a MagicPoint que la imagen fue dimensionada para 800x600 y que si estamos por ejemplo a 1024x768, la escale para adaptarla a la nueva resolución.

A continuación mostramos dos capturas de la misma transparencia pero a 800x600 y a 1024x768. Mientras que la imagen de la lagartija mantiene la misma relación de tamaño con la transparencia, vemos que el fondo sin embargo ha dejado un espacio en blanco en la de 1024x768.

Figura 19. Transparencia a 800x600



Figura 20. Transparencia a 1024x768



Resolver el problema del fondo de la imagen es sencillo: basta con decir a MagicPoint que dicho fondo es para 800x600. El propio MagicPoint se encargará de adaptarlo si las resoluciones son diferentes.

Podemos limitar el número máximo de colores a utilizar para una imagen, o realizar un zoom de la misma. Veamos un ejemplo:


```
%image "lagartija.jpg" 2 800x600
```

Figura 21. Lagartija a 2 colores



Respecto a la animaciones, MagicPoint soporta GIF animados y MNG, las animaciones que utilizan PNG como formato base.

8. Transiciones

Para los usuarios de otros programas de presentaciones, el uso de transiciones en la visualización de los diferentes elementos de una transparencia, y en el paso de una transparencia a otro, es algo muy habitual. Las transiciones pueden dar ritmo a una ponencia y además, nos permiten ir mostrando la información de una transparencia de forma progresiva.

MagicPoint dispone de soporte para este tipo de transiciones utilizando diferentes efectos de cambio. Vamos a ver algunos de ellos.

La primera posibilidad es ir mostrando los elementos paso a paso, utilizando la tecla "SPC" (barra espaciadora) para ir mostrando cada uno de los elementos. Basta con poner el comando "%pause" entre los elementos que queremos ir visualizando de forma progresiva para lograrlo.

9. Otras características destacadas

9.1. Utilizando XML para construir las presentaciones

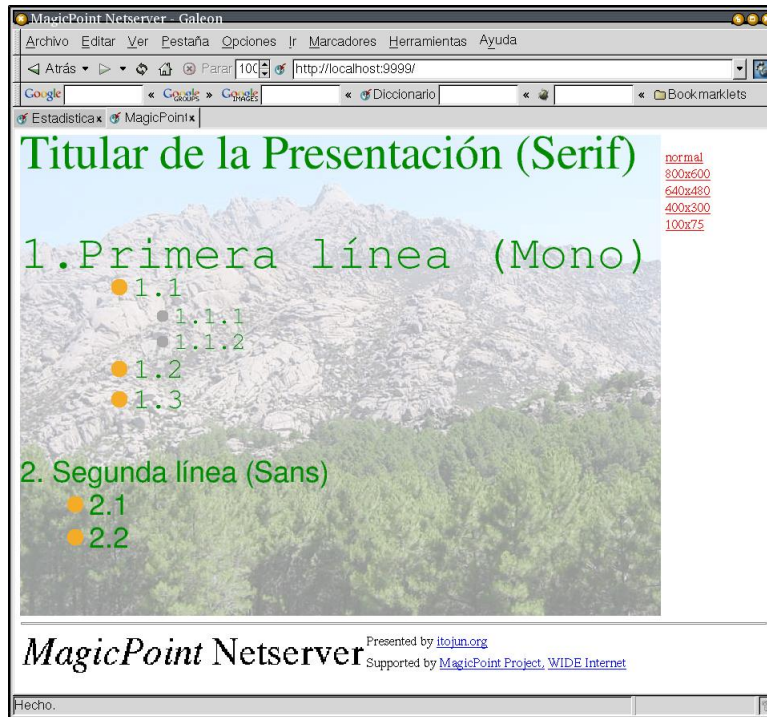
Existe un conversor de un formato XML de MagicPoint al lenguaje de MagicPoint por lo que podemos trabajar con dicho XML. Se utiliza una hoja XSL llamada `sdoc2mgp.xsl` (<http://xml.da-cha.org/>).

9.2. Presentaciones distribuidas

Existe el `mgpnet` que es precisamente para llevar a cabo este tipo de labores. Su uso es realmente sencillo ya que basta con que ejecutemos la presentación cambiando "mgp" por "mgpnet" y pasándole los mismos parámetros. Esto provocará que MagicPoint inicie un pequeño servidor de web en el puerto 9999 de la máquina donde se está llevando a cabo la presentación. Cualquier asistente a la presentación podrá acceder a ese servidor de web y podrá ir viendo mediante un navegador las transparencias según las va cambiando el ponente.

En realidad lo que se hace es realmente sencillo. Se captura de forma periódica la pantalla de MagicPoint utilizando `xwintoppm` y es la que se envía los clientes que se conectan al servidor de web de MagicPoint.

Figura 22. Interfaz web a una presentación



Para que todo funcione de forma correcta es necesario tener las herramientas "netpbm" instaladas, ya que las utiliza para generar la imagen a entregar al cliente web.

9.3. Pintar sobre las transparencias

Si en una transparencia pulsamos la tecla "x" pasaremos a un modo que nos permite trazar en color rojo líneas sobre la transparencia. Esto es realmente útil para resaltar aspectos de la presentación o incluso para realizar dibujos que permitan aclarar conceptos.

Figura 23. Pintando sobre una transparencia



9.4. Conversión a otros formatos

La herramienta MagicPoint dispone de conversores del formato de MagicPoint a HTML, LaTeX o PS, por lo que podemos adaptar nuestra presentación para su publicación web o impresión de forma sencilla y rápida.

10. ¿Qué le falta a MagicPoint?

- Editor gráfico, similar a como LyX lo es para LaTeX. En realidad si que existe algún intento en esta línea, como Ewipe (<http://www.jnet-jp.to/~sekisita/ewipe/>), basado en Tcl/Tk y que funciona tanto en GNU/Linux y otros Unix, como en Windows. Nos ha proporcionado esta información Victor Jose Valdés Márquez (victor.valdes@hispalinux.es).

Es un programa de presentaciones independiente de Magic Point pero es capaz de guardar la presentación como HTML o como mpg.

Figura 24. La herramienta Ewipe

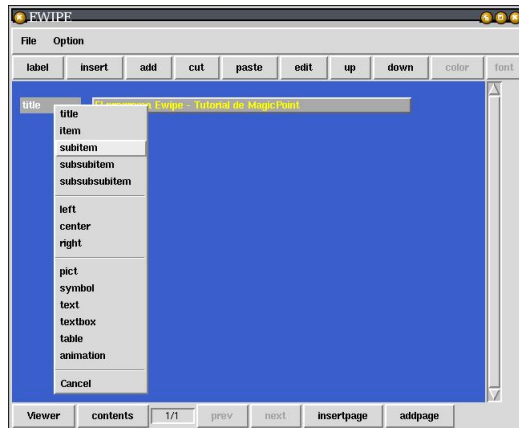
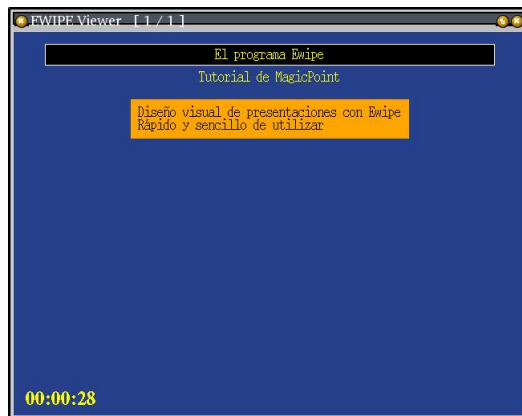


Figura 25. Visualizador de Ewipe



Tras una rápida prueba con él, es realmente sencillo de utilizar y muy rápido, aunque sin duda tiene ciertas limitaciones. Pero personas que lo han probado se han terminado decantando por seguir utilizando vi o emacs para crear sus presentaciones.

- Posibilidades de crear diagramas: para esto nos podemos apoyar en programas especializados en diagramas como Dia, desde los cuales podemos exportar la imagen

en formato PNG y luego convertirlo a JPG si es necesario. Con la opción de autorefresco de MagicPoint podemos ir viendo como resulta el trabajo conjunto entre Dia y MagicPoint.

11. Soporte de Emacs y vi para MagicPoint

Tanto vi como Emacs tienen modos de edición específicos de edición para MagicPoint por lo que cualquiera de estos dos editores son opciones excelentes a la hora de trabajar con ficheros de Magic Point.

En realidad, la simplicidad de su sintaxis hace que cualquier editor de texto sea utilizable a la hora de la creación de presentaciones de MagicPoint. Recordar que con la actualización automática de MagicPoint, podemos ir viendo el resultado de nuestros cambios en el editor tan pronto los guardemos en un fichero, por lo que tenemos un entorno prácticamente visual de creación de presentaciones. Eso sí, tenemos el gran inconveniente de tener que conocer la sintaxis y posibilidades de MagicPoint para poder explotar el programa.

12. Otros programas de presentaciones

- KPresenter
- Open Office Impress
- Agnubis (alfa)

13. Conclusiones

MagicPoint ha logrado captar muchos adeptos dentro del mundo de software libre y no es nada extraño ver como se presentan muchas charlas con transparencias utilizando esta herramienta.

La velocidad a la que se pueden crear las presentaciones es difícilmente igualable desde cualquier otro programa de presentaciones, aunque al precio de tener que conocer la sintaxis de MagicPoint. Emacs con el formato XML puede ser una solución que incluso sea capaz de resolver este problema.

El motor de presentaciones es ligero y sencillo de utilizar en una presentación, permitiendo incluso pintar sobre las transparencias, por lo que es una opción realmente interesante para presentarse ante una audiencia.

Concluyendo, MagicPoint es una excelente opción a la hora de realizar presentaciones en GNU/Linux y en cuanto disponga de una interfaz gráfica, podrá competir de igual a igual con programas como Impress.

14. Referencias

- Herramientas para MagicPoint (<http://lionel.kr.fh-niederrhein.de/PAGES/dozenten/dalitz/data/software/mgp/mgp.html>)
- Fuentes True Type en GNOME 2
(<http://es.gnome.org/documentacion/articulos/truetype-gnome/truetype-gnome/book1.html>)