

# Manual de Grafoscopio

- Versión: 0.1
- Autor: Offray Vladimir Luna Cárdenas
- Correo: offray@mutabit.com

## Agradecimientos

Grafoscopio no hubiera sido posible sin la ayuda de muchas personas vinculadas a diversos lugares comunitarios e institucionales. Aquí un listado de algunos de esos lugares y personas.

- mutabiT.
- HackBo, el hackerspace de Bogotá, en particular las personas asistentes a los talleres de Indie Web Science y en especial Iván Pulido, Rafael Medina y Camilo Hurtado.
- Las comunidades de Pharo, Moose y Agile Visualization.
- El HiTec Lab de la Fundación Universitaria Los Libertadores.
- La Fundación Visonte.

Algunas personas han ayudado mucho con su escucha y apoyo, al margen de cualquier vinculación institucional y/o comunitaria. Entre ellas: Andrés Calderón, Hilda Cárdenas y Yanneth Gil.

## ¿Qué y para qué es grafoscopio?

Grafoscopio es una herramienta amoldable y flexible para la creación de *documentos* interactivos estructurados por capas.

Sirve para alentar nuevas prácticas escriturales y de visualización de datos que puede ser usado en distintos contextos: escritura académica, educación, narrativa y periodismo de datos, ciencia e investigación abiertas, ciudadanas y/o de garage, narrativa transmedia, activismo y muchos más.

Veámos en detalle algunos elementos de la definición:

- Sobre los *documentos* digimos que eran:
  - *Documentos estructurados y por capas*: Grafoscopio permite centrarse en las palabras y su estructura, sin distracciones de otra índole. Delega la apariencia a plantillas y programas externos permitiendo concentrarse sólo en la escritura y la estructura de la misma. Para la estructura usa un árbol que permite expresar secciones del texto: títulos, subtítulos, capítulos, ideas,

e incluso partes invisibles, que ayudan a redactar aquello que sí se va a ver. Para las palabras usa un sencillo y popular lenguaje de etiquetas (llamado markdown), que se aprende en pocos minutos y permite expresar elementos del texto: énfasis (*cursiva*), énfasis fuerte (**negrilla**), listas de items, numeraciones, tablas y referencias a gráficos.

- *Documentos Interactivos*: Grafoscopio permite que los documentos, además de palabras y estructura, incluyan automatismos que extraen y calculan información externa y crean gráficos dinámicos a partir de ella, entre muchos otros usos. Incluso todo el árbol de un documento es programable, con lo cual podemos usar un mismo árbol (o algunas de sus partes) para crear diversos documentos a partir de distintos recorridos, lo cual brinda posibilidades extensas: galerías de imágenes web a partir de todas las figuras referenciadas, exportar el documento como un micrositio web, tomar partes de Internet e integrarlas a un documento, etc. Te invitamos a aprender a programar, pero incluso si es una decisión que no has tomado, podrás emplear esas funcionalidades a partir de los *plugins* y extensiones que la comunidad irá construyendo.

- Sobre la herramienta digimos que era *amoldable y flexible*. Con esto nos referimos a que es fácil adaptar la herramienta a otros propósitos, contextos o intenciones. Esto definitivamente requiere de programación y de que entiendas el entorno de desarrollo de software integrado con grafoscopio, el paradigma de programación orientada a objetos y el lenguaje Smalltalk. Si ya sabes programar, este otro paradigma será beneficioso para mejorar la forma en que aboradas problemas, incluso si no lo usas en tu entorno cotidiano; y si aún no sabes, te invitamos a una experiencia con un nivel de flexibilidad, interactividad que no hemos encontrado en otros lados. Tomará tiempo y paciencia, pero será progresivo, muy gratificante, empoderador y definitivamente menos demorado que los abordajes usuales. Esto va en contra de los libros tipo “aprenda a programar en 21 días” y como decía “alguien en Internet”, no se encuentran libros similares sobre tocar un instrumento musical, pintar o aprender a cocinar. La maestría cuesta, sin embargo, grafoscopio está allí para ayudarte. Puedes pasar de los documentos estructurados, sin ninguna programación a los documentos interactivos, y de estos a la programación y modificación del entorno completo. Notarás que gracias a la uniformidad del entorno, las transiciones serán fluidas, en caso de que quieras emprenderlas, y dispondremos de espacios comunitarios (digitales y análogos), así como momentos de encuentro para facilitar el aprendizaje. La subsección “Aprende” de la página web referenciada al final, muestra algunos de esos espacios y momentos.

Ya sea que estés interesado en la escritura sin distracciones o en los documentos

dinámicos e interactivos, grafoscopio será una herramienta que te permita explorar y expresar tus ideas de modos más fluidos.

Para mayor información sobre Grafoscopio visita:

<http://mutabit.com/grafoscopio/>

## Instalación y ejecución

Empezar a usar grafoscopio será sencillo. Consistirá en 3 pasos: 1) descargar la versión para tu plataforma, 2) descomprimirlo y 3) ejecutarlo. Si lo quieres llevar contigo o copiarlo a otras máquinas, bastará con que copies la carpeta donde está el programa y listo! Incluso lo podrás ejecutar desde memorias USB.

### Gnu/Linux

- Descargar grafoscopio para Gnu/Linux de <http://mutabit.com/grafoscopio>
- Descomprimirlo.
- Ejecutar el archivo `grafoscopio.sh` que aparece en la raíz de la carpeta descomprimida

**ADVERTENCIA:** Si está ejecutando una distribución de Gnu/Linux de 64 bits, debe instalar el soporte para ejecución de las librerías de 32 bits. Usualmente se llaman ia-32.

### Windows

- Descargar grafoscopio para Gnu/Linux de <http://mutabit.com/grafoscopio>
- Descomprimirlo.
- Ejecutar el archivo `grafoscopio.bat` que aparece en la raíz de la carpeta descomprimida

### OSX

Aún no se ha empaquetado esta versión.

### Otras formas

Otra manera de instalar grafoscopio consiste en descargar primero la versión de Moose para nuestra plataforma y luego usar el gestor de paquetes que viene integrado para instalar desde allí Pharo. Esencialmente se ejecutan 4 pasos:

1. Descargar la variante de la máquina virtual de Pharo para [Windows](#), [GNU/Linux](#), ó [OSX](#) y descomprimirla.
2. Descargar la [imagen de Moose](#).
3. Lanzar la máquina virtual en la plataforma y abrir la imagen de Moose.
4. Abrir un playground y ejecutar lo siguiente:

```
Gofer new
  smalltalkhubUser: 'SvenVanCaekenberghe' project: 'STON';
  package: 'STON-Core';
  load.
```

```
Gofer new
  smalltalkhubUser: 'Offray' project: 'Grafoscopio';
  package: 'Grafoscopio';
  load.
```

Y listo!

Para ejecutar grafoscopio, dentro de un *workspace* ejecutamos:

```
GrafoscopioBrowser startDockingBar
```

## Panorámica Rápida

Si estás en esta parte del documento imaginamos que ya tienes instalado grafoscopio, e incluso que estás mirando este documento desde grafoscopio mismo. Si ese no es el caso, te recomendamos que:

- Vayas a <http://mutabit.com/grafoscopio>, instales grafoscopio, lo ejecutes y abras este documento haciendo click en el menú superior en **Ayuda > Manual en grafoscopio**.
- Maximices la ventana del documento y hagas click en la fecha que aparece junto a “Manual de Grafoscopio” y luego selecciones “Panorámica Rápida”. Deberías estar viendo este documento.

Ahora que estamos seguros de que estás leyendo este documento dentro de grafoscopio, te mostraremos cómo funciona la interface:

- A la izquierda tienes un árbol. Este jerarquiza la información en ramas o nodos. Un nodo consta de dos partes básicas:

- Una cabecera o título, que suele tener un texto corto, por ejemplo, para este nodo es “Panorámica Rápida”
- Un cuerpo o contenido, que ya despliega los textos detallados del nodo, por ejemplo, el texto que ahora lees.

A su vez un nodo puede agrupar a otros nodos. En ese caso el nodo que agrupa a los otros se conoce como padre y los agrupados en éste, se conocen como hijos. Por ejemplo, el nodo padre de este nodo se titula “Manual de Grafoscopio” y un nodo hijo de este nodo se titula “Modificar y exportar subárboles” Cuando un nodo tiene hijos, se reconoce porque tiene una flecha a su izquierda que apunta hacia la derecha. Si haces click en esa flecha, apunta hacia abajo y muestra los hijos del nodo. Si haces click en una flecha hacia abajo, apunta de nuevo a la derecha y oculta los hijos del nodo.

Lo anterior nos permite navegar el árbol de dos maneras: haciendo click en cada nodo, de acuerdo a lo explicado, o haciendo click en cualquier parte del árbol y desplazándonos luego con las flechas del teclado.

Para continuar, despliega el árbol de la izquierda y ubícate en el hijo de este nodo titulado “Modificar y exportar subárboles”

### **Modificar y exportar subárboles**

¡Bien! estás en el primer ejercicio. Si leíste el nodo “Panorámica rápida” ya sabes la terminología básica de nodos padres e hijos y también como navegar en un árbol o lo intuías por tu cuenta (la prueba es que estás acá).

Este primer ejercicio consistirá en aprender a modificar un documento, lo cual permitirá luego crear los propios.

Cuando estás dentro de una parte de un árbol, como acá, el nodo en el que estás parado, sus hijos, los hijos de esos hijos y así sucesivamente constituyen lo que se llama un “subárbol”.

Por ejemplo, si en el tutorial digital te paras en el nodo titulado “Subárbol de ejemplo” se despliega lo siguiente:

- Primera sección
- Segunda sección
  - Primera subsección
  - Segunda subsección
    - Primera sub-subsección
    - Tercera subsección

Como verás, podemos anidar información, con lo cual vamos dando cuenta de la estructura que queremos que tenga un documento. Si quieres, dale una mirada a

los contenidos del subárbol, pero *no olvides luego volver acá*, pues aquí continúa la explicación.

Bien, digamos que quieres ver cómo luce este subárbol como documento en HTML. Para exportar cualquier subárbol como HTML haces lo siguiente:

- Párate en el nodo que quieres exportar, en este caso el titulado “Subarbol de ejemplo”
- Haces click en el botón con puntos suspensivos “...”, que aparece más a la derecha (si, no es una muy buena elección para un ícono, estamos trabajando en ello).
- Elige la opción “Arbol > Exportar como HTML”. Te preguntará si quieres exportar todo el árbol, refiriéndose a todo el documento o el subárbol, refiriéndose a donde estás ubicado. Elige la segunda opción.
- Luego haz click en el menú de puntos suspensivos

### **Etiquetas, tipos de nodos, y modos**

**Nodos de configuración** Una mención especial merece el nodo “Configuración”. notarás que tiene una etiqueta con el nombre “config” y si haces click en él verás unos datos de la forma:

Eso es lo que se llama un diccionario en Smalltalk, y es una manera de representar atributos y sus valores. En este nodo, estamos diciendo dónde queremos que se guarde el documento que se produciría al exportar este subárbol. La mejor forma de verlo en acción es

### **Nodos invisibles**

### **Nodos de notas a pie de página**

### **Nodos de imágenes y figuras**

### **Nodos de código**

### **Nodos transmedia**

## **Escultores de icebergs o el por qué de grafoscopio**

Acá te contamos del lugar que espera llenar grafoscopio. De las fuentes de inspiración y los espacios que no creíamos que estuviesen ocupados.

## **La metáfora del iceberg para la crear, escribir y visualizar: lo que ves es sólo la superficie de lo que tienes**

El iceberg es una buena metáfora para la creación en general: el resultado es sólo muestra la superfiie de un proceso que es más profundo. En la mayoría de los proyectos, cada informe, texto, visualización, esta sustentado por un conjunto de soportes, autores, archivos e insumos, que quedan por fuera del documento que apeló a los mismos para existir. Dichas conexiones quedan ocultas bajo el texto o la gráfica; las tablas tienen los datos, pero no las consultas que los extrajeron, los insumos se desperdigan en la cabeza de los autores, las carpetas y referencias bibliográficas cuyo uso más allá de las citas. Se puede tener un texto sucinto y de fácil lectura o extenso y detallado, pero siempre se oculta las complejidades que lo construyen, lo cual dificulta la participación y transparencia posterior, propia de los procesos más horizontales e incluyentes.

Por ello se requieren metáforas escriturales distintas, que integren las distintas capas y niveles de profundidad propios de proyectos complejos como estos. En esencia se trata de pasar de las metáforas *Lo Que Ves Es Lo Que Obtienes* o WYSIWYG (por las siglas en inglés para *What You See Is What You Get*) a metáforas de “escritura tipo *iceberg*” donde *Lo Que Ves Es Sólo La Superficie De Lo Que Tienes*.

Grafosocpio intenta explorar un buen balance entre detalle y legibilidad usando las estructuras arbóreas en la que la escritura se organiza por capas. Las capas más profundas, contienen los detalles que permite la estructuración y escritura de documentos como los antes mencionados, y en las capas más externas, está aquello que el lector va a ubicar e incluso la organización para diferentes tipos de lectores, permitiendo a la vez dar cuenta, en un documento integrado, de recursos externos, anidamientos y relaciones internas y otros elementos constitutivos pero distintos del texto.

Este tipo alternativo de formas escriturales ayudaría en varios sentidos:

- Se alejaría de las comparaciones clásicas entres abordajes similares. Una metáfora escritural distinta requiere otra forma de pensar respecto al proceso de escritura que va más allá de la hoja en blanco y las carpetas, propias de los procesadores de palabra tradicionales.
- Facilitaría la colaboración entre los miembros de un grupo que cra en colectivo.
- Facilitaría la trazabilidad asociada a los procesos de creación.

La invitación es a sumergirnos juntos en esta exploración y construcción compartida.

## Fuentes de inspiración: Ideas prestadas y propias

Grafoscopio es un proyecto en permanente evolución y no lo digo como frase cliché, sino en el sentido de que intenta adecuarse a las necesidades cambiantes del entorno, es decir a las de sus usuarios y co-creadores (para ver un trazo de hacia dónde podría ir, te invito a consultar la sección titulada “Mapa de Ruta” en este mismo documento). De hecho, grafoscopio surgió de esas necesidades cambiantes y particulares, pues nada de lo que había hasta el momento daba cuenta precisamente la experiencia de escritura y uso del computador que yo quería tener y que, creo, que sería útil para más personas. El proyecto ha recibido inspiración de diversas fuentes como TeXmacs, Leo, IPython, LaTeX, Lout, pandoc, varios lenguajes de etiquetamiento ligero (ascii doc, markdown, reStructuredText y txt2tags), y comparte ideas con otros sistemas como pollen y org mode, que conocí después. Sin embargo, grafoscopio intenta crear una experiencia que no está propiamente en ninguna de las herramientas antes mencionadas (a pesar de que usa algunas de ellas por ejp pandoc y fossil, como piezas del software, que le dan nueva funcionalidad sin reinventar la rueda). Pharo/Smalltalk es el vehículo para explorar el encuentro, mutación y recombinación de estas ideas y el surgimiento de nuevas.

### Ideas similares:

- Al igual que LaTeX, TeXmacs y Lout intenta producir documentos de alta calidad visual, particularmente en su exportación como pdf.
- De IPython notebook (o Jupyter, el proyecto posterior), TeXmacs y Mathematica, toma la idea de escribir documentos interactivos que van más allá del texto y lo combinan con cálculos, visualizaciones, simulaciones, etc.
- De Leo y mathematica toma la idea de documentos computables. El documento es un árbol con ramificaciones que son calculables (celdas en mathematica e IPython y nodos en Leo). Al igual que en Leo las partes del árbol (los nodos) pueden introducir automatismos que modifican el documento en parte o en su totalidad o incluso podrían modificar el funcionamiento e interface del software (como los macros de la ofimática clásica). Esta idea de un documento programable y que puede programar todo el entorno ha estado en sistemas como Lisp desde los 60's e emacs y ha tomado cuerpo en sucesores recientes de este legado como Pollen/Racket y org-mode.
- De los lenguajes de etiquetamiento ligero, toma la idea de una escritura centrada en el texto, en las palabras, que se puede escribir fácilmente y sin preocuparse por adornos visuales al delegar los temas de presentación a hojas de estilo y programas externos (idea que está en otros sistemas como Ulysses, Substance, IPython, Ghost, etc). text2tags en particular merece una especial mención, por su sintaxis extensible por el usuario vía expresiones regulares. Y aunque es una forma de extensión poco elegante, comparada con que provee Pollen, vía funciones, o el mismo Smalltalk, vía

mensajes, si fue una primera aproximación a la idea de que un lenguaje de documentación podía permanecer simple a la vez de extensible.

#### Ideas diferentes:

- A diferencia de TeXmacs, LaTeX, Leo e IPython, intenta tener un empaquetado e instalación sencillas, tipo descargue, descomprima y use, sin pensar mucho en el sistema operativo, los gestores de paquetes, etc. Las dependencias externas deben ser mínimas, portables (compiladas estáticamente) y venir dentro del instalador en el archivo comprimido (pandoc, fossil y a futuro lout, constituyen dichas dependencias)
- A diferencia de Leo, se concentra en la experiencia de escritura de textos interactivos. Leo es más versátil, pero también más complicado, pues tiene que lidiar con cómo se integra con archivos externos y las instrucciones uso e instalación y modificación están orientadas a un público más técnico (casi cualquier cosa se tiene que hacer editando archivos de Leo). Los documentos interactivos tiene capas, a diferencia de IPython, donde, si el documento se pone muy largo, puede hacerse pronto inmanejable y el afinamiento para convertirlo en pdf implica afinar manualmente el archivo LaTeX intermedio.
- A diferencia de Leo, TeXmacs, IPython es más fácil de entender, extender y modificar. Fuí usuario de Leo varios años y sólo hasta los últimos años empecé a cambiarlo de modo superficial. Los cambios que hice a TeXmacs también se vieron limitados. Traduje buena parte del manual a español e hice algunos scripts para personalizar los menús de la interface. TeXmacs está hecho con Scheme y C++, lenguajes bien diferentes y si bien con el primero se podría hacer parte de los cambios, si se quiere llegar a profundidad, se requiere del segundo. IPython consta de múltiples capas que requieren conceptos y tecnologías diversos para hacer que funcione: json, python, html, zeroMQ, programación de cliente y servidor, lenguajes de *scripting*, programación imperativa y objetual.

Todas las anteriores herramientas están hechas desde la tradición Unix de tener archivos estáticos, que se editan desde editores externos, con idiosincrasias distintas, lo cual genera una curva de aprendizaje más inclinada. En cambio, creé el primer alfa grafoscopio en un par de meses, incluido el aprendizaje de los rudimentos de Smalltalk (lenguaje y entorno). Esto es un mérito del entorno altamente interactivo, uniforme e integrado y la sintaxis sencilla y poderosa que provee Pharo/Smalltalk, frente al paradigma más convencional y extendido, pero menos fluido, de los sistemas operativos.

Dicho todo esto, grafoscopio es una herramienta que espera tener su propio lugar. No pretende desplazar a ninguna y, como dije, le debe mucha inspiración a las que he mencionado. Guardo profundo respeto y admiración por los autores, las

comunidades detrás de tales herramientas y por las herramientas mismas y si bien las uso menos en la medida que grafoscopio madura, siguen teniendo un lugar en mi baticinturón (y mi corazón :-P).

## Exploración Recomendada

Estamos preparando unos documentos interactivos para la exploración de grafoscopio. Puedes descargarlos, abrirlos en grafoscopio y mirar cómo funcionan. Aquí están los enlaces:

- Agile Visualization: | [Sitio Web](#) | [Documento de grafoscopio](#)

## Mapa de Ruta

¿Cuándo un artefacto (digital o no) está listo para liberarse al público? Me parece que la respuesta tiene que ver con la experiencia de uso. El principal criterio de dicha experiencia es que no debe ser confusa. Esto no quiere decir que esté completa o que sea totalmente amigable o “intuitiva” (ese término siempre me ha parecido engañoso y me pregunto: ¿de acuerdo a la intuición de quién?). Se trata sobre todo de que las personas sepan qué esperar del artefacto.

Muchas de estas expectativas se pueden informar con la documentación. Algo que haga claro cuáles son las ofertas, limitaciones y potencialidades del artefacto. Este mapa de ruta intenta aclarar esos aspectos.

He organizado el mapa de ruta en forma de un tablero Kanban, con tres lugares “Por Hacer”, “Hecho” y “Haciendo”, que junto con el repositorio de código fuente (en Monticello) y de documentación (en fossil) del proyecto muestran cómo se van implementando las diferentes características que se integran en la experiencia de usuario. Puedes revisar cada una de las tres secciones del Kanban para ver en qué estamos trabajando, que hemos resuelto y qué tenemos proyectado hacia el futuro.

Para la primera versión pública grafoscopio debería desde soportar una primera experiencia tres pasos: descargue, descomprima y explore. Esto quiere decir que debe tener una barra de tareas con acceso a las funcionalidades más empleadas y la documentación en distintos formatos.

De ahí en adelante se puede establecer un modelo de *rolling release*, similar al de algunas distribuciones del sistema operativo Gnu/Linux, como Arch, en el que acceder a nuevas características nuevas del sistema, simplemente sea actualizarse a la última versión (para esto actualmente existe una opción de menú integrada a la interfaz gráfica, que hace esto).

Ese es el mapa, por lo pronto, pero se reciben sugerencias de futuras rutas.

## Kanban

**Por Hacer** Las características futuras previstas son:

- Un procedimiento asistido para la instalación de TeX.
- Soporte para creación de pdf locales vía Lout y un conversor de postscript a pdf (¿quizás ghostscript?).
- Más ejemplos de documentos interactivos, particularmente asociados a ciencia abierta, ciudadana y de garage.
- ¿Pruebas unitarias? (son necesarias, lo que no sé es si para esta versión ya tenemos rudimentos de como se hacen).
- Mejorar los nodos de texto con:
  - Botones para embeber notas al pie de página, imagenes, cambiar el tamaño de las fuentes.
  - Resaltado sintáctico para markdown: Los nodos de texto actuales no muestran pistas de color que permitan guiar al autor. La idea es que se soporte opciones de resultado sintáctico y hacia futuro algunas ayudas tipográficas (cursivas, negrillas, etc). Luego se pueden dar soporte a otros lenguajes.
  - Mejor comportamiento como editor de texto, respetando el espacio en blanco con los saltos de línea.
- Objetos en los nodos: Actualmente tenemos dos tipos de nodos, aquellos que son sólo texto, y los playgrounds, pero aún no están tan bien integrados como se podría. Se requiere un mejor tiepo de integración, haciendo que el nodo genérico sorpote distintos tipos de mensajes de los nodos embebidos y delegue a los mismos disintas comportamientos. También se requieren distintos tipos de nodos, empezando por los nodos de imagen, quer permitan ver cómo lucen tales imagenes y enviarles mensajes (escalarlas, exportarlas, etc).
- Aztlek propone la idea de que el árbol de grafoscopio permita cambiar el código fuente del mismo, o presentarlo de manera arbórea, en lugar de desde el navegador de clases. Es una idea interesante, habría que mirar cuánto del cambio de interface se requiere o se puede mapear de uno al otro, pues el navegador de clases es bastante poderoso y adecuado para la escritura de código, mientras que grafoscopio está centrado en la escritura de texto, acompañado por código y visualizaciones.

## Haciendo

- [X] Integrado pandoc para la exportación a distintos formatos y fossil para el control histórico de documentos y la colaboración.
- En caso de que se quiera exportar como pdf debe indicar que se debe instalar TeX en la versión de Windows, Linux o Mac, con los enlaces respectivos.

## Hecho

### Barra de herramientas fija

- Una barra de herramientas fija con los siguientes menus:
  - Lanzar
    - \* Nuevo documento interactivo
    - \* Documentos interactivos recientes ...
    - \* Documentos interactivos de ejemplo.
  - Ayuda.
    - \* Manual en HTML La cargaría desde el sistema de archivos y vendría integrado en la descarga.
    - \* Manual en Pdf .
    - \* Arbol de grafosocpio Se debe colocar la ayuda en formatos distintos a grafosocpio, pues las gráficas integradas no se ven dentro del árbol.