

Manual de Freeplane



Luis Javier González Caballero
kewapo@gmail.com

31 de enero de 2019

Acerca del autor



Luis Javier González Caballero es un Técnico en Informática con amplia experiencia en soporte a usuarios. Ha realizado algunos trabajos de programación, fundamentalmente en Python. Es Ingeniero Técnico Industrial por la [Universidad de León](#) e Ingeniero en Informática por la [Universidad de Educación Nacional a Distancia](#).

Es una persona activa con una gran capacidad de aprendizaje, lo que le permite ponerse al día rápidamente en los proyectos en los que se involucra.

Aparte de su carrera profesional colabora con numerosas organizaciones sin ánimo de lucro y proyectos libres y es amante de los animales. Vive en León, una pequeña ciudad en el noroeste de España con su familia.

@ kewapo@gmail.com

Índice general

I	Freeplane básico	8
1.	Mapas mentales	9
1.1.	Introducción	9
1.2.	La sociedad de la información	9
1.3.	El cerebro	10
1.4.	Procesando la información	11
1.5.	Elaboración de los mapas mentales	11
1.6.	Ejemplos	12
1.6.1.	Mapas al estilo de Tony Buzan	12
1.6.2.	Mapa de Gestión	13
1.6.3.	Mapa circular	15
2.	Comenzando con Freeplane	17
2.1.	Instalación	17
2.2.	La ventana de la aplicación	17
2.3.	Creando el primer mapa	19
2.4.	Uso del mapa	20
2.5.	Elementos de un mapa	22
2.6.	Elementos de un nodo	23
3.	Personalizando nuestro mapa	25
3.1.	El panel de formato	25
3.2.	Modificando el texto del nodo	25
3.3.	Aspecto de los nodos	26
3.4.	Iconos	28
3.5.	Nubes	28
3.6.	Colores	28
3.7.	Aspecto de los bordes	29
4.	Añadiendo otros elementos	31
4.1.	Imágenes	31
4.1.1.	Imagen dentro de un nodo	31
4.1.2.	Imagen bajo el nodo	32
4.1.3.	Imagen de fondo del mapa	32
4.2.	Atributos	33
4.3.	Enlaces	33
4.3.1.	Enlaces internos	34
4.3.2.	Enlaces externos	34
4.3.3.	Enviar correo	34
4.3.4.	Enlace a un menú de freeplane	34
4.3.5.	Enlace a un comando del sistema operativo	34
4.4.	Fórmulas	34
4.4.1.	Elementos básicos	35
4.4.2.	Elementos de los nodos	35

4.5. Fechas	36
5. Definiendo estilos y plantillas	39
5.1. Introducción	39
5.2. Estilos predefinidos	39
5.3. Modificando los estilos ya definidos	40
5.4. Guardando nuestros estilos: las plantillas	41
5.4.1. Utilizar estilos definidos en otro mapa	41
5.4.2. Guardar el mapa actual como plantilla	42
5.4.3. Usar la plantilla	42
5.5. Creando una plantilla para nuestros proyectos	42
6. Utilizando el mapa con otras aplicaciones	46
6.1. Introducción	46
6.2. Utilizándolo como imagen	47
6.3. Guardando el esquema de la información	47
6.4. Mapa como página web	48
6.5. Importando datos	49
6.6. Importar los favoritos del navegador	50
6.7. Importar la estructura de carpetas de nuestras unidades	50
7. Buscando información en nuestro mapa	51
7.1. Búsqueda rápida	51
7.2. Expresiones regulares de Freeplane	52
7.3. Filtros	52
7.3.1. Gestión de los filtros guardados	54
8. Haciendo presentaciones con Freeplane	55
8.1. Introducción	55
8.2. Enseñando nuestro mapa	55
II Freeplane avanzado	57
9. Texto avanzado	58
9.1. HTML en los nodos	58
9.2. Tutorial rápido de HTML	59
9.2.1. Formato básico	59
9.2.2. Caracteres especiales	59
9.2.3. Enlaces	60
9.2.4. Referencia a imágenes	60
9.2.5. Tablas	60
9.2.6. Listas	61
9.2.7. Definiciones	61
9.2.8. Definiendo estilos	62
9.3. Látex en los nodos	62
9.3.1. Qué es latex	62
9.3.2. Utilizando latex en nuestros mapas	62
9.3.3. Algunos comandos	63
9.3.4. Más comandos	63
9.4. Aprendizaje mediante mapas mentales	64
10. Formato avanzado	67
10.1. Estilos condicionales	67

11. Presentaciones con Freeplane	69
11.1. Opciones simples para mostrar el mapa	69
11.2. Presentaciones complejas en freeplane	69
11.3. Diseño de presentaciones	70
12. Programando scripts	72
12.1. Introducción	72
12.2. Tutorial rápido de groovy	72
12.2.1. Datos básicos	72
12.2.2. Operadores	73
12.2.3. Palabras reservadas	73
12.2.4. Sentencias de control	73
12.2.5. Expresiones regulares	74
12.2.6. Closures	75
12.2.7. Closures y listas	76
12.2.8. Métodos usados con closures	76
12.2.9. Listas	77
12.3. Configuración de Freeplane para la ejecución de scripts	78
12.4. Comenzando a programar scripts	78
12.4.1. El editor de scripts	78
12.4.2. La API de freeplane	79
12.4.3. El primer script	80
12.4.4. Elementos de un nodo	80
12.4.5. Operaciones con nodos	81
12.4.6. Configuración de Freeplane para la ejecución de scripts	82
12.5. Scripts de inicio	83
12.6. Examinando propiedades y métodos de las clases de groovy	83
12.7. Uso de librerías externas	83
12.8. Ejemplos	83
12.8.1. Actualiza el atributo status de los nodos del mapa	84
13. Fórmulas avanzadas	87
13.1. Introducción	87
13.2. Cómo referirnos a los elementos del mapa	87
13.3. Funciones de conversión	88
13.4. Funciones de listas	88
13.5. Utilización de closures en las fórmulas	89
13.6. Groovy en las fórmulas	89
13.7. Ejemplos	90
13.8. Nueva forma de referirse a los nodos	90
13.8.1. Conceptos previos	90
13.8.2. Acceso a nodos en fórmulas y scripts	91
13.8.3. Otras fórmulas	92
13.8.4. Acceso a los nodos en los hiperenlaces	92
14. Configurando Freeplane	94
14.1. El directorio del usuario	94
14.2. Opciones de configuración	94
14.3. Opciones de la línea de comandos	96
14.4. Variables de entorno	96

III	Desarrollando Freeplane	97
15.	Colaborando en el desarrollo de Freeplane	98
15.1.	Conceptos	98
15.2.	Configuración de un entorno de desarrollo de scripts.	98
15.3.	Desarrollar add-ons	99
15.4.	Desarrollando Freeplane	99
15.5.	Qué necesitaremos	100
15.6.	Obteniendo el código más reciente.	101
15.7.	Importándolo	101
15.8.	Comenzando el desarrollo.	102
IV	Apéndices	103
16.	Glosario	104

Índice de figuras

1.2.1. Mapa conceptual	10
1.2.2. Mapa mental	10
1.6.1. Mapa al estilo de Tony Buzan	12
1.6.2. Mapa usado para la gestión	13
1.6.3. Mapa para gestionar una agenda	14
1.6.4. Mapa circular	15
1.6.5. Mapa: ¿Quién soy?	16
2.2.1. Pantalla de Freeplane	18
2.3.1. Primer paso	19
2.3.2. Segundo paso	19
2.3.3. Tercer paso	20
2.4.1. Mapa expandido	21
2.4.2. Mapa colapsado	21
2.5.1. Elementos de un mapa	22
2.6.1. Elementos de los nodos	23
3.2.1. Modificando el texto del nodo	25
3.2.2. Modificando el texto del nodo	26
3.3.1. Modificando la forma del nodo	26
3.3.2. Formas disponibles para los nodos	27
3.3.3. Opciones de los bordes	28
3.5.1. Tipos de nubes	28
3.6.1. Colores del nodo	29
3.6.2. Colores de los bordes	29
3.6.3. Diferencia entre Bordes y Borde del nodo	29
3.7.1. Opciones de los enlaces entre nodos	30
3.7.2. Tipos de conexión entre nodos (bordes)	30
4.1.1. Añadir imágenes a los nodos	31
4.1.2. Mapa con una imagen de fondo	32
4.2.1. Añadiendo atributos	33
4.3.1. Nodo con enlace	33
4.4.1. Fórmula	35
4.4.2. Mapa ejemplo para las fórmulas	36
4.4.3. Ejemplo de un nodo con fórmula	37
4.5.1. Gestión del tiempo	37
4.5.2. Planificador del tiempo	38
5.1.1. Definiendo estilos	39
5.2.1. Estilos de Freeplane	40
5.2.2. Panel del estilo del nodo	40
5.3.1. Editar estilos	41
5.5.1. Creación de una plantilla. Fase 1	42
5.5.2. Definiendo el estilo para las facturas	43

5.5.3. Creación de una plantilla. Fase 2	43
5.5.4. Estilos de nuestra plantilla	44
5.5.5. Nuevo mapa creado a partir de la plantilla del proyecto	45
6.1.1. Ventana del menú Archivo / Exportar.	46
6.3.1. Exportamos el mapa como archivo de excel	48
6.5.1. Importar copiando de un archivo de texto	49
7.3.1. Barra de herramientas de filtros	52
7.3.2. Ventana de filtros	53
9.1.1. Edición avanzada del nodo	58
9.3.1. Ejemplo de \LaTeX	64
9.4.1. Dividir Ideas	65
9.4.2. División horizontal y vertical del conocimiento	65
9.4.3. Ejemplo de división del conocimiento	66
10.1.1. Estilos condicionales	68
11.2.1. Generar presentaciones	71
12.2.1. Objetos de la API de Freeplane	78
12.4.1. Consola de groovy	79
12.4.2. Documentación de la API	80
12.8.1. Nodo INSERTAR	84
13.8.1. Configurar la referencia del nodo	91
13.8.2. Nueva forma de referirse al nodo	91
13.8.3. Ejemplo de las nuevas referencias	93

Parte I

Freeplane básico

Capítulo 1

Mapas mentales

1.1. Introducción

Un mapa mental es un diagrama usado para representar palabras, ideas, tareas, lecturas, dibujos, u otros conceptos ligados y dispuestos radicalmente alrededor de una palabra clave o de una idea central. Son una forma lógica y creativa de tomar notas, organizar, asociar y expresar ideas, que consiste, literalmente, en cartografiar nuestras reflexiones sobre un tema. Los mapas mentales son herramientas muy útiles::

Qué son

- Favorecen la producción de ideas.
- Mejora la productividad.
- Permite apreciar simultáneamente la visión global del problema así como sus detalles.
- Favorece la memorización.

En este documento vamos a presentar Freeplane, una aplicación para crear mapas mentales. Es multiplataforma, por lo cual funciona en Windows, Linux y Mac. No solo nos permiten dibujar mapas mentales. Además también tiene herramientas para visualizarlos: podemos plegar nodos, desplegarlos, hacer zoom, añadir notas y fórmulas a los nodos y muchas otras cosas que veremos en este manual.

Los elementos constructivos básicos son los nodos y los enlaces entre ellos. Podemos ver el área de trabajo de Freeplane en la figura 2.2.1. La web donde podemos encontrar esta aplicación es www.freeplane.org desde donde podremos descargarla.

1.2. La sociedad de la información

La sociedad actual es considerada como sociedad de la información. Tenemos acceso a ingentes cantidades de datos y conceptos que, o bien necesitaremos integrar en la estructura conceptual de nuestro cerebro o analizar las opciones que se nos presentan. Esta exposición constante a nuevos datos se da en todos los estadios de desarrollo de la persona, desde la escuela hasta el mundo de la empresa.

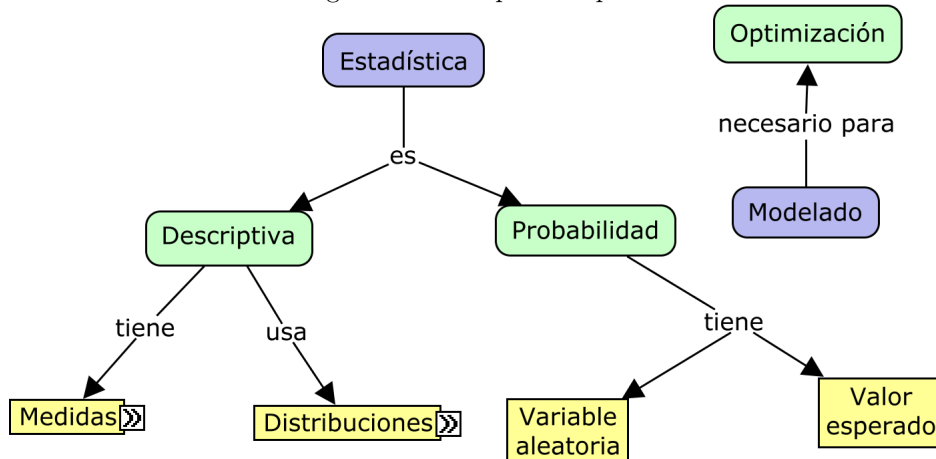
Exceso de información

- En el mundo académico confiamos en que las capacidades de aprendizaje de nuestro cerebro nos sirvan para salir airoso.
- En el mundo empresarial confiamos tanto en nuestra capacidad de análisis como en nuestra intuición para realizar un análisis correcto. A menudo efectuamos listas, planes y programaciones para ayudarnos.

Sin embargo contamos con herramientas más potentes que pueden facilitarnos estas tareas:

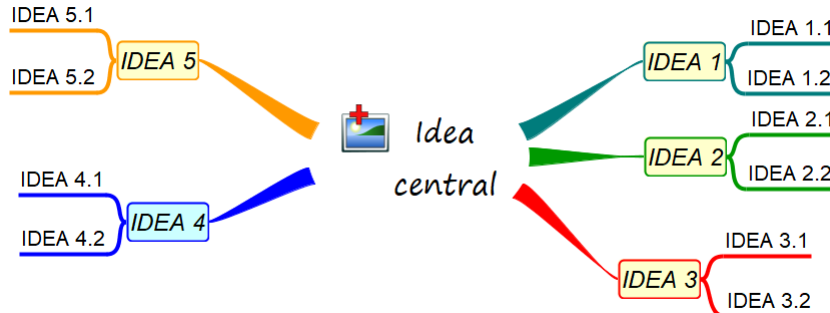
mapas conceptuales: Diagrama en el cual anotamos en cajas los conceptos que nos interesan, indicando mediante líneas que los enlazan las relaciones existentes entre ellos. A diferencia de los mapas mentales no se distribuyen entorno a una idea central. Tienen una forma de red conceptual y se utilizan en la comprensión de problemas complejos.

Figura 1.2.1: Mapa conceptual



mapas mentales: Anotamos las distintas ideas entorno a una idea central. Facilita la generación de ideas y sirve para aclarar problemas complejos. También simplifica la memorización y aprendizaje significativo.

Figura 1.2.2: Mapa mental



1.3. El cerebro

El cerebro de las personas se estructura en capas. En la más externa tenemos las distintas capacidades conscientes. Con ella podemos dirigir nuestro pensamiento y utilizar sus habilidades analíticas, visuales, auditivas, etc. en nuestro beneficio de forma consciente. Sin embargo hay capas internas a las que es más complicado acceder. Conforman nuestra intuición y es donde el cerebro envía toda la información “vieja” para evitar saturar a la capa externa. A menudo este acceso lo obtenemos en forma de ideas repentinas ó intuiciones, asociaciones de ideas o simplemente sensaciones o “corazonadas”.

Si bien podemos usar conscientemente nuestras habilidades analíticas, visuales, etc, no podemos decir lo mismo de las habilidades que residen en las zonas interiores. Nuestro cerebro funciona mediante asociaciones automáticas: al pensar en una idea el cerebro la encadena con nuevos conceptos. Para potenciar nuestra capacidad mental y nuestra creatividad podemos servirnos de los mapas mentales cuya eficacia se basa en el pensamiento radiante: el cerebro genera nuevas ideas a partir de una idea principal básica.

De esta forma un mapa mental es una herramienta gráfica que nos da acceso al enorme potencial de nuestro cerebro. Su eficacia se basa en que funciona de forma análoga a la de nuestra mente: del mismo modo que la mente funciona describiendo círculos alrededor de una idea central, en un mapa mental

colocamos la idea sobre la que queremos trabajar en el centro y añadimos nuevas ideas a medida que van surgiendo situándolas alrededor de esa idea principal. Se desarrolla desde el centro y nos permite tener una visión global y organizada de la información. Es la forma más sencilla de gestionar el flujo de información que genera nuestro cerebro.

1.4. Procesando la información

La antigua manera de procesar información linealmente de izquierda a derecha crea numerosos problemas cuando se trata de organizar todos los datos que recibimos. Necesitamos un modo más eficaz para representar todo ese torrente de ideas. No sólo en párrafos sucesivos de texto escrito sino también englobando elementos cognitivos de ambos hemisferios cerebrales. Los mapas mentales nos permiten “Ver nuestros pensamientos”. Podemos utilizarlos para:

- Organizar la información.
- Separar lo trivial de lo importante.
- Recopilación sencilla y flexible de datos.
- Estructurar hechos.
- Aclarar ideas.
- Facilitar la comprensión de temas complejos.
- Retención natural y práctica de las ideas.
- Planificación estratégica.
- Comunicaciones efectivas.

1.5. Elaboración de los mapas mentales

Para la construcción de un mapa mental partiremos de una superficie blanca sin líneas ni marcas. Lo primero y más importante a la hora de su elaboración es tener claro su propósito. Para ello definiremos claramente la idea principal. Estos son los pasos para su construcción:

Elaboración
de mapas
mentales

1. Identificar la idea principal. Anotarla en el centro del mapa.
2. En torno a esta idea central ir anotando las ideas relacionadas con el tema. Escribir estos subtemas alrededor de la idea central. Tener en cuenta que pondremos cada idea en una rama diferente del mapa.
3. Añadir colores, imágenes y símbolos que identifiquen a todas estas ideas. Esto ayudará a nuestro cerebro en la elaboración del mapa.
4. Desarrollar los subtemas ramificando las ideas que parten del centro del mapa. Hay que simplificar al máximo el contenido del mapa.

Hay varias reglas que nos pueden ayudar:

1. Empieza en el centro de la hoja en blanco. De esa forma las ideas podrán moverse en todas direcciones.
2. Dibuja en el centro una imagen que simbolice la idea principal. Así establecerás un núcleo de atención.
3. Utiliza muchos colores. Los colores despiertan tu creatividad.
4. Partiendo de la idea principal traza ramas de distintos colores hacia el exterior con las palabras clave más importantes. Haz las líneas más finas a medida que te alejas del centro.

5. Traza líneas curvas en lugar de rectas. Son menos aburridas.
6. Trata de utilizar una o dos palabras clave por cada línea. Es más fácil que recordar una frase completa.
7. Utiliza todas las imágenes que puedas.

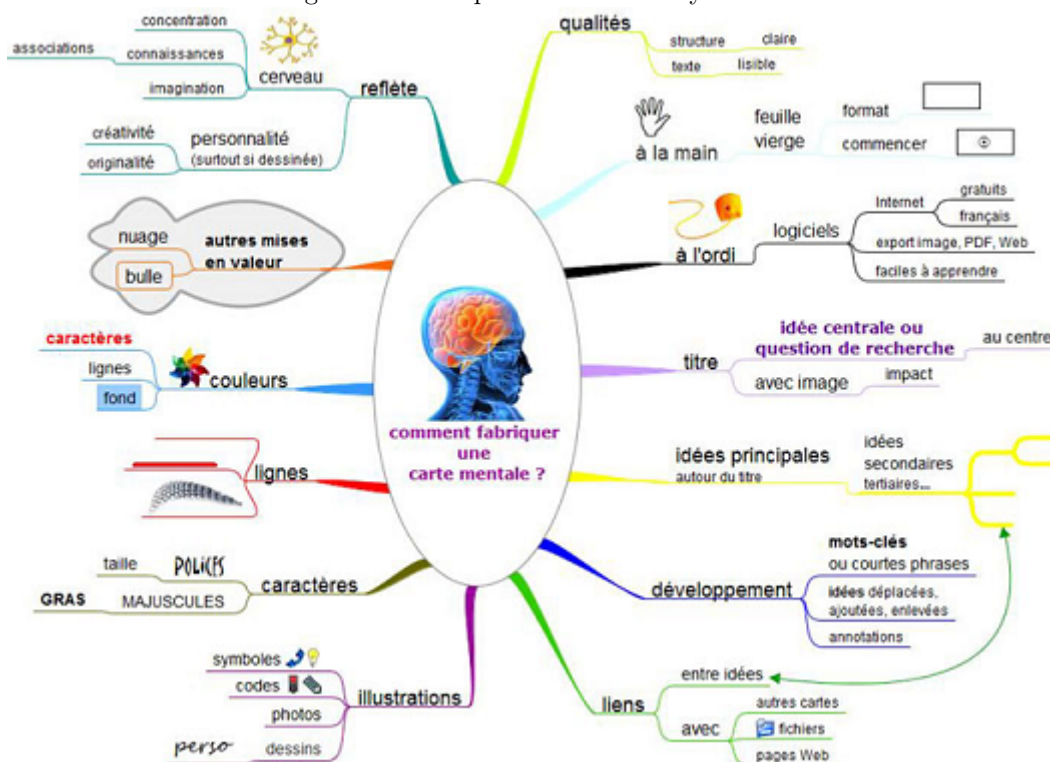
Aunque podemos construirlos utilizando papel, lápices y pinturas el uso de aplicaciones específicas añaden rapidez y versatilidad en su realización. Este tipo de software nos permite de forma sencilla no solo “pintar” los conceptos y su relación, sino también modificar su estructura cambiando la ubicación de ramas y nodos según nos apetezca. Además como tenemos acceso a gran cantidad de iconos, imágenes e ilustraciones de forma muy sencilla su uso en nuestro software favorito de mapas mentales potencia aún más su utilidad. Freeplane pone a nuestro alcance una enorme cantidad de posibilidades que analizaremos en este manual.

1.6. Ejemplos

Vamos a ver qué podemos esperar. Freeplane es una aplicación muy versátil. Puede crear mapas mentales de estilos muy distintos. Desde los aconsejados por Tony Buzan con colores e imágenes hasta uno más sobrio con líneas rectas y sin adornos. A continuación se muestran algunas ideas.

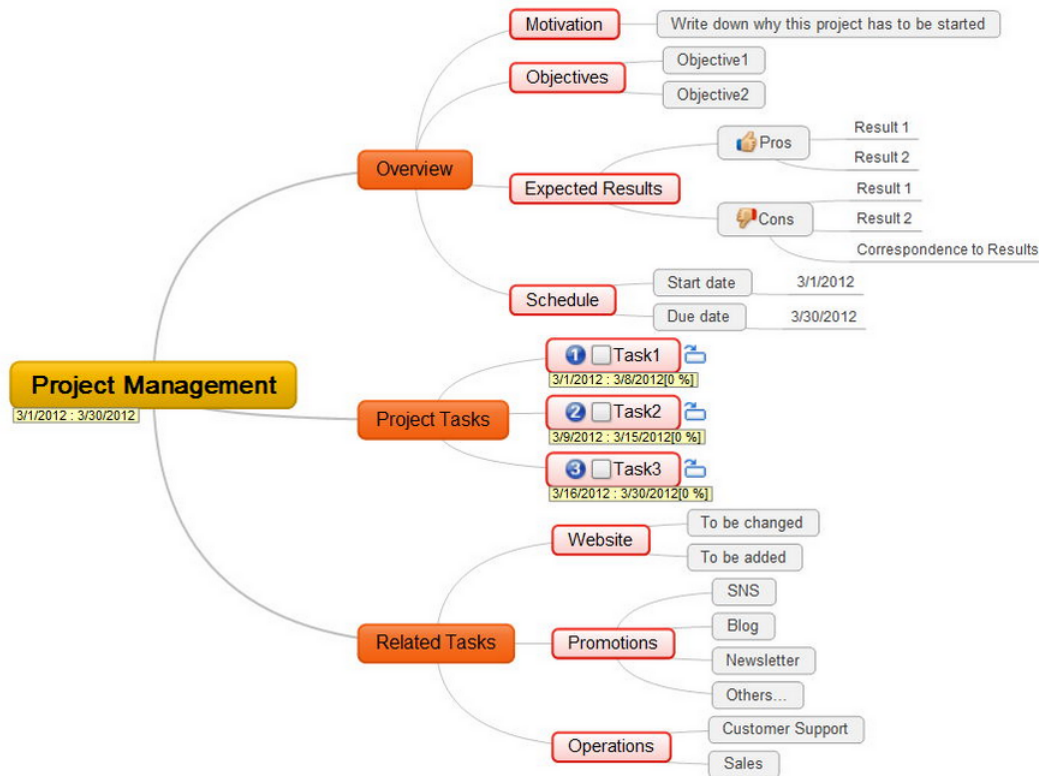
1.6.1. Mapas al estilo de Tony Buzan

Figura 1.6.1: Mapa al estilo de Tony Buzan



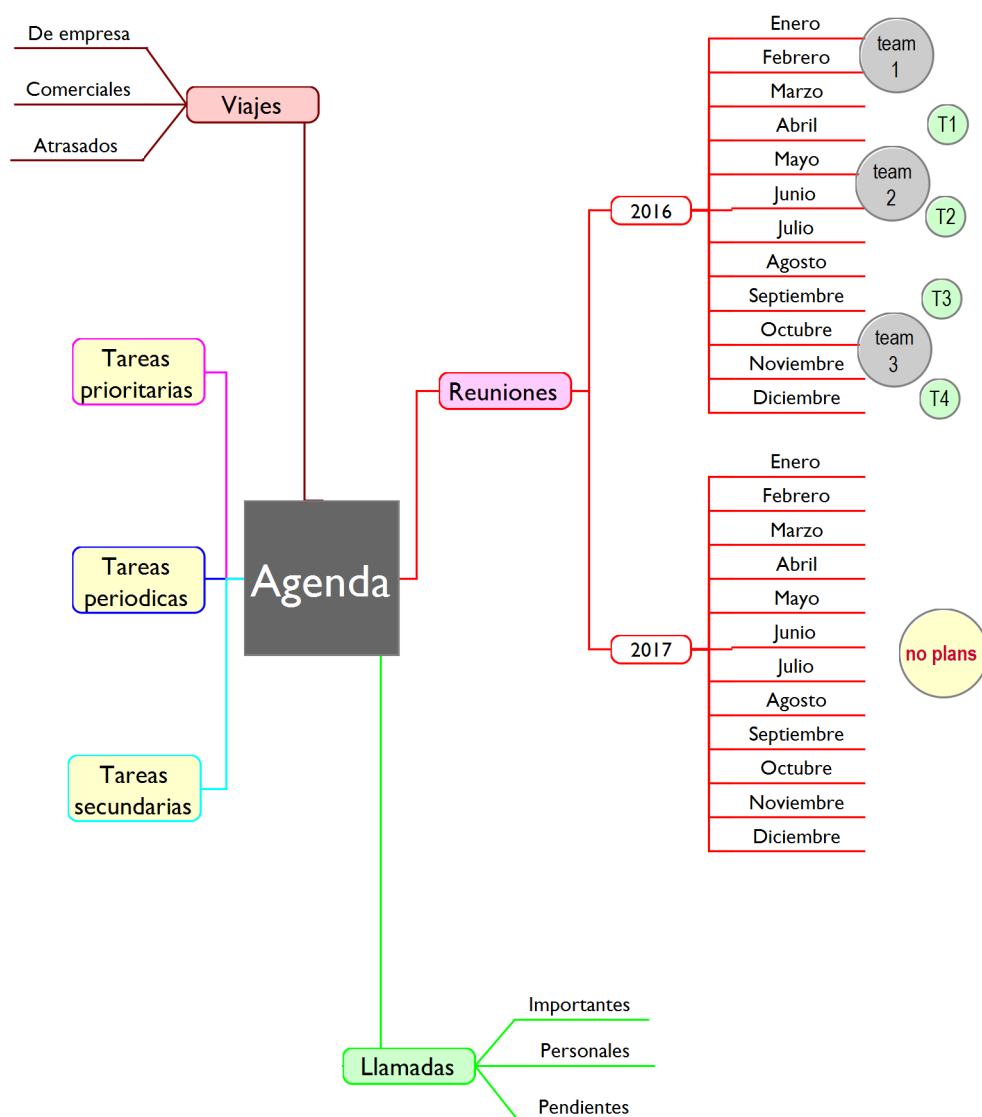
1.6.2. Mapa de Gestión

Figura 1.6.2: Mapa usado para la gestión



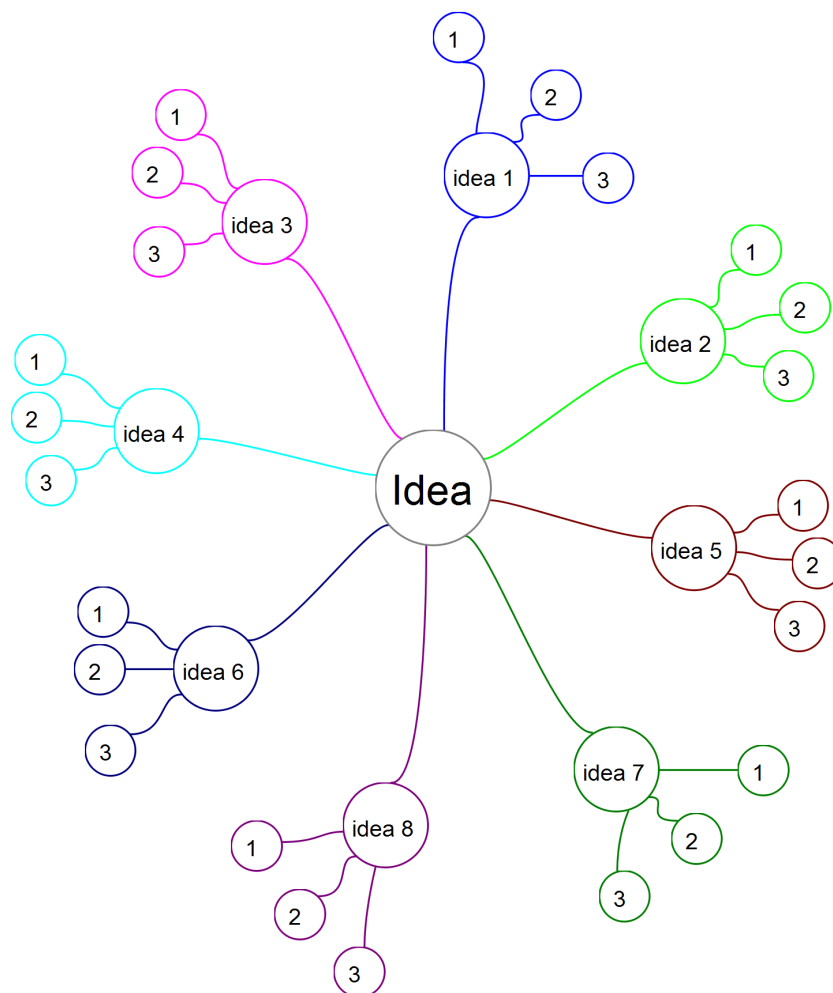
Estos mapas utilizan líneas horizontales y nodos en burbuja. Se le pueden añadir ciertas imágenes. Asimismo, suelen utilizarse nodos de anchos uniformes (igual valor en máximo y en mínimo). En las figuras aparecen dos ejemplos de este tipo de mapas

Figura 1.6.3: Mapa para gestionar una agenda



1.6.3. Mapa circular

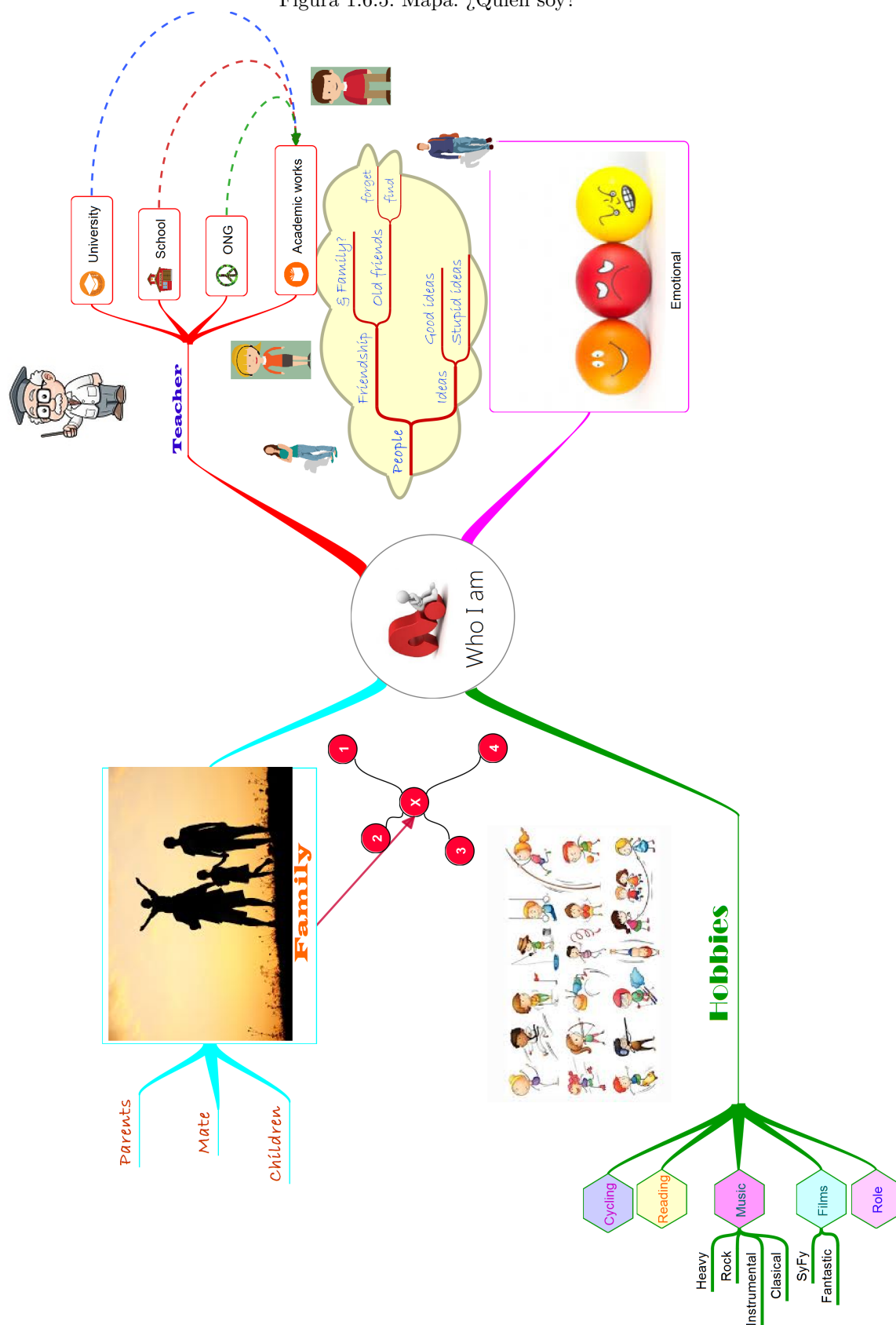
Figura 1.6.4: Mapa circular



Si damos a los nodos forma oval y marcamos uniforme podemos conseguir el efecto que aparece en la imagen.

Utilizando correctamente todos los elementos antes descritos tenemos una herramienta muy versátil.

Figura 1.6.5: Mapa: ¿Quién soy?




Capítulo 2

Comenzando con Freeplane

2.1. Instalación

Puedes encontrar Freeplane en <https://www.freeplane.org>. Sólo tienes que acceder a este sitio y descargarla pinchando en los enlaces que aparecen allí. Como verás, se distribuye con licencia “GNU Free Documentation License” que permite su uso gratuito y libre. Hay versiones para Windows, Linux y Mac lo que la convierte en una de las aplicaciones para mapas mentales más versátiles. Para instalarla únicamente tienes que lanzar el ejecutable y seguir los pasos que te van apareciendo. Es una tarea sencilla. Únicamente asegúrate de tener actualizada tu versión de Java.

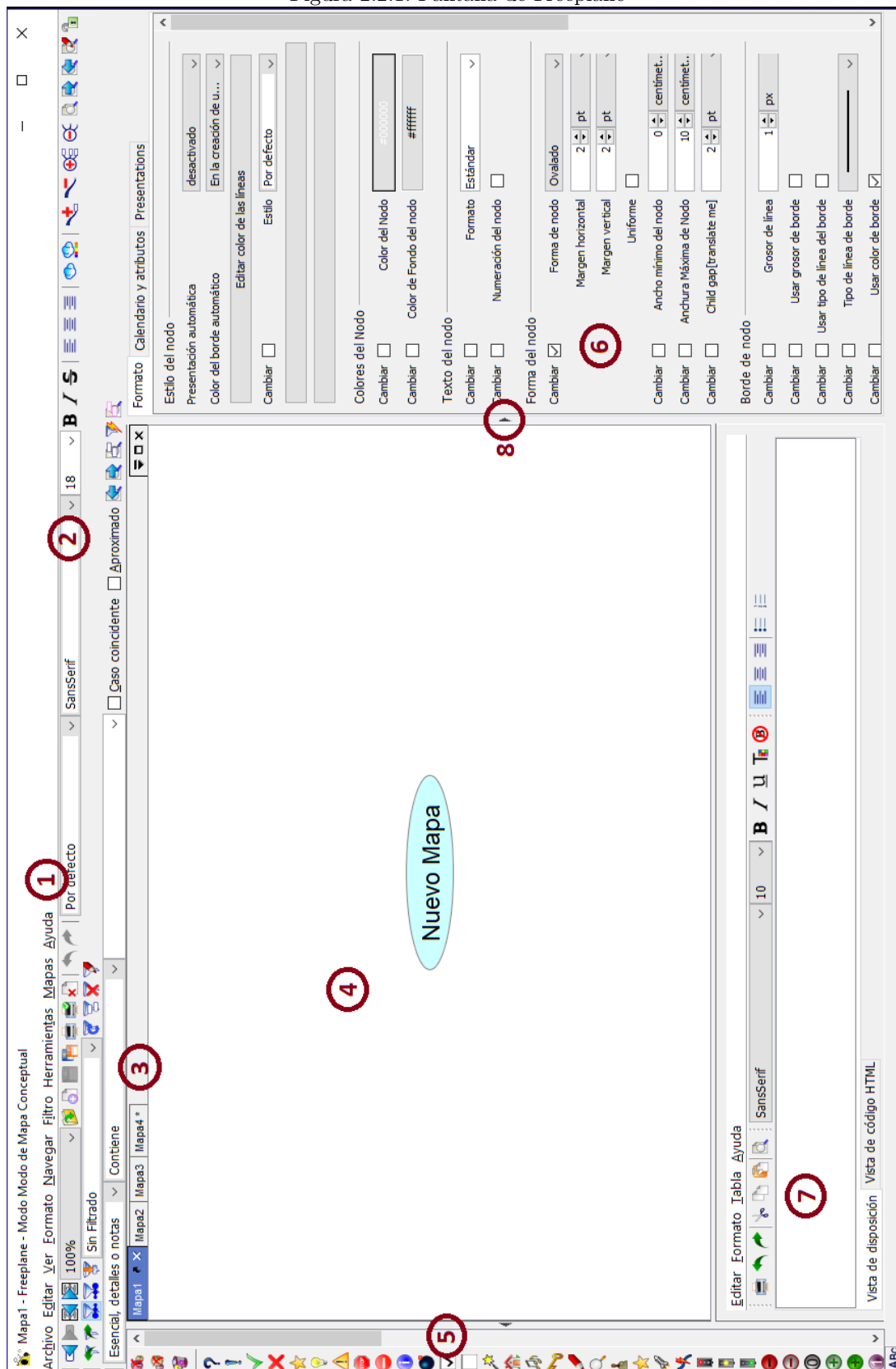
2.2. La ventana de la aplicación

Cuando arrancamos Freeplane nos aparece el área de trabajo con un mapa conteniendo únicamente el nodo raíz. Pinchando la tecla  insert añadiremos nodos a este primero. Podemos continuar moviéndonos por los nuevos nodos con las flechas de desplazamiento y añadiendo otros nuevos. Otras características de freeplane es que podemos hacer zoom a los mapas o buscar un concepto entre todos los mapas pinchando el icono del lápiz de la barra superior, que es el penúltimo. Una vez tenemos la estructura del mapa podremos darle formato a cada nodo y añadir iconos, figuras y otros elementos.

Las partes de la ventana de freeplane son las siguientes (ver figura 2.2.1):

- Barra de menús en la parte superior (1).
- Barra de herramientas con iconos para guardar, abrir, estilo del nodo, fuente... (2).
- Barra con los mapas abiertos (3).
- Espacio central para dibujar el mapa (4).
- Barra de iconos en el lateral izquierdo (5).
- Panel de formatos, a la derecha (6).
- Panel de notas que podemos posicionarlo de forma flotante, a la derecha o en la parte inferior (7).
- Flecha que nos sirve para mostrar u ocultar el panel de formato lateral (8).

Figura 2.2.1: Pantalla de Freeplane



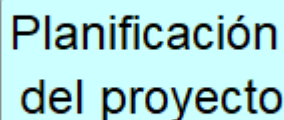
Así es
Freeplane

2.3. Creando el primer mapa

Cuando lanzamos la aplicación nos aparece con un mapa vacío que contiene únicamente el nodo raíz seleccionado. Vamos a cambiarle el texto. Para ello únicamente tenemos que escribir lo que deseamos, en nuestro caso “Planificación del proyecto”.

Primer paso

Figura 2.3.1: Primer paso



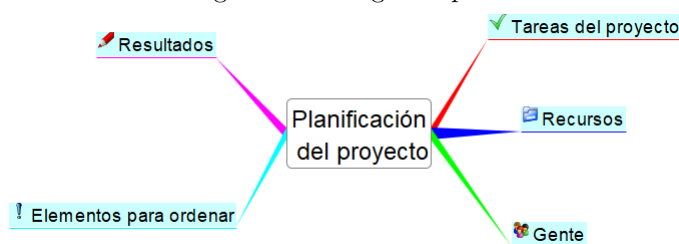
A partir de este primer nodo podemos comenzar a crear nuestro mapa. Cada vez que pulsemos la tecla **INSERT** se añadirá una nueva rama. Se añaden alternativamente a la derecha y a la izquierda. Pulsando esta tecla 5 veces crearemos las ramas que necesitamos. Las seleccionaremos una a una con el ratón cambiando el texto.

Segundo paso

Luego añadiremos algunos iconos a las ramas. Para ello mostraremos la barra de iconos que es elemento (5) en la figura 2.2.1. Si no aparece visible podéis mostrarla pinchando en Ver / Controles / (Des)activar barra de herramientas izquierda. Ahora ya podremos añadir cuantos iconos queramos a los nodos. Para ello pincharemos en el nodo que deseemos y luego en el icono elegido que aparecerá inmediatamente dibujado a la izquierda del texto.

Otra cosa que podemos hacer es cambiar el color del nodo y el tipo de borde con el panel de formato de la derecha (elemento 6 en la figura 2.2.1). Las opciones que tendremos que cambiar son las del apartado colores del nodo, forma del nodo y borde del nodo en dicho panel.

Figura 2.3.2: Segundo paso



Lo siguiente es crear las tareas del proyecto añadiendo 4 nuevos nodos a partir de “Tareas del proyecto”. En estas tareas crearemos iconos de progreso pinchando en el menú Editar / Iconos / Progreso. La tarea 1 contiene un progreso aumentado mientras que el resto un progreso normal que añadiremos pinchando en Editar / Iconos / progreso / aumentar progreso. Además, hay un nodo sumario en las tareas 2 a 4 que se crea seleccionándolas y pinchando en **ALT** + **MAYUSCULAS** + **INSERT** o con el menú Editar / Nuevo nodo / Nuevo nodo principal (nodos seleccionados).

Tercer paso

En el nodo gente añadiremos a nuestros colaboradores con iconos en sus nodos. Además, hemos copiado las tareas a las que se encargarán pegándolas a partir de ellos como clones: (**CTRL** + **C**) para

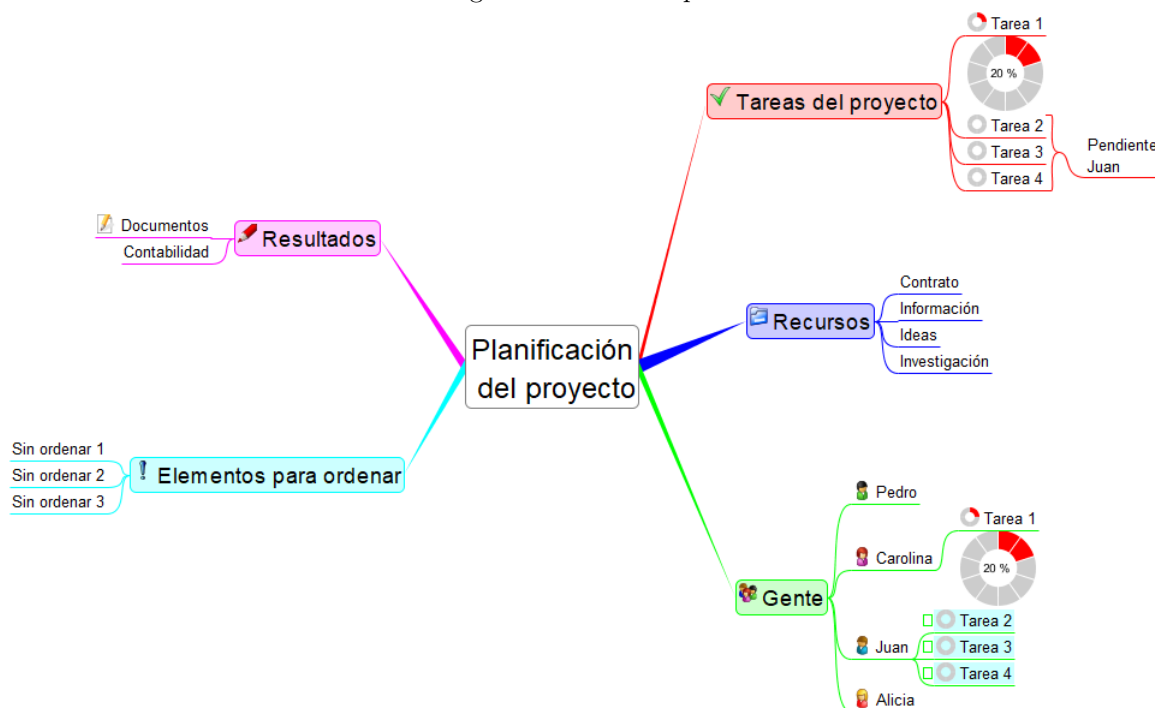
copiar la tarea y **CTRL** + **D** para pegarla como clon). Los nodos clones cambian simultáneamente al modificar uno de ellos.

El resto son nodos normales que hemos añadido pinchando en **INSERT**.

Una vez finalizado obtenemos la siguiente fase de nuestro mapa en la que podemos ver:

- Hay algunos iconos, como en los nodos hijo de la raíz.
- El tipo de los nodos raíz y sus hijos es “burbuja”
- Hemos cambiado sus colores
- Hemos añadido unos iconos de progreso normales en las tareas 2, 3 y 4 (Editar / Iconos / Progreso / Subir nivel)
- Hemos añadido un icono de progreso mayor en la Tarea 1 (Editar / Iconos / Progreso / Progreso aumentado 10 %)
- Hemos subido el progreso de la Tarea 2 un par de veces haciendo doble click sobre el círculo de progreso (**CTRL** + doble click para bajar progreso).
- En cada una de las personas hemos añadido clones a las tareas de las que se encargan. De esta manera si cambiamos algo en la Tarea 1 de Carolina se actualiza automáticamente en las Tareas del proyecto.

Figura 2.3.3: Tercer paso



2.4. Uso del mapa

Ya tenemos la versión inicial de nuestro mapa. Pinchamos en Archivo / Guardar para evitar perder el trabajo.

Ahora podremos continuar rellenandolo. Seguramente tendremos que añadir nuevas tareas e incluso

Guardando
el mapa

Añadiendo
nodos

podremos contratar a un nuevo colaborador para realizarlas. Esto, como hemos visto, lo conseguimos pinchando en el nodo y luego en la tecla **INSERT**.

También podemos aumentar el porcentaje de finalización de cada tarea haciendo doble click encima del icono de progreso expandido.

Otra cosa que podemos hacer es mover un nodo. Por ejemplo, para cambiar una tarea de persona sólo tenemos que pinchar encima y arrastrarla hacia la persona elegida. También podemos colocar los nodos hijo de “Elementos para ordenar” en sus ubicaciones definitivas de la misma forma: arrastrándolas al nodo que corresponda. A la hora de soltar el nodo que queremos mover en su nueva ubicación, Freeplane nos da dos opciones: si lo soltamos en el borde externo del nodo lo pondremos como hijo suyo. sin embargo, si lo hacemos en su borde superior será su hermano y se colocará encima de él.

Si vemos que el mapa es muy pequeño podemos hacer zoom con el desplegable de la barra superior de herramientas ó con la ruleta del ratón si mantenemos pulsada la tecla **CTRL**.

Si alguna de los nodos tiene una rama demasiado extensa podremos colapsarla. Para ello observaremos los pequeños circulitos que aparecen al final del nodo al pasar con el ratón sobre él. En la figura 2.4.1 vemos una rama extensa expandida con el circulito para poder colapsarla que ha aparecido al poner el ratón sobre él.

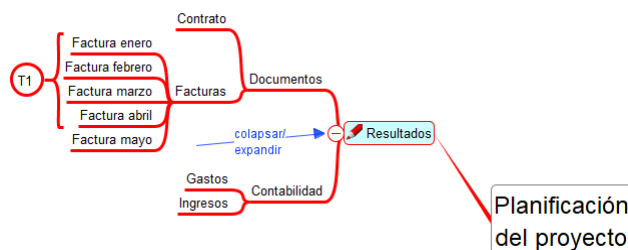
Icono de progreso

Mover nodos

Zoom

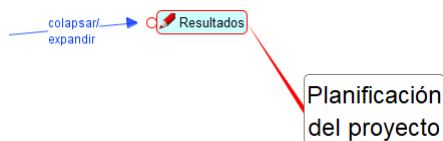
Expandir Colapsar

Figura 2.4.1: Mapa expandido



Pinchando sobre ese nodo esa rama se cerrará, apareciendo ahora el mapa de la figura 2.4.2. Vemos que en una rama cerrada aparece continuamente el circulito que nos servirá para abrirla sin necesidad de tener que poner el ratón encima del nodo.

Figura 2.4.2: Mapa colapsado



Si aún así es demasiado extenso podemos abrir otra vista del mismo mapa con el menú Ver / Nueva Vista del mapa. Si arrastramos ahora la nueva pesaña que ha aparecido a uno de los bordes podremos ver varias zonas del mapa simultaneamente.

Y si queremos presentar el proyecto a nuestros colaboradores podremos poner el mapa en pantalla completa con **F11** y usar las opciones de presentación básicas accesibles en el menú Ver / Ver configuración y luego las opciones Spotlight y Centrar el nodo.

Cuando hayamos terminado ya podemos guardar el mapa mediante el menú Archivo, Guardar. En resumen, podemos realizar las siguientes actividades:

- Seleccionar nodos: Freeplane está configurado para que si nos detenemos brevemente en un nodo la aplicación lo selecciona. Este comportamiento podemos cambiarlo en las Preferencias, en la pestaña de comportamiento si modificamos la opción “Método de selección”. También podemos elegir múltiples nodos haciendo click encima de ellos mientras tenemos pulsada la tecla **CTRL**.

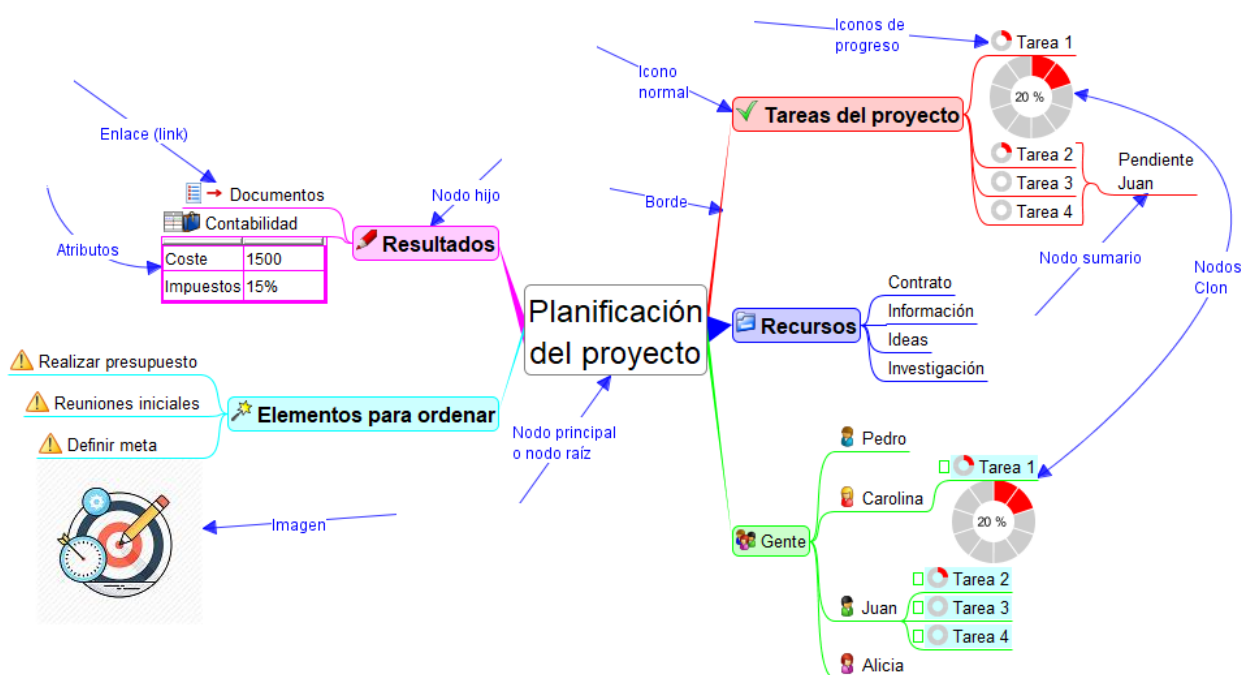
Nueva vista de mapa

Guardando el mapa

- Expandir y colapsar nodos. Para ello pincharemos en unos circulitos que hay al final de cada nodo. (ver figuras 2.4.1 y 2.4.2).
- Buscar un nodo que contenga una cadena concreta. Para ello pincharemos en Editar / Buscar / Buscar.
- Ver el mapa en una escala determinada. Podemos hacer zoom con la primera lista desplegable que aparece en la barra de herramientas, elemento (2) en la figura 2.2.1.
- Poner Freeplane en pantalla completa pulsando la tecla **F11**.
- Ver varios mapas a la vez. Para ello moveremos la pestaña del mapas que queramos a uno de los bordes de la aplicación.
- Ver secciones distintas del mismo mapa en dos ventanas de Freeplane. Podemos abrir otra pestaña con el mismo mapa pinchando en Ver / Nueva vista de mapa y luego mover dicha pestaña a uno de los bordes del area de trabajo de Freeplane.
- Ver un esquema del mapa si vamos al menú Ver / Ver configuración / Vista en esquema.

2.5. Elementos de un mapa

Figura 2.5.1: Elementos de un mapa



Elementos de un mapa

Un mapa no solo está formado por las ideas que vamos añadiendo (nodos) y por los enlaces a estas ideas. Freeplane dispone de muchos más elementos que aparecen en la figura 2.5.1 marcados con flechas azules. Son los siguientes:

Nodo principal: Es el nodo raíz y está situado en el centro. En la figura anterior es el nodo "Planificación del proyecto".

Nodos hijos: Parten de un nodo situado ya en el mapa. Podemos expandirlos y colapsarlos.

Nodo sumario: Agrupa varios nodos hermanos y sirve para mostrar un resumen. En el mapa anterior, es el nodo “Pendiente Juan”, a la derecha de las tareas.

Nodo clon: Un nodo que aparece en varias ramas del mapa. Si lo modificamos en una de ellas queda modificado en las demás. Se crean copiando el nodo y pegándolo mediante la combinación de teclas **CTRL** + **D** ó pinchando en el menú Editar / Pegar clon. Podemos identificarlo porque al seleccionarlo con el ratón aparece un pequeño cuadro a su izquierda. Son los nodos Tarea1 a Tarea 4. Vemos en los nodos hijo de “Gente” esos cuadritos, a la izquierda de los iconos de progreso de las tareas.

Bordes: Líneas que unen el nodo padre con los nodos hijo.

Enlaces: Apuntan a información que puede ser del mapa, de otros mapas ó externa a freeplane. Al pinchar sobre ellos nos remite a dicha información a la que se refieren. En la figura aparece como una pequeña flechita a la izquierda del nodo “Documentos”, después del icono.

Conectores: Líneas de unión que se ponen entre nodos arbitrarios del mapa.

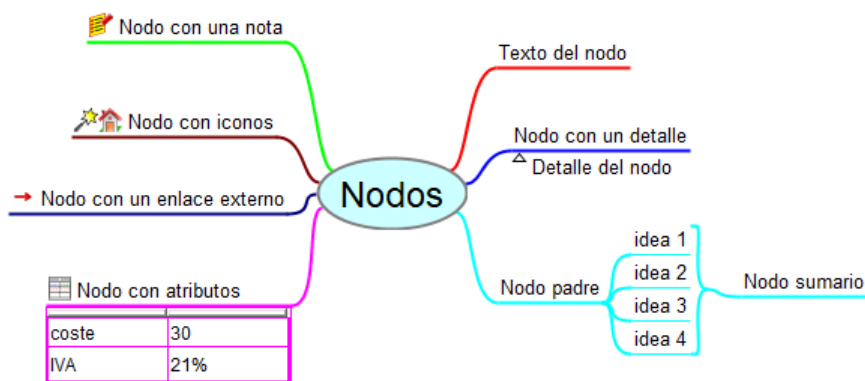
Atributos: Elementos bajo los nodos que contienen datos numéricos ó textuales. Son propiedades que podemos añadir al nodo si pinchamos con el botón derecho del ratón y elegimos Editar atributos. Están compuestos por un nombre y un valor. En el nodo Contabilidad son “Coste” e “Impuestos”.

Iconos: Hay iconos normales que podemos añadir con la barra de iconos de la izquierda (Freeplane viene con un juego estándar de iconos que se puede ampliar) ó iconos de progreso.

Imágenes: También podemos añadir imágenes en los nodos.

2.6. Elementos de un nodo

Figura 2.6.1: Elementos de los nodos



Hasta ahora hemos visto a cada una de las ideas como un nodo que contiene un texto. Sin embargo un nodo es mucho más. En la figura 2.6.1 se ven los elementos básicos de los nodos.:

Elementos de un nodo

Texto del nodo: Es el elemento básico y fundamental del nodo y el que vemos en el mapa. Habitualmente es texto simple pero podemos añadir código HTML si lo editamos pulsando **ALT** + **INTRO**. Aparecerá un editor con dos pestañas abajo: Vista de disposición (para incluir texto normal) y Vista de código HTML para incluir más formatos.

Detalles del nodo: Texto, figuras y otros elementos que aparecen en el nodo. También podemos añadir elementos HTML.

Iconos: Iconos que añadimos al nodo desde la barra lateral.

Nota: Normalmente un nodo consta de un texto corto. Si deseamos añadir explicaciones extensas tendremos que añadir una nota al nodo. Podemos mostrar la ventana de notas si pinchamos con el botón dercho en el fondo del mapa y elegimos del menú Mostrar/Esconder ventana de notas. Sólo se mostrará si activamos el panel de notas o si ponemos el cursor sobre el nodo.

Atributos: propiedades que añadimos bajo los detalles del nodo.

Enlaces: Enlaces que se ponen en un nodo y que apuntan o bien a otros nodos del mapa o a recursos externos, archivos o páginas web

Capítulo 3

Personalizando nuestro mapa

3.1. El panel de formato

El panel lateral que aparece en la parte derecha de Freeplane nos va a servir para cambiar el aspecto del mapa. Aparece al pulsar las teclas **ALT** + **P** ó pinchando en la pequeña flecha del borde derecho (Es el elemento (6) en la figura 2.2.1) y está dividido en secciones:

El panel de formato

- Estilo del nodo. Veremos los estilos más adelante.
- Colores del nodo. Podemos modificar el color del texto y del fondo del nodo .
- Texto del nodo. Tipo de nodo. Normalmente es Estandar pero como veremos más adelante un nodo puede contener una fórmula, código latex o un numero con o sin decimales.
- Forma del nodo. Aparte de la habitual, la forma del nodo puede ser ovalada, rectangular, hexagonal...
- Borde del nodo. Anchura, color y tipo de línea del nodo
- Fuente del nodo. Tipo de letra, tamaño, cursiva, negrita...
- Iconos. Tamaño de los iconos.
- Nubes. Forma y color de la nube (ver más adelante).

3.2. Modificando el texto del nodo

Tipo y tamaño de la letra, negrita, cursiva, tachado

Figura 3.2.1: Modificando el texto del nodo

Fuente del Nodo	
Cambiar <input type="checkbox"/>	Fuente del Nodo Georgia
Cambiar <input type="checkbox"/>	Fuente del Nodo 10
Cambiar <input type="checkbox"/>	Fuente Negrita <input type="checkbox"/>
Cambiar <input type="checkbox"/>	Strike through[translate me] <input type="checkbox"/>
Cambiar <input type="checkbox"/>	Fuente Cursiva <input type="checkbox"/>
Cambiar <input type="checkbox"/>	Alineamiento de texto Default[translate me]
Cambiar <input type="checkbox"/>	Hiperenlace <input type="checkbox"/>

A la hora de personalizar el mapa es importante no solo tener visible la barra superior que aparece bajo los menús sino también el panel de formato que aparece a la derecha que es el que usaremos casi siempre. Dentro de ese panel hay una sección, "Fuente del nodo" que vemos en la figura 3.2.1. Como vemos podemos cambiar la fuente, el tamaño del texto y ponerlo en negrita o cursiva aparte de centrarlo en el nodo o alinearlo a izquierda o derecha.

Tipo de nodo:
texto normal,
fórmula,
número de
decimales,
forma de
la fecha...

Figura 3.2.2: Modificando el texto del nodo

A medida que vayamos construyendo más mapas tendremos necesidad de representar no solo texto sino también números o incluso añadir fechas. Aquí es donde interviene la sección "Texto del nodo" del panel lateral (ver figura 3.2.2) donde escogeremos el formato del número o de la fecha en el desplegable "Formato". Algunas de las opciones más usadas son:

Formatos:
texto,
numéricos
y de fecha

- Texto: el contenido del nodo se muestra como texto normal
- #0.####: Es un número decimal pero se mostrarán sólo 4 decimales (se redondeará).
- #.00: Número decimal del que se redondeará a 2 decimales
- #: Sólo se mostrará la parte entera.
- #.## %: Es un porcentaje. Se mostrará el número multiplicado por 100.
- d/MM/yy: Es una fecha. Se mostrará el día, mes y año como números separados por la barra inclinada.
- dd-MMM-yyyy: Con este formato el mes se mostrará como las tres primeras letras de su nombre.
- HH:mm: Es una hora y se mostrará separada de los minutos con el símbolo de los dos puntos.

Además, en esa misma sección podemos marcar la casilla de numeración del nodo. Todos los nodos que dependen del mismo padre serán numerados de forma secuencial.

3.3. Aspecto de los nodos

Forma
y ancho
del nodo

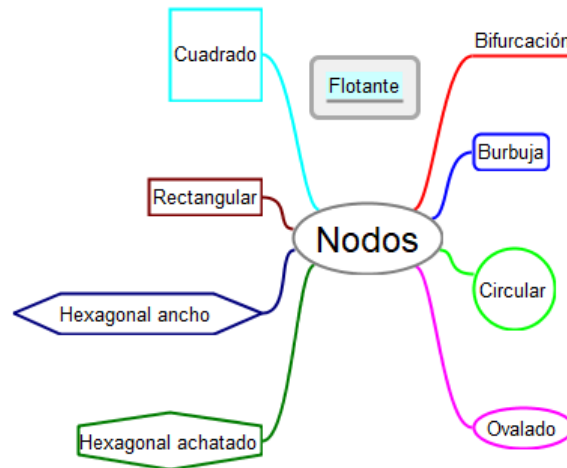
Figura 3.3.1: Modificando la forma del nodo

Pasamos ahora a modificar el aspecto que tienen los nodos. Si miramos el apartado “Forma del nodo” que aparece reproducido en la figura 3.3.1 vemos que tenemos opción de modificar la forma del nodo así como sus márgenes.

También aquí indicaremos a freeplane qué ancho mínimo y máximo tiene el nodo. Esto nos va a servir para obligar a los nodos del mapa sean del mismo ancho. Otra opción importante es indicar a Freeplane qué espacio queremos que haya entre los nodos hermanos (huevo en hijo vertical).

En la figura 3.3.2 aparecen todas las formas que podemos dar a los nodos:

Figura 3.3.2: Formas disponibles para los nodos



Formas de los nodos

Bifurcación: Es la habitual. Consiste únicamente en una línea debajo del nodo.

Burbuja: Se trata de un recuadro con los bordes redondeados.

Rectángulo: Damos al nodo forma rectangular. Se convierte en un cuadrado marcando más abajo la opción de “Uniforme”.

Oval: Forma ovalada. Puede convertirse en un círculo marcando más abajo la opción de “Uniforme”.

Wide hexagonal: Hexágono apoyado en una de sus caras.

Narrow hexagonal: Hexágono apoyado en uno de sus vértices.

Si marcamos “Uniform” convertimos a nuestra forma en una forma regular (todos los lados iguales).

Justo debajo de la forma del nodo hay otra sección interesante, “Borde del nodo” (ver figura 3.3.3). Se refiere a las líneas utilizadas para dibujar la forma de los nodos. Las más importantes son:

- Grosor de la línea
- Tipo de línea: punteada, con rayas, continua...
- Color del borde del nodo

Grosor, color y tipo de línea para los bordes

Mucho cuidado de no confundir el borde del nodo, que es la línea que sirve para dibujar la forma del nodo con el borde que une un nodo padre con sus nodos hijos. La primera se cambia en la sección “Borde del nodo” y la segunda en la sección “Borde”. Podemos tener un nodo ovalado con la línea punteada pero cuya línea de unión con sus nodos hijos sea continua. Además, el óvalo del nodo podría ser azul y la línea de unión con sus nodos hijos roja.

Figura 3.3.3: Opciones de los bordes

Borde de nodo

Cambiar <input type="checkbox"/>	Grosor de línea	<input type="text" value="1"/> px
Cambiar <input type="checkbox"/>	Usar grosor de borde	<input type="checkbox"/>
Cambiar <input type="checkbox"/>	Usar tipo de línea del borde	<input type="checkbox"/>
Cambiar <input type="checkbox"/>	Tipo de línea de borde	<input type="text" value="—"/> v
Cambiar <input type="checkbox"/>	Usar color de borde	<input checked="" type="checkbox"/>
Cambiar <input type="checkbox"/>	Color	<input type="text" value="#808080"/>

3.4. Iconos

En ese apartado modificamos el tamaño con el que se dibujará el icono en cada nodo. Únicamente contiene una propiedad: “Tamaño del icono” con el que modificaremos el tamaño con el que se dibujará el icono del nodo.

3.5. Nubes

Forma y color
de las nubes

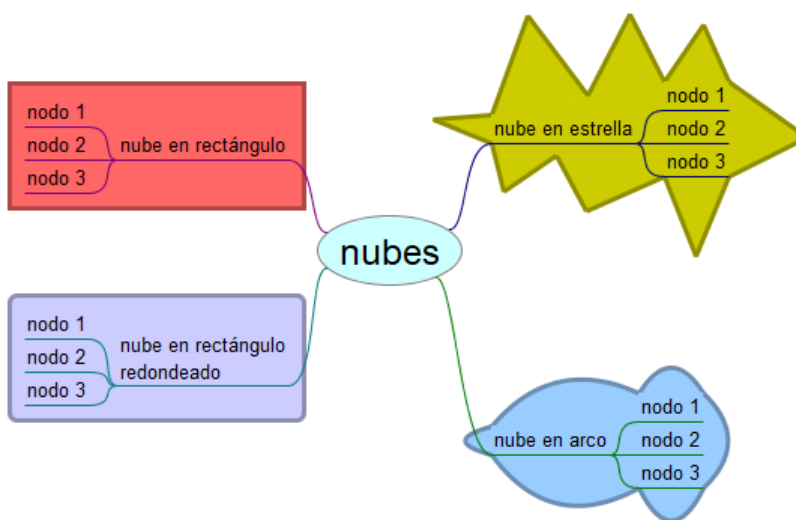


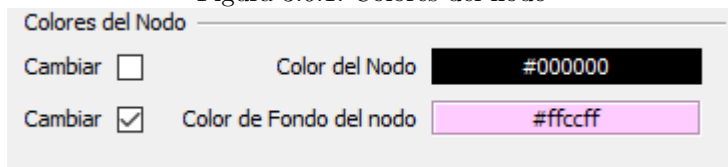
Figura 3.5.1: Tipos de nubes

La última sección del panel se refiere a las nubes cuyo aspecto vemos en la figura 3.5.1. Son marcos de diferentes formas alrededor de un grupo de nodos. Dichos marcos pueden ser rectangulares pero también de formas irregulares rectas o redondeadas. Esto le da un atractivo aspecto al mapa.

3.6. Colores

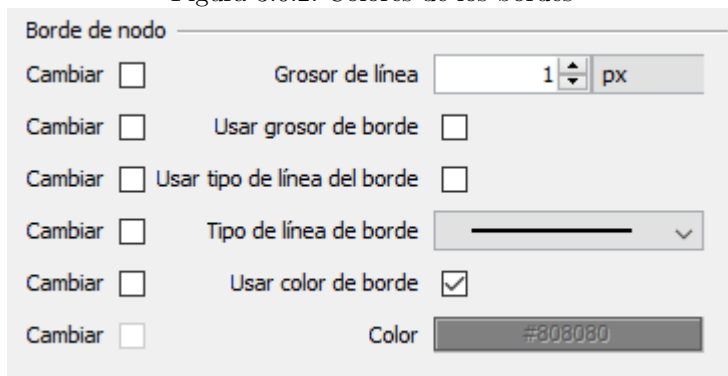
En el apartado “Colores del nodo” podemos modificar el color tanto del texto como del fondo del nodo. Sin embargo, además de estos dos elementos, también podemos definir el color de los enlaces entre

Figura 3.6.1: Colores del nodo



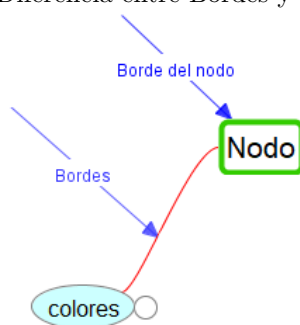
nodos, para lo cual nos iremos a los apartados “Borde del nodo” y “Bordes”. Estos apartados se muestran en las figuras 3.6.1, 3.6.2 y 3.7.1.

Figura 3.6.2: Colores de los bordes



La diferencia entre los bordes y los bordes del nodo se muestra en la figura 3.6.3. Veamos esa imagen: si queremos cambiar los bordes utilizamos la sección “Bordes” que se muestra en la figura 3.7.1. Si por el contrario lo que queremos es cambiar el borde del nodo utilizaremos la sección del panel de la figura 3.6.2. Para cambiar el color de la letra o del fondo del nodo se usa la sección del panel mostrada en la figura 3.6.1.

Figura 3.6.3: Diferencia entre Bordes y Borde del nodo



3.7. Aspecto de los bordes

Freeplane usa la palabra borde para denominar dos elementos: para las líneas que unen los nodos hijos con su nodo padre y para la línea que dibuja la forma del nodo. En esta sección vamos ahora a modificar las líneas que unen los nodos del mapa. En la figura 3.7.1 tenemos las opciones que nos da Freeplane:

Figura 3.7.1: Opciones de los enlaces entre nodos

Bordes	
Cambiar <input type="checkbox"/>	Anchura del borde <input type="text" value="0"/>
Cambiar <input type="checkbox"/>	Tipo de línea de filo <input type="text" value="—"/>
Cambiar <input type="checkbox"/>	Estilo del borde <input type="text" value="Bezier"/>
Cambiar <input type="checkbox"/>	Color del Borde <input type="text" value="#808080"/>

- Anchura del borde.
- Tipo de línea: como antes, continua, punteada, etc.
- Color del borde.
- Estilo del borde: es la opción más interesante.

Hay varios estilos de bordes que podemos ver en la figura 3.7.2:

Tipos de
bordes

Lineal. Conexión lineal indirecta uniforme. Su grosor es el mismo en toda su trayectoria.

Lineal con ángulos. Conexión lineal de grosor variable, mas grueso al principio y más fino al final

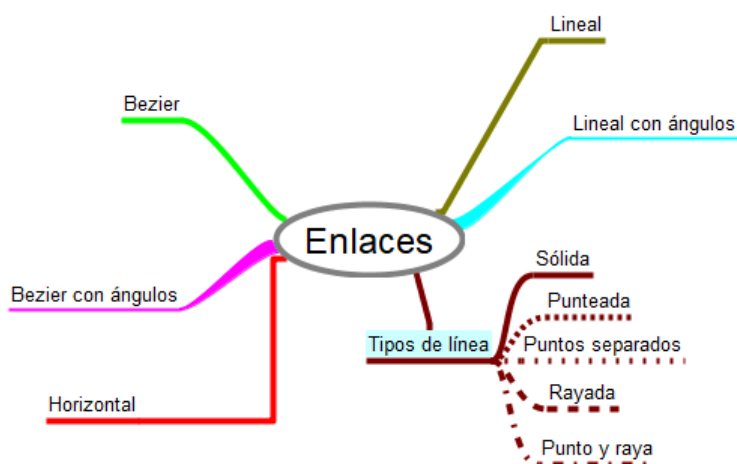
Horizontal. Conexión con el nodo con líneas horizontales y verticales. Grosor uniforme.

Bezier. Es una línea curva de grosor uniforme.

Bezier con ángulos. Línea curva de grosor variable. Más grueso al principio y va disminuyendo al final

Ocultar el borde: no se muestra la línea de unión entre el padre y nuestro nodo.

Figura 3.7.2: Tipos de conexión entre nodos (bordes)

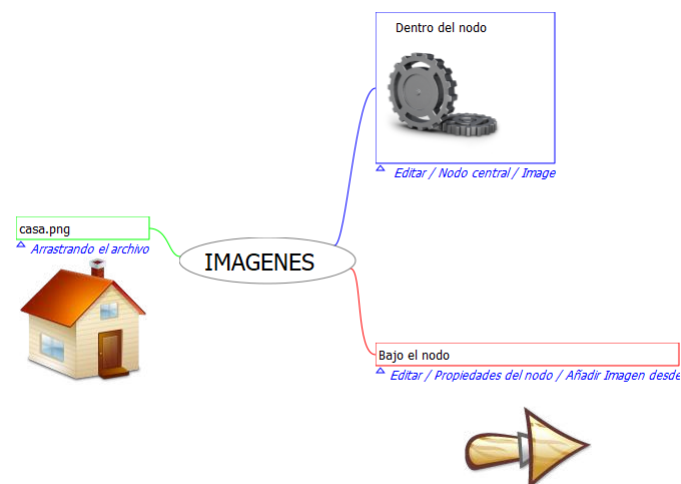


Capítulo 4

Añadiendo otros elementos

4.1. Imágenes

Figura 4.1.1: Añadir imágenes a los nodos



Ubicaciones
para las
imágenes

Además de texto, iconos y atributos en nuestros mapas también pueden aparecer imágenes. Nuestras imágenes pueden aparecer en diferentes ubicaciones:

- Dentro del nodo.
- Bajo el nodo.
- Imagen de fondo del mapa.

En la figura 4.1.1 están estas opciones.

4.1.1. Imagen dentro de un nodo

Como hemos visto un nodo es una forma (un cuadrado, óvalo, hexágono) dentro del cual hay un texto y que contiene elementos extra tales como texto bajo él, atributos y enlaces. Sin embargo también podemos hacer que dentro de la forma del nodo aparezca una imagen e incluso una imagen bajo la cual podremos escribir un texto.

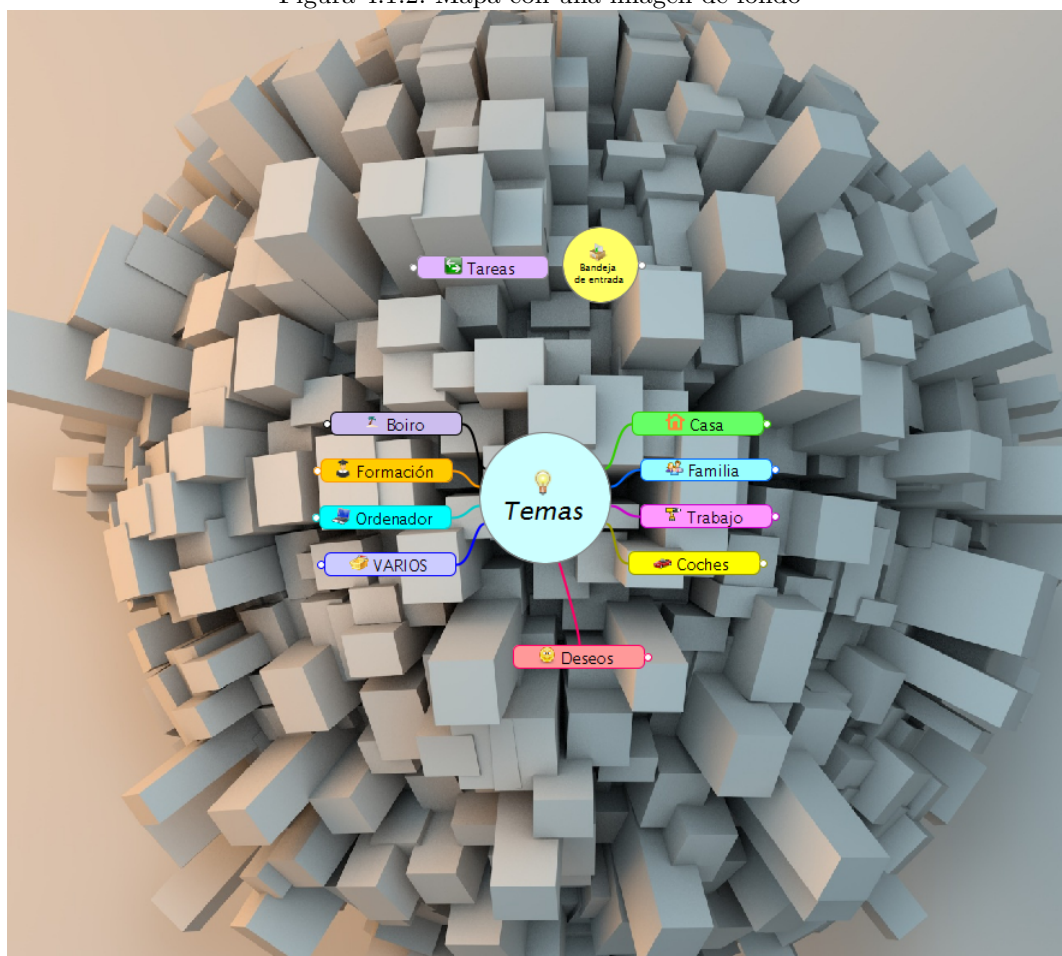
Añadir una imagen dentro de un nodo es sencillo. Únicamente hay que pinchar en el menú Editar / Nodo central / Imagen y elegir el fichero de la imagen. Una vez aparezca en el mapa ya podremos hacer doble click en ella y añadirle texto que aparecerá a su lado. Si queremos que aparezca bajo la imagen podemos teclear **ALT** **ENTER** lo cual insertará una nueva línea.

4.1.2. Imagen bajo el nodo

Es una tarea muy sencilla. Únicamente tenemos que arrastrar la imagen desde el sistema operativo hasta el área de trabajo de Freeplane y ya está. Sin embargo si ya tenemos el nodo en el mapa podemos ir al menú Editar / Propiedades del nodo / Añadir Imagen desde. En ambos casos la imagen aparecerá bajo el texto del nodo.

4.1.3. Imagen de fondo del mapa

Figura 4.1.2: Mapa con una imagen de fondo

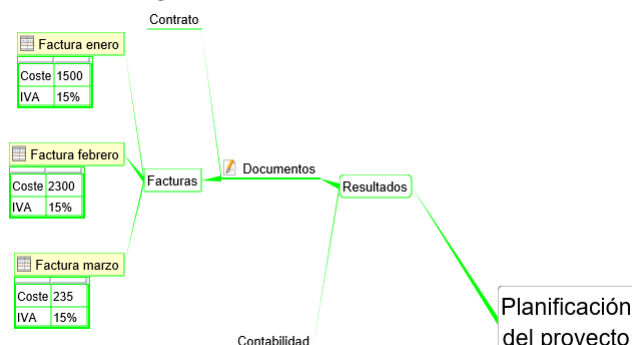


Algo muy interesante es la opción de cambiar el aburrido fondo blanco de todos los mapas. Si nos vamos a Formato / Mapa de fondo / Fondo del mapa podemos elegir un color personalizado para el mapa.

Sin embargo lo mejor es que también podemos añadir una imagen que se mostrará tras el mapa (ver imagen 4.1.2). Esto se consigue desde el menú Formato / Mapa de fondo / Imagen de fondo.

4.2. Atributos

Figura 4.2.1: Añadiendo atributos



Hemos visto que un nodo tiene un texto, una forma, color, tipo de línea e incluso un detalle bajo él. Ahora veremos los atributos que podemos verlos como propiedades que añadimos a cada nodo. Propiedades que definimos como un nombre de propiedad al que aplicamos un valor. Por ejemplo, en las facturas serían atributos “Coste” e “IVA”. Tendríamos una factura con una propiedad llamada “Coste” de valor 1.500, otra propiedad llamada “IVA” con el valor 0.15 etc. Si los nodos fueran vehículos podríamos añadir los atributos “Precio”, “Potencia” y “Consumo”. Mediante los atributos ampliamos la utilidad de un nodo con propiedades personalizadas.

Si vamos al mapa que comenzamos en el capítulo 2 veremos que hay un apartado para incluir los documentos del proyecto. Unos documentos fundamentales de todo proyecto son las facturas. Para almacenarlas añadiremos un nodo Facturas hijo del nodo “Documentos”. En este nodo incluiremos todas las facturas del proyecto, tal y como vemos en la figura 4.2.1. Si deseamos que Freeplane pueda acceder a los detalles de las facturas tales como el coste, el IVA, etc lo más adecuado es añadir dos atributos. Uno para el coste y otro para el IVA. Para añadir estos atributos nos iríamos al panel de formato pero eligiendo la pestaña “Calendario y atributos” de la parte superior. En la sección inferior hay dos botones, “Nuevo atributo” y “Anchura óptima”. Pinchando en el primero podemos añadir los atributos que deseemos. Y pinchando en el segundo modifica su ancho para ajustarlo exactamente a su contenido. Hemos puesto como IVA el valor 0.15 escogiendo el formato #.## % para que lo muestre como porcentaje.

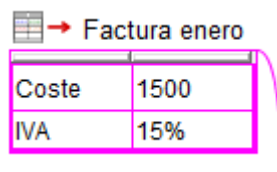
Para trabajar con atributos, Freeplane proporciona varias herramientas que veremos más adelante en la sección para usuarios avanzados. Por ejemplo, existe un gestor de atributos accesible desde el menú Editar, Propiedades del nodo, Gestor de atributos mediante el cual podemos restringir los valores que puede contener un atributo, entre otras cosas.

4.3. Enlaces

Imaginemos que además de tener almacenados los detalles de las facturas también quisiéramos acceder a los documentos originales al hacer click encima del nodo de la factura. Esto se consigue mediante los enlaces, que funcionan de forma similar a los de cualquier página web: pinchando encima Freeplane nos lleva al destino del enlace. Esto se realiza pinchando con el botón derecho sobre el nodo y eligiendo del menú contextual Enlaces, Añadir enlaces (Seleccionar). Aparecerá un selector de archivos para elegir la factura correspondiente.

Otra forma de añadir enlaces a archivos arrastrar el documento sobre uno de los nodos del mapa como hacíamos con las imágenes. Aparecerá como texto el nombre del fichero y se creará un enlace para que

Figura 4.3.1: Nodo con enlace



podamos acceder a él haciendo click en él. El enlace tiene la forma de una pequeña flecha roja. Hay dos tipos de enlace: a documentos o lugares externos o a otros nodos.

Al añadir un enlace a un documento aparecerá una pequeña flecha roja al principio del nodo tal y como vemos en la figura 4.3.1.

4.3.1. Enlaces internos

Son enlaces entre nodos. Al pinchar encima del nodo con enlace Freeplane nos llevará al nodo enlazado que puede estar en el mapa actual o en otro mapa distinto. Para definirlos pinchamos en el nodo al que queremos añadir el enlace y después el nodo enlazado. Luego del menú contextual que aparece al pinchar este segundo nodo con el botón derecho elegiremos la opción Enlaces / Agregar enlace local ó Editar / Enlaces / Agregar enlace local.

Si el segundo nodo está en otro mapa el proceso es algo diferente:

1. Iremos al nodo destino y pincharemos con el botón derecho del ratón.
2. Elegiremos la opción Copiar la URL del nodo.
3. Iremos al nodo donde queremos poner el enlace y elegiremos del menú contextual Enlace / Campo de texto.
4. Pegaremos la URL antes copiada.

Otra forma sería:

1. Nos posicionaremos en el nodo destino y luego en Editar / Enlaces / Definir enlace a la etiqueta del ancla. Esto lo fija como un ancla.
2. Nos ponemos en el segundo nodo y elegimos Editar / Enlaces / Convertir enlace a etiqueta del ancla.
3. Nota: si elegimos por el contrario Editar / Enlaces / Crear enlace de una etiqueta de ancla el enlace se hará al revés: el ancla tendrá el enlace y el destino será el enlazado.

4.3.2. Enlaces externos

Como ya he dicho la forma más sencilla es arrastrar un archivo a uno de los nodos de la ventana de Freeplane. Por ejemplo, si arrastramos la dirección que aparece en el navegador a un nodo se creará un enlace hacia esa página web. Pinchando encima se abrirá el navegador y podremos ver esa página.

Otra forma para añadir un enlace externo es elegir el menú Editar / Enlaces / Elegir archivo. Con eso se establecerá un enlace que nos llevará al archivo antes seleccionado. o también podemos elegir Editar / Enlaces / Campo de texto que nos permite teclear lo que deseemos, por ejemplo, una dirección web.

4.3.3. Enviar correo

Entrando en Editar / Enlace / Enlace (campo de texto) y tecleando mailto:kewapo@gmail.com al pinchar en el nodo se abrirá el editor de correo electrónico para que enviemos ese correo.

4.3.4. Enlace a un menú de freeplane

Podemos hacer que uno de nuestros nodos se comporte de forma idéntica a si pinchamos en uno de sus menús. Para ello pinchamos en el nodo y luego en Editar / Enlaces / Hiperenlace (entrada de menú). Saldrá una lista con los menús de la aplicación para que la elijamos.

4.3.5. Enlace a un comando del sistema operativo

Otra opción es pinchar en Edición / Enlaces / Ejecutar un comando del S.O. Aparecerá un cuadrito para que tecleemos el comando (por ejemplo, si ponemos notepad se abrirá el block de notas de Windows)

4.4. Fórmulas

Crea un nodo
y escribe:
= 2 + 3

Figura 4.4.1: Fórmula



Un nodo también puede contener una fórmula. Mediante ellas podemos realizar cálculos basándonos en los elementos de otros nodos. Es una expresión que calcula Freeplane de forma dinámica. Puede incluir referencias a otros nodos, operadores aritméticos y funciones matemáticas. Podemos identificar un nodo con fórmula porque Freeplane pone un marco de color verde claro alrededor de él. En la figura 4.4.1 aparece un nodo en el cual hemos escrito “= 2 + 3”. Vemos que aparece un 5.

Para añadir una fórmula a un nodo únicamente tenemos que comenzar a escribir con el símbolo de igual. Por ejemplo, si en el texto de un nodo determinado escribimos = 2 * 3 aparecerá un 6. Otros ejemplos son:

- = 2 + 3: aparecerá el número 5.
- = (3 * 4) + “ veces”: aparecerá el texto “6 veces”
- =(parent.value=="calcular") ? 2 + 3 : "2 + 3": Si el padre del nodo contiene el texto “calcular” se mostrará un 5 en el nodo. Si es otra cosa, se mostrará “2 + 3”.

Las fórmulas de freeplane son un recurso muy potente. El último ejemplo es una expresión condicional: **expresión_booleana ? resultado_si_verdadero : resultado_si_falso**.

Si vamos a utilizar fórmulas con números decimales es conveniente modificar el símbolo que utiliza Freeplane para la coma decimal pues en algunos casos podemos obtener errores. Para evitar errores de este tipo nos iremos a las preferencias, pestaña de comportamiento y en la sección “Formateo y análisis de datos” elegimos del desplegable de Idioma para los formatos el inglés.

4.4.1. Elementos básicos

Podemos ver en la tabla siguiente un resumen de los operadores que podemos encontrar en las fórmulas de Freeplane.

Operadores
básicos

Operador	Se escribe	Ejemplo	Resultado
+	+	= 2 + 3	5
-	-	= 11 - 1	10
*	*	= 2 * 3	6
/	/	= 50 / 2	25
potencia	**	= 3 ** 2	9
paréntesis	()	3 * (10 + 5)	45
número pi	Math.PI	= Math.PI	3.141592
número e	Math.E	= Math.E	2.718281
seno	Math.sin()	= Math.sin(2 * Math.PI)	0
coseno	Math.cos()	= Math.cos(2 * Math.PI)	1
tangente	Math.tan()	= Math.tan(2 * Math.PI)	0
logaritmo decimal	Math.log10()	= Math.log10(100)	2
logaritmo neperiano	Math.log()	= Math.log(Math.E)	1

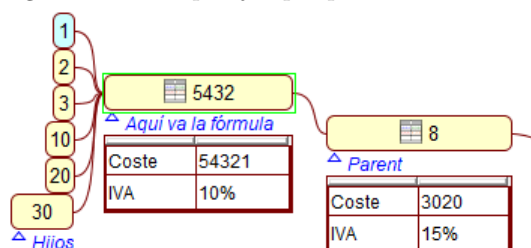
4.4.2. Elementos de los nodos

Aparte de los elementos estáticos de cualquier fórmula tales como los operadores anteriores y los números, en nuestras fórmulas también podemos referirnos a los nodos del mapa. De esta forma podemos sumar en el nodo padre el valor de todos sus nodos hijo o incluso acceder a los atributos de los nodos y operar con ellos.

En la siguiente tabla encontramos cómo podemos referirnos a estos elementos. Utiliza el mapa de la figura 4.4.2 para los resultados de los ejemplos. La fórmula la escribimos en el nodo central que contiene el texto 5432.

Descripción	Ejemplo	Resultado
Valor del padre	<code>= 2 * parent.value</code>	16
Valores de los nodos hijos	<code>= children.value</code>	[1, 2, 3, 10, 20, 30]
Cantidad de hijos	<code>= children.size</code>	6
suma de los nodos hijos	<code>= children.sum(0)</code>	66
Atributo coste del nodo	<code>= node['Coste']</code>	54321 (texto)
Valor numérico del coste	<code>= node['Coste'].to.num0</code>	54321 (número)
Atributo Coste del padre	<code>= parent['Coste']</code>	3020
Precio final = Coste * IVA	<code>= node['Coste'].to.num0 * node['IVA'].to.num0</code>	5432.1

Figura 4.4.2: Mapa ejemplo para las fórmulas



Hay que hacer una puntualización. Por defecto Freeplane trata a todos los números como si fuera texto. Esto quiere decir que no se puede realizar operaciones numéricas con ellos puesto que son tratados como si fueran letras. Para poder operar hay que acceder a su valor numérico. Esto es lo que hacemos al escribir `parent.value` ó `node['Coste'].to.num0`.

Realmente las fórmulas de Freeplane son más que fórmulas pues podemos poner en ellas cualquier código que produzca un resultado. Esto lo veremos con mucha más profundidad en la parte avanzada.

Vemos en la figura 4.4.3 nuestro mapa con un ejemplo. El nodo con el texto “3 Facturas” contiene realmente la fórmula que aparece en el cuadro “Editar fórmula”. Dicho cuadro de diálogo es el que aparece cuando editamos un nodo con una fórmula. A medida que vayamos añadiendo más facturas ese número irá cambiando pues contiene la expresión “children.size” que nos proporciona la cantidad de hijos del nodo indicado.

4.5. Fechas

Freeplane tiene características para gestionar el tiempo tales como recordatorios y fechas en los nodos. La herramienta fundamental para la gestión del tiempo es accesible mediante Herramientas, Mostrar calendario. Aparecerá la ventana de Administración del tiempo (ver figura 4.5.1).

Con ella podremos:

- Añadir la fecha actual al nodo
- Añadir un recordatorio en la fecha seleccionada

Figura 4.4.3: Ejemplo de un nodo con fórmula

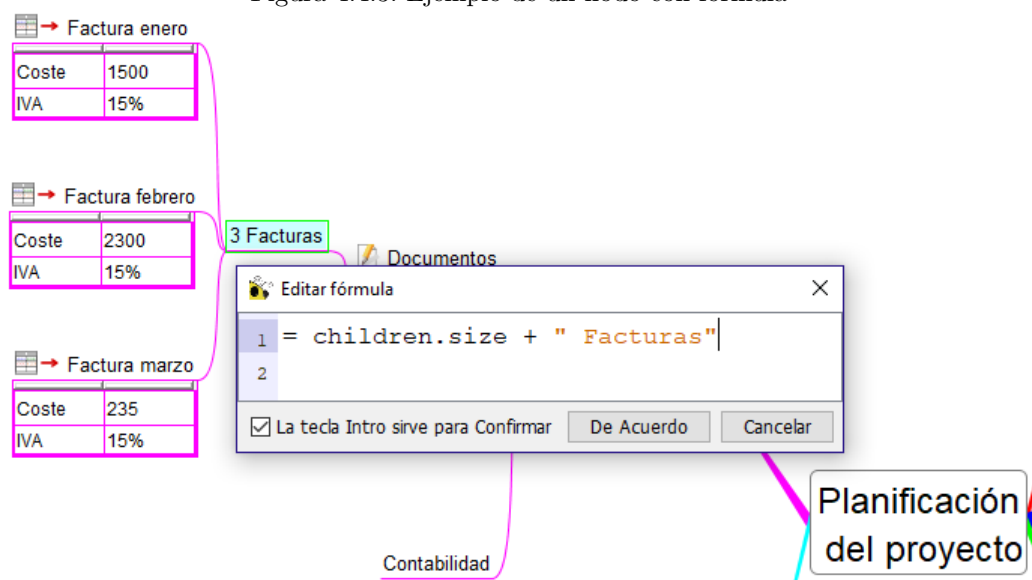
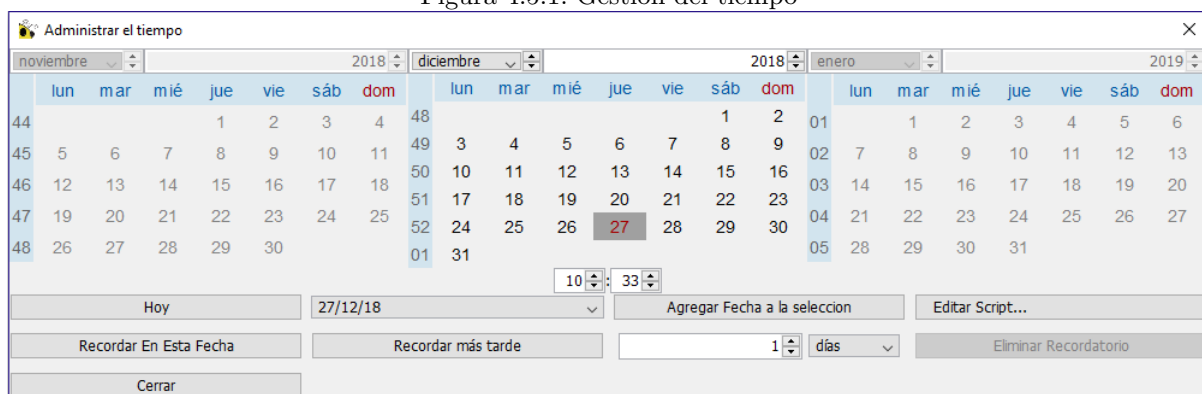


Figura 4.5.1: Gestión del tiempo



- Añadir un recordatorio en otra fecha posterior
- Ejecutar un script cuando llegue una fecha / hora determinada.

Tres nodos con recordatorios

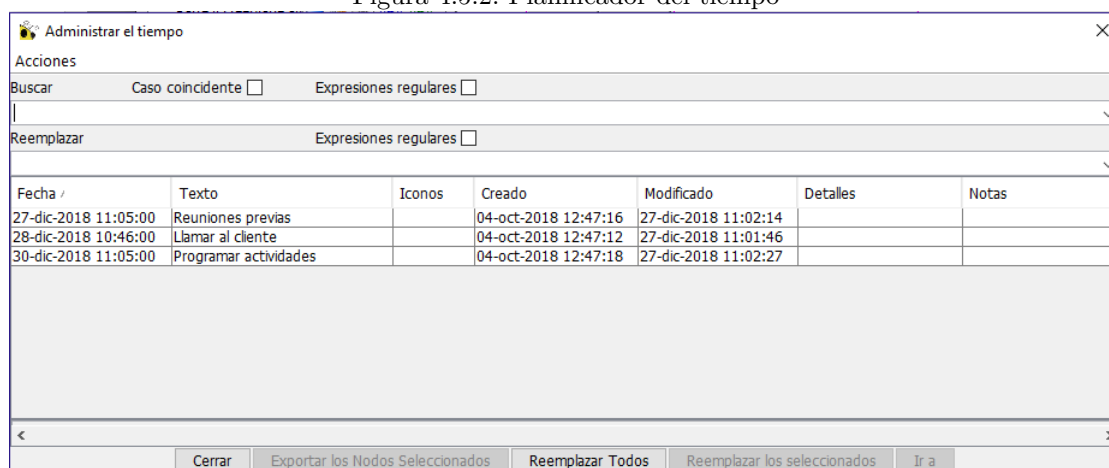
En dicha herramienta se pueden ver tres minicalendarios y una hora bajo ellos. Pinchando en el botón “Hoy” los calendarios y hora se posicionan en la fecha y hora actuales. Podemos irnos a otros meses si cambiamos el mes del centro (eligiéndolo del desplegable del mes que hay a la izquierda del año). También podemos modificar la hora. Una vez cambiada la fecha / hora de los minicalendarios podremos añadir dicha fecha al nodo con el botón “Agregar Fecha a la selección”.

O también un recordatorio en esta fecha al pinchar en “Recordar En Esta Fecha”. Cuando llegue ese momento Freeplane mostrará un aviso que puede configurarse en las preferencias de la aplicación. Normalmente se verán los nodos con los recordatorios parpadeando y mostrando un pequeño icono. Para avisar en el caso de que el nodo con el recordatorio esté colapsado, Freeplane hará parpadear toda la ruta desde la raíz al nodo con el recordatorio. También se puede configurar para que aparezca una ventana emergente con la notificación.

Cuando nos salte la notificación podremos ir a ella y eliminarla para que el nodo no siga parpadeando.

Si en lugar de recordarlo en esa fecha queremos recordarlo días u horas posteriores pincharemos en “Recordar más tarde” indicando previamente cuantos días / horas / minutos de retraso tendrá el aviso (desplegables bajo el botón “Agregar Fecha a la selección”).

Figura 4.5.2: Planificador del tiempo



Fecha /	Texto	Iconos	Creado	Modificado	Detalles	Notas
27-dic-2018 11:05:00	Reuniones previas		04-oct-2018 12:47:16	27-dic-2018 11:02:14		
28-dic-2018 10:46:00	Llamar al cliente		04-oct-2018 12:47:12	27-dic-2018 11:01:46		
30-dic-2018 11:05:00	Programar actividades		04-oct-2018 12:47:18	27-dic-2018 11:02:27		

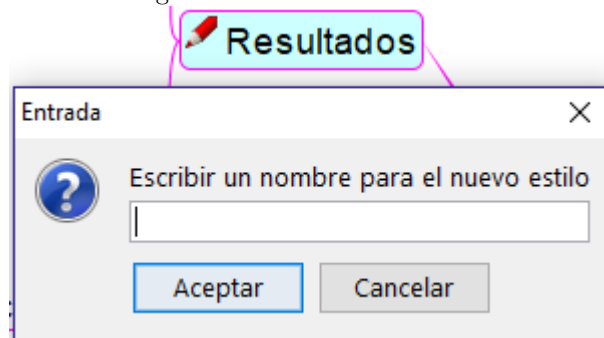
Podemos efectuar cierta planificación de tareas añadiendo recordatorios a los nodos que contienen las tareas a realizar. Una vez hecho tenemos acceso al planificador si pinchamos en Herramientas, Mostrar la lista del planificador de tiempo. Aparecerá la ventana de la figura 4.5.2 en la que se pueden ver las tareas planificadas, su fecha así como los iconos, detalles, notas y fechas de creación y modificación de los nodos en cuestión.

Capítulo 5

Definiendo estilos y plantillas

5.1. Introducción

Figura 5.1.1: Definiendo estilos



En el capítulo 3 hemos modificado el aspecto de nuestro mapa cambiando alguno de sus elementos. Para ello íbamos aplicando uno a uno todos los cambios que necesitábamos. Y si queríamos dos nodos iguales no teníamos más remedio que ir al segundo nodo y volver a aplicar los mismos cambios que en el primero.

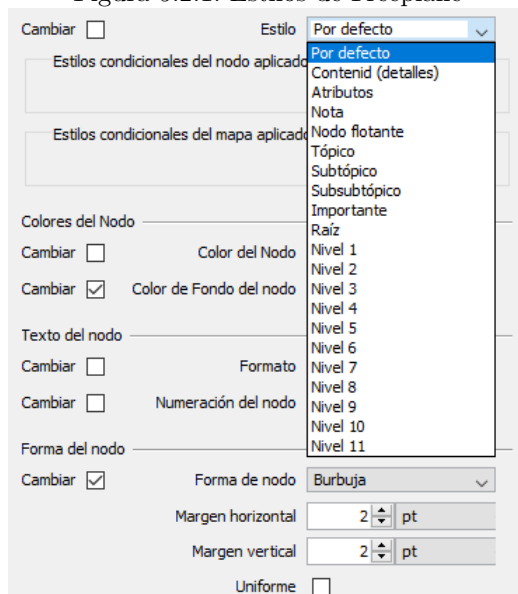
Para evitar el tedio que supone aplicar una y otra vez los mismos formatos Freeplane dispone de los Estilos. Un estilo es, por lo tanto, un conjunto de formatos al que damos un nombre. De esta manera podemos aplicarlos todos juntos con un solo click. Freeplane viene ya con un conjunto básico de estilos. Si nos vamos al panel de formato lateral (que si recordáis es el elemento número 6 en la figura 2.2.1) veremos un desplegable que se llama “Estilos”. Si lo expandimos nos aparecen los estilos, tal y como se ve en la figura 5.2.1. Cuando escogemos uno de ellos veremos cómo se modifica el aspecto del nodo seleccionado.

Definir un estilo es muy fácil. Para ello únicamente tenemos que aplicar todos los formatos a uno de los nodos y una vez tenga el aspecto deseado le daremos un nombre mediante el menú Formato, Gestionar estilos, Nuevo estilo a partir de la selección con el nodo seleccionado. Aparecerá el cuadro de la figura 5.1.1 donde escribiremos el nombre del estilo.

5.2. Estilos predefinidos

Freeplane incorpora ya algunos estilos definidos, como vemos en la figura 5.2.1. Los estilos nivel 1 a nivel 10 son los que se aplican si activamos la opción del panel de formato “Color del borde automático” y elegimos del desplegable “por niveles”. Los colores que se aplican pueden modificarse pinchando en el

Figura 5.2.1: Estilos de Freeplane

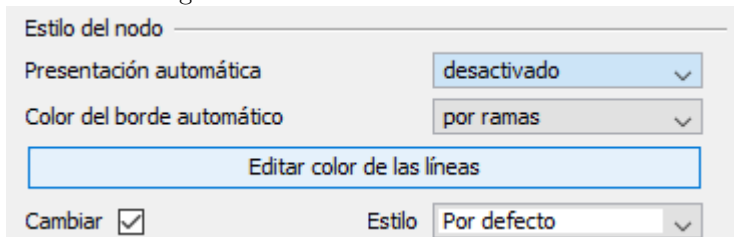


botón “Editar color de las líneas”. Si por el contrario elegimos “por ramas” los colores antes definidos se aplican a medida que se van añadiendo ramas.

El estilo llamado Raíz es el que se aplica al nodo raíz. Los denominados Tópico, Subtópico, Subsubtópico e Importante varían el color, tamaño de la letra y en el último caso añade un icono.

Podemos hacer que Freeplane varíe el color de los nuevos nodos

Figura 5.2.2: Panel del estilo del nodo



5.3. Modificando los estilos ya definidos

Modificando los estilos ya definidos

A medida que vayamos definiendo nuestros propios estilos éstos irán apareciendo en los desplegables de los estilos de los nodos. Una opción fundamental a la hora de trabajar con estilos en freeplane es el Editor de estilos que podemos ver en la figura 5.3.1. Es un mapa en el cual aparecen todos los estilos del mapa con el que estamos trabajando. Se muestra al pinchar en Formato, Gestionar estilos, editar estilos.

Mediante este panel podemos cambiar el formato del estilo así como borrar los estilos que no nos interesen. Vemos que aparece desplegado continuamente el panel de formato lateral de la derecha.

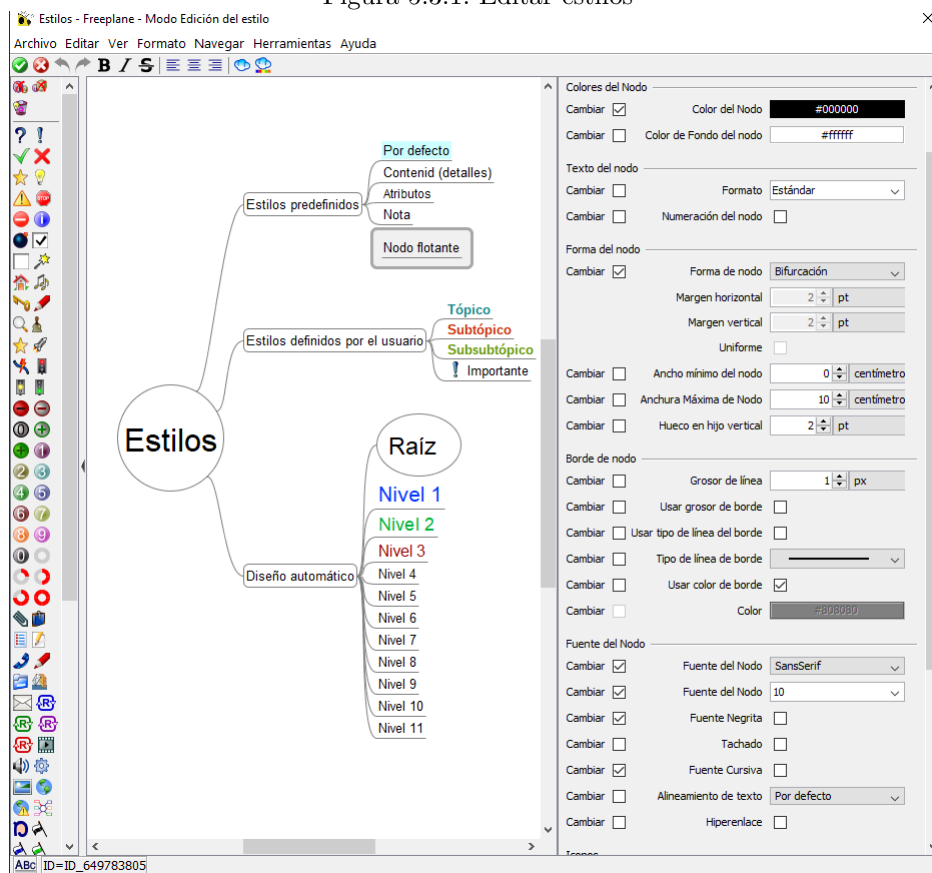
Hay cinco estilos interesantes: Por defecto, Contenido, Atributos, Nota y Nodo Flotante.

Por defecto es el formato que tienen todos los nodos si no les hemos cambiado nada. Podemos cambiar, por ejemplo, el color y el tamaño de todos los nodos al modificar este estilo.

Contenido es el formato que se aplica a los detalles del nodo que, como hemos visto, es el texto que podemos añadir bajo el nodo.

Atributos nos indica cómo se verán los atributos de los nodos.

Figura 5.3.1: Editar estilos



Nota indica cómo se verá la nota en el panel de notas.

Nodo Flotante: formato aplicado a los nodos flotantes. Un nodo flotante es aquel que podemos añadir al mapa al hacer doble click en una zona del mismo mientras mantenemos presionada la tecla **INSERT**.

Vemos que los estilos Tópico, Subtópico, Subsubtópico e Importante aparecen en la rama “Estilos definidos por el usuario”. Estos estilos los podemos borrar pinchando con el botón derecho del ratón y eligiendo “Eliminar estilo definido por el usuario”.

Los estilos definidos por nivel los borraríamos pinchando con el botón derecho en cualquier nodo y eligiendo “Eliminar estilo automático por nivel”. Borrará el último nivel que haya (que inicialmente es el nivel 11).

Una vez hechos los cambios deseados en los estilos ya podemos guardar. Para ello pinchamos en el menú Archivo, De acuerdo. Si nos hemos liado y no queremos guardar nada pincharíamos en Archivo, Anular.

5.4. Guardando nuestros estilos: las plantillas

Guardando la plantilla para su uso posterior

Si te has pasado mucho tiempo creando estilos en tu mapa podrás guardarlos para utilizarlos como plantilla en lugar de la plantilla inicial que se carga al crear un nuevo mapa.

Como puedes imaginar, una plantilla es un diseño básico del mapa con estilos definidos. Para poder crearla tenemos varias opciones que se explican en las secciones siguientes

5.4.1. Utilizar estilos definidos en otro mapa

Cuando creamos un nuevo mapa podemos cargar los estilos de otro mediante el menú Formato, Gestionar estilos, cargar estilos del mapa...

5.4.2. Guardar el mapa actual como plantilla

Imaginemos que queremos usar el mapa que llevamos haciendo durante todo este tiempo como plantilla. De esta forma cada vez que tengamos un proyecto nuevo, al crearlo con esta plantilla ya aparecerán todas las ramas fundamentales: Tareas, Recursos, Gente, Resultados y Elementos para ordenar así como los estilos que hayamos definido.

Esto se realiza guardando el mapa que queremos usar como plantilla en el directorio templates del directorio del usuario que podemos mostrar al pinchar en herramientas, abrir el directorio del usuario.

Cualquier mapa que aparezca en esa carpeta se podrá utilizar como plantilla.

5.4.3. Usar la plantilla

Una vez ya tengamos nuestra plantilla guardada en la ubicación correcta podremos usarla al elegir el menú Archivo, Nuevo mapa usando plantilla. Freeplane mostrará la carpeta Templates del directorio del usuario donde podremos elegir la plantilla que queramos.

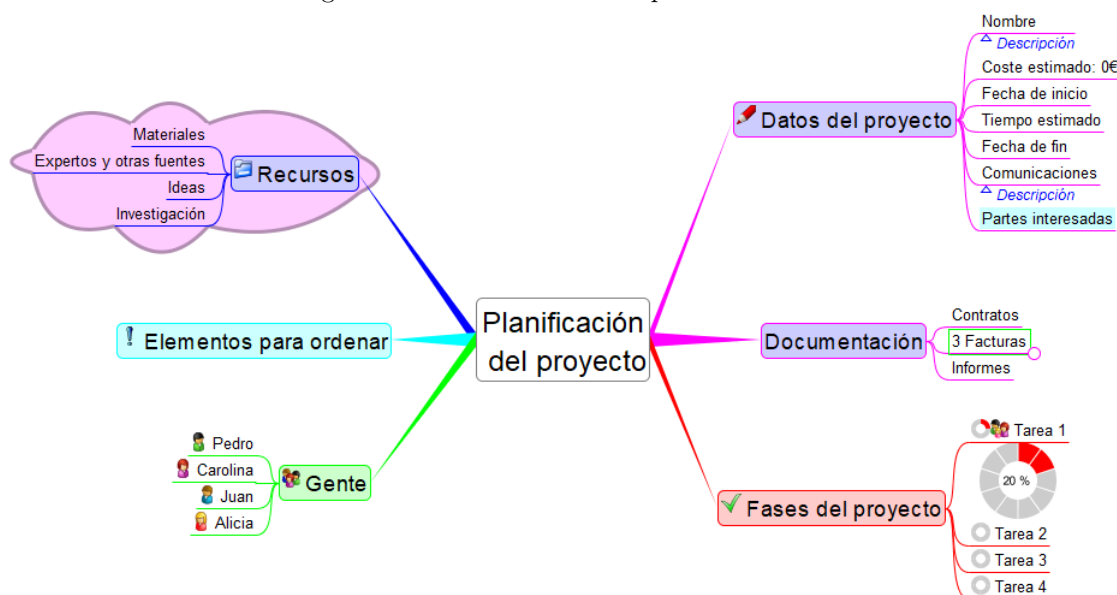
También podemos utilizar cualquier otro mapa aunque no lo hayamos guardado como plantilla si en la ventana anterior navegamos hasta dicho mapa y lo elegimos.

Los pasos para crear una plantilla son, por lo tanto:

1. Definir la estructura de nodos que queremos que aparezca inicialmente
2. Crear los estilos del mapa
3. Guardarlo en el directorio templates del directorio del usuario
4. Crear un nuevo mapa a partir de la plantilla anterior.

5.5. Creando una plantilla para nuestros proyectos

Figura 5.5.1: Creación de una plantilla. Fase 1



¡Vamos con un ejemplo!

Como práctica vamos a utilizar el mapa de los primeros capítulos como base para una plantilla para mapas mentales de gestión de proyectos. Vamos a partir de una versión modificada de aquel mapa. Aparece en la figura 5.5.1. Abrir vuestro mapa y realizar todas las modificaciones para que coincida con el de la figura indicada.

Ahora vamos a definir una serie de estilos que nos faciliten la labor a la hora de introducir nuevos elementos. Vamos a partir de las facturas. Cambiaremos su forma para que sea rectangular de color de fondo gris claro. Le pondremos ancho mínimo y máximo 5 y márgenes cero. Cuando tengamos un nodo con estos formatos aplicados lo definiremos como estilo:

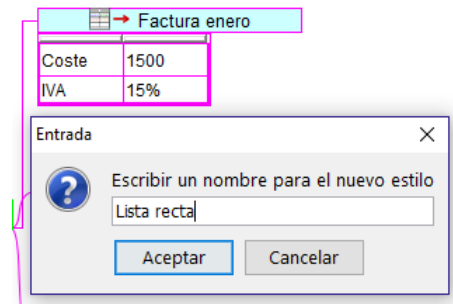
1. Seleccionamos el nodo
2. Elegimos el menú Formato, Gestionar Estilos, Nuevo estilo a partir de la selección y tecleamos Lista recta.

Ya tenemos el estilo para este tipo de documentos. Podemos usarlo, aparte de para las facturas, para otras cosas. Por eso lo hemos llamado “lista recta”.

El tipo de letra por defecto es algo feo. Vamos a cambiarlo:

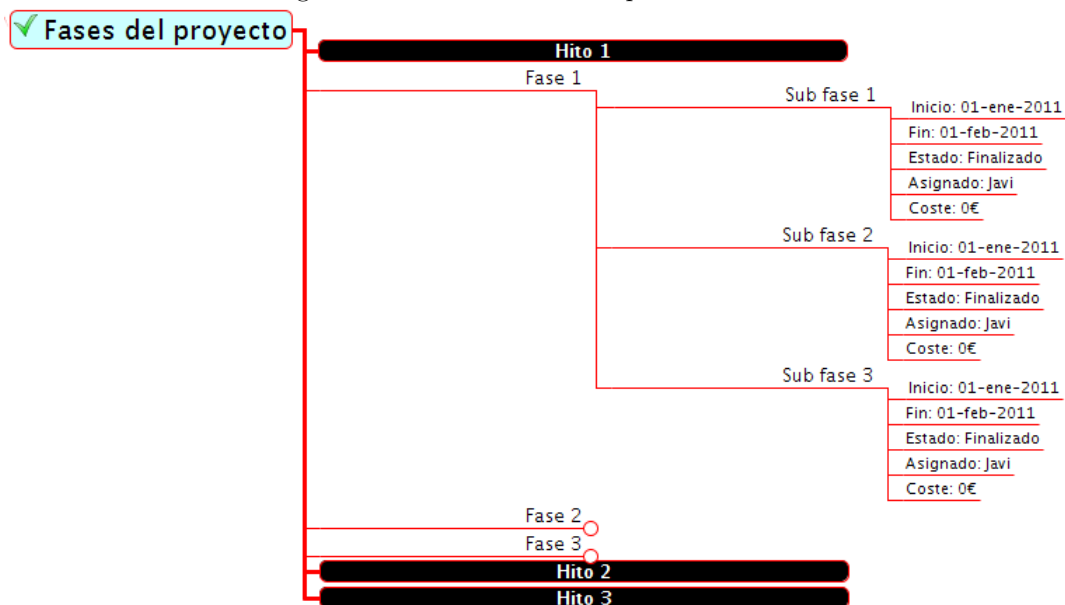
1. Menú Formato, Gestionar Estilos, Editar estilos para sacar el mapa con los estilos definidos. Vemos que ya aparece el que acabamos de definir, “lista recta”.
2. Pinchamos en el nodo “Por defecto” y elegimos como tipo de letra uno algo más atractivo. Por ejemplo, Lucinda Sans.
3. Salimos mediante el menú Archivo, De acuerdo. Vemos que se han modificado todos los nodos.

Figura 5.5.2: Definiendo el estilo para las facturas



La rama para las fases del proyecto no está muy trabajada. Sería mejor usar otro tipo de estructura. Primero vamos a borrar todas las tareas. Luego crearemos los nodos en la rama “Fases del proyecto” tal y como aparecen en la figura 5.5.3.

Figura 5.5.3: Creación de una plantilla. Fase 2



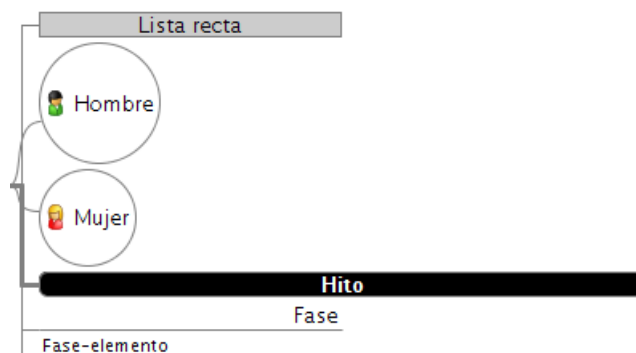
Los hitos son nodos de forma de burbuja con ancho máximo y mínimo de 10, fondo negro y letras blancas en negrita. Hemos arrastrado manualmente los nodos para que queden debajo del nodo padre y darle así un aspecto más atractivo a esa rama. Las fases y subfases son nodos con forma de bifurcación y anchos mínimo y máximo de 5 que también hemos arrastrado hacia abajo. Los detalles de las subfases con nodos de tipo bifurcación de tamaño de letra de 8 sin anchos definidos. Lamentablemente, la posición de los nodos no se puede fijar mediante estilos por lo que debemos hacerlo manualmente. Sin embargo podemos guardar la plantilla con un par de Hitos y fases de ejemplo.

Cuando ya tenemos la estructura creada definimos tres estilos: uno para el hito, otro para la fase y subfase y otro para los detalles de las subfases.

Para las personas definiremos dos estilos: uno para el hombre con un icono de un hombre y otro con el icono de una mujer. Serán ovalados pero marcando la casilla Uniforme para que sean círculos. Podemos definir un ancho determinado para darle un aspecto más uniforme. Al final los estilos quedan tal y como aparecen en la figura 5.5.4.

Los 6 estilos
que hemos
definido

Figura 5.5.4: Estilos de nuestra plantilla

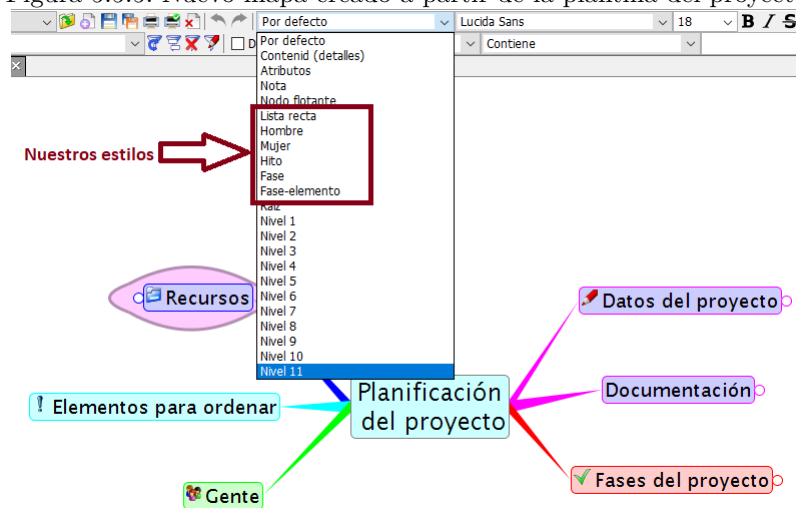


Ahora únicamente nos queda guardar la plantilla para poder utilizarla con los proyectos nuevos. Para ello elegimos el menú Archivo guardar. Le ponemos de nombre “Proyecto” y lo dejamos en un lugar conocido, por ejemplo el escritorio del ordenador.

Luego vamos al menú Herramientas, Abrir el directorio del usuario y arrastramos el archivo Proyecto.mm que habíamos creado en el escritorio al directorio templates del directorio del usuario.

Ahora ya podemos crear nuestros mapas de proyectos eligiendo Archivo, Nuevo mapa a partir de plantilla. Escogemos la plantilla proyecto.mm y obtenemos la imagen de la figura 5.5.5. En el desplegable superior aparecen los estilos que hemos definido: “Lista recta”, “Hito”, “Fase”, “Fase-elemento”, “Hombre” y “Mujer”.

Figura 5.5.5: Nuevo mapa creado a partir de la plantilla del proyecto



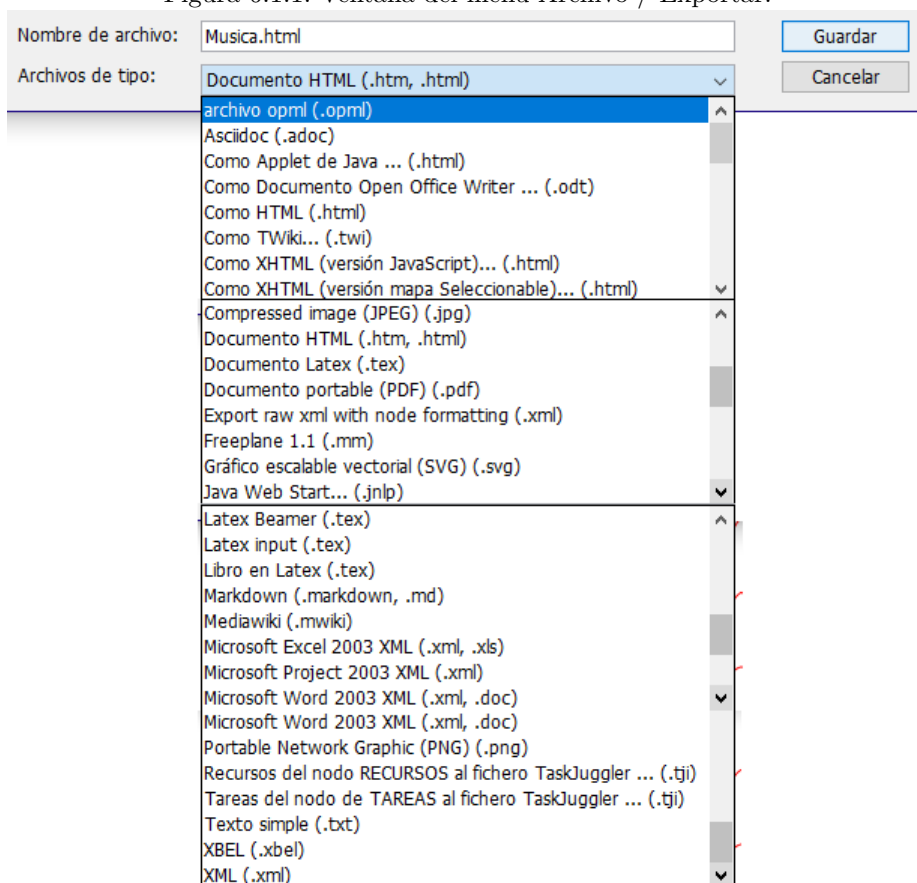
Capítulo 6

Utilizando el mapa con otras aplicaciones

6.1. Introducción

Formatos reconocidos por Freeplane para exportar el mapa

Figura 6.1.1: Ventana del menú Archivo / Exportar.



Ya tenemos nuestro mapa construido. Hemos ido creando la estructura de ramas que responde a la visión que tenemos del problema y deseamos compartirlo con otras personas.

Ya sea como imagen o como estructura de ramas, Freeplane dispone de múltiples formatos para exportar la información del mapa, accesibles desde el menú Archivo, exportar. En la figura 6.1.1 aparecen

todas las opciones disponibles. A continuación se muestran algunas de ellas.

6.2. Utilizándolo como imagen

Lo más sencillo a la hora de obtener una imagen del mapa con el que estamos trabajando es ir a Archivo, Exportar y elegir alguno de los formatos de imagen disponibles en la lista:

- Compressed image (JPEG): Guarda el archivo como imagen .jpg
- Gráfico escalable vectorial (SVG): Guarda el archivo como imagen vectorial, .svg
- Portable Network Graphic (PNG): Guarda el archivo como imagen .png

Previamente debemos asegurarnos de expandir las ramas que deseamos mostrar y colapsar aquellas que no nos interesan y elegir una ubicación y un nombre para el fichero de imagen que deseamos guardar. El mapa se guardará exactamente como lo muestre Freeplane.

Una opción interesante si deseamos enviar a alguien una foto de nuestro mapa es escoger de la lista el formato PDF. Freeplane generará una página pdf con la imagen que puede ser vista desde cualquier equipo.

6.3. Guardando el esquema de la información

Si lo que deseamos es guardar la estructura de la información podemos exportar el mapa a un fichero de texto, txt. Si exportamos el mapa de la sección anterior, y que vemos en la figura 5.5.5, obtendríamos el siguiente fichero de texto (se muestra incompleto):

```
#MindMapExport FreeplaneVersion:freeplane 1.7.0
```

```
1 Planificación del proyecto
1.1 Recursos
1.1.1 Materiales
1.1.2 Expertos y otras fuentes
1.1.3 Ideas
1.1.4 Investigación
1.2 Datos del proyecto
1.2.1 Nombre
DETAILS:
Descripción
1.2.2 Coste estimado: 0€
1.2.3 Fecha de inicio
1.2.4 Tiempo estimado
1.2.5 Fecha de fin
1.2.6 Comunicaciones
DETAILS:
Descripción
1.2.7 Partes interesadas
1.3 Documentación
1.3.1 Contratos
1.3.2 4 Facturas
1.3.2.1 Factura enero
1.3.2.2 Factura febrero
1.3.2.3 Factura marzo
1.3.2.4 Fatura abril
1.3.3 Informes
1.4 Fases del proyecto
1.4.1 Hito 1
1.4.2 Fase 1
```

El texto del nodo se muestra en una línea numerada. Los detalles se muestran bajo el nodo con el texto “DETAILS” y los atributos no se exportan.

Otra forma más potente es exportarlo como fichero de excel. Para este mismo ejemplo obtendríamos la imagen 6.3.1.

Figura 6.3.1: Exportamos el mapa como archivo de excel

	A	B	C	D	E	F
1	Planificación del proyecto					
2		Recursos				
3			Materiales			
4			Expertos y otras fuentes			
5			Ideas			
6			Investigación			
7		Datos del proyecto				
8			Nombre			
9			Coste estimado: 0€			
10			Fecha de inicio			
11			Tiempo estimado			
12			Fecha de fin			
13			Comunicaciones			
14			Partes interesadas			
15		Documentación				
16			Contratos			
17			4 Facturas			
18			Factura enero	Names	Values	
19				Coste	1500	
20				IVA	15%	
21			Factura febrero	Names	Values	
22				Coste	2300	
23				IVA	15%	
24			Factura marzo	Names	Values	
25				Coste	235	
26				IVA	15%	
27			Fatura abril			
28			Informes			
29		Fases del proyecto				
30			Hito 1			
31			Fase 1			

El mapa
exportado
a Excel

El texto del nodo se muestra indentado en celdas de excel. Los detalles bajo el nodo se guardan como comentarios de las celdas y los atributos se muestran como una tabla, según podemos ver en la imagen anterior. Una vez tenemos la información en excel ya podemos tratarla y hacer cálculos.

Si lo que deseamos es obtener un archivo estándar para compartir la información con aplicaciones de datos lo mejor es utilizar el formato de intercambio de información .XML escogiendo Export raw xml with node formatting (XML).

6.4. Mapa como página web

Aunque hay en la lista de formatos para exportar opciones para guardar el mapa como página web, la configuración de seguridad actuales de los navegadores habituales impiden su utilización, pues bloquean los elementos java y javascript de las páginas web. dichos formatos serían:

- Como HTML
- Como XHTML (versión mapa seleccionable)
- Documento HTML
- Java Web Start (.jnlp)

6.5. Importando datos

Podemos insertar ramas enteras a nuestro mapa de freeplane directamente desde un editor de textos **utilizando la opción de copiar y pegar**. Dado el siguiente archivo de texto (se muestran los tabuladores como segmentos al comienzo de las líneas):

```
Tareas del proyecto
    Tarea 1
    Tarea 2
    Tarea 3
    Tarea 4
Recursos
    Contrato
    Información
    Ideas
    Investigación
Documentos
    Facturas
        Factura enero
            Coste 1500
            IVA 0.15
        Factura febrero
            Coste 2300
            IVA 0.15
        Factura marzo
            Coste 235
            IVA 0.15
```

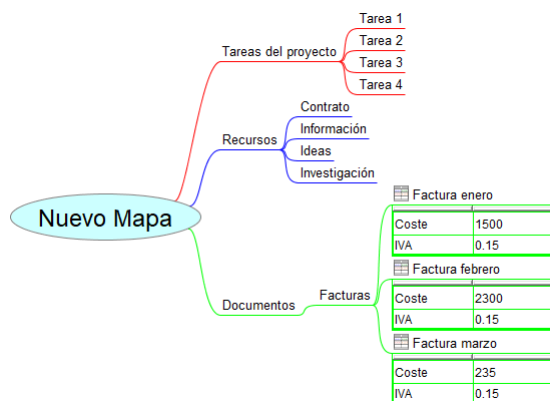


Figura 6.5.1: Importar copiando de un archivo de texto

En el archivo anterior utilizamos tabuladores para definir los nodos hijos de uno determinado. o sea, que los tecleamos bajo él pero indentadas mediante un tabulador. De esta forma “Tarea 1”, “Tarea 2”, “Tarea 3” y “Tarea 4” son nodos hijos de “Tareas del proyecto”.

Así mismo si un nodo tuviera atributos los teclearíamos bajo él utilizando la sintaxis “+ \t nombre_del_atributo \t valor_del_atributo” siendo \t un tabulador. En el archivo anterior vemos que “Factura enero”, nodo hijo de “Facturas” tiene dos atributos: Coste = 1500 e IVA=0.15.

Si copiamos en el portapapeles de windows el contenido del fichero anterior y lo pegamos en un nuevo mapa, el resultado que obtenemos aparece en la figura 6.5.1.

Otras formas de importar datos es mediante el menú Archivo, Importar. Podemos utilizar otro mapa o bien un fichero XML.

6.6. Importar los favoritos del navegador

Desde el menú Archivo, Importar, Favoritos del Explorer obtendremos un mapa con todos los favoritos que hemos añadido al explorer de Windows. Sólo tendremos que elegir nuestra carpeta de favoritos que suele estar ubicada en `C:\Users\<usuario>\Favoritos`.

6.7. Importar la estructura de carpetas de nuestras unidades

Si pinchamos en Archivo, Importar, Estructura de carpetas se mostrará un archivo con la estructura de carpetas del directorio escogido. Otra forma de ver un mapa mental con la estructura del disco duro del ordenador es elegir Mapas, Modos, Gestor de archivos.

Capítulo 7

Buscando información en nuestro mapa

7.1. Búsqueda rápida

Cuando el mapa con el que estamos trabajando se va llenando podemos tener problemas a la hora de localizar la información. Para esto Freeplane dispone de una herramienta muy interesante: la barra de búsquedas. En el margen derecho de esta página podemos ver qué aspecto tiene (se muestra en vertical). Puede mostrarse pulsando **CTRL** + **F** o eligiendo del menú Ver, Controles, Barra de herramientas de filtros y se encuentra justo sobre el mapa. Esta barra tiene varias partes:

- Casilla Denegar: al marcarla busca los nodos que no coincidan con el texto tecleado.
- Desplegable de la ubicación donde buscar: Nos permite buscar en el texto del nodo, en los detalles y en las notas (valor por defecto). También podemos buscar iconos, conectores, estilos etc.
- Desplegable del tipo de comparación: Por defecto se marca “Contiene” lo cual permite buscar aquellos nodos que contienen la cadena de búsqueda. También podemos buscar valores numéricos iguales, superiores o inferiores a uno dado o la potente opción de “Expresiones regulares” que introduciremos a continuación.
- Desplegable para introducir la cadena de búsqueda: aquí tecleamos lo que queremos buscar
- Caso coincidente: Encuentra solo si el texto coincide totalmente con la cadena introducida.
- Aproximado: Encuentra nodos que contengan una aproximación al texto tecleado.
- Flechas de navegación (azules): Avanzan y retroceden al siguiente nodo encontrado
- Icono de Filtro rápido: Deja visibles únicamente los nodos encontrados
- Icono de Destacar los nodos: marca con un recuadro aquellos nodos que coinciden.

En esta barra podemos buscar en varias ubicaciones del mapa: en el texto del nodo, en los detalles, en las notas, podemos buscar iconos, enlaces etc. Normalmente está seleccionado “Esencial, detalle o notas” indicando que es allí donde buscará. Únicamente tenemos que escribir aquello que deseamos buscar y pinchar en las flechas azules de esa misma barra para navegar por todas las coincidencias. Si marcamos la casilla “Denegar” Freeplane encontrará todos los nodos que NO coincidan con la búsqueda realizada. Y si marcamos “Aproximado” encontrará todo lo que se parezca.

Sin embargo hay otras dos opciones más interesantes que son las dos últimas opciones de esa barra de búsquedas:

- Filtro rápido para dejar visibles sólo los nodos coincidentes.
- Destacar todos los nodos coincidentes: marca con un recuadro aquellos nodos que coinciden.

Debajo se muestra la barra de búsqueda



En la sección 12.2 se profundizará más sobre este tema

7.2. Expresiones regulares de Freeplane

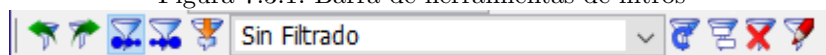
Una de las opciones de búsqueda más potentes son las expresiones regulares. Son caracteres comodín que nos permiten hacer búsquedas genéricas. Así si tecleamos eso buscará palabras de 7 letras que tengan la terminación -eso como “Proceso” o “Expreso”, pues el punto equivale a cualquier carácter. Si buscamos car+o buscará las palabras caro, carro, carro pero también Icaro. pues un símbolo más supone repetir una o más veces la letra precedente. Un asterisco significa que el carácter precedente se puede repetir cero o más veces de forma que car*o buscará cao, caro, carro, etc y también Ícaro, encaro, etc. El símbolo ^ corresponde al principio de la línea y \$ al final de forma que ^A+ buscará todos los nodos que comiencen con la letra A. Las expresiones regulares nos pueden dar muchos problemas. Por ejemplo sin en la búsqueda anterior nos confundimos y tecleamos ^A* nos encontrará, sin embargo todos los nodos del mapa pues la A se repite CERO o más veces al principio. Hay símbolos genéricos más interesantes. Así si ponemos \s nos buscará un espacio en blanco y \w es cualquier letra de la a a la z mayúscula o minúscula. Debemos tener en cuenta que si buscamos usando expresiones regulares, “E” es distinto a “e” pues se diferencian mayúsculas de minúsculas. \d es un dígito. Si ponemos \d\d\d encontrará números de 3 dígitos.

Las expresiones regulares nos dan muchas más opciones que veros en la parte avanzada de esta guía. De momento estas nos permitirán realizar búsquedas muy poderosas. En la tabla siguiente se muestra una lista reducida de opciones. En la parte avanzada veremos esto en profundidad.

carácter	significado
.	Cualquier carácter
*	cero o más repeticiones del carácter anterior
+	uno o más repeticiones del carácter anterior
^	Comienzo de línea
\$	Fin de línea
\w	Cualquier carácter de una palabra: a-zA-Z
\W	Cualquier carácter que no sea el anterior
\s	Espacio en blanco
\S	Cualquier carácter que no sea un espacio en blanco
\d	Un dígito
\D	Cualquier cosa que no sea un dígito
\t	Tabulación
\r	Retorno de carro
\n	Nueva línea

7.3. Filtros

Figura 7.3.1: Barra de herramientas de filtros

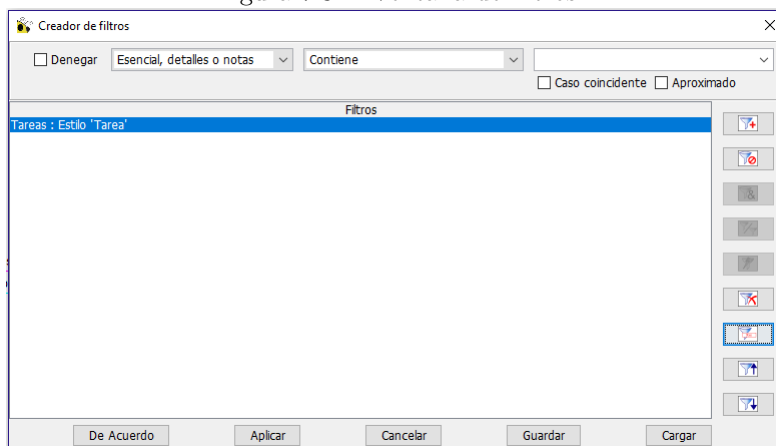


Podemos “podar” nuestro mapa aplicándole un filtro. Mostrará solo los nodos coincidentes.

Otra forma de visualizar la información que buscamos es aplicar un filtro a los nodos. Un filtro sirve para ver determinados nodos del mapa ocultando el resto. Tenemos acceso a una barra de filtros mediante Ver, Controles, Barra de herramientas de filtros. Se muestra en la figura 7.3.1. En ella podemos aplicar filtros predefinidos, mostrar los nodos ascendientes o descendientes, quitar un filtro aplicado o editarlo.

La ventana que aparece cuando pinchamos en el menú Filtro, Editar o pinchamos en el último icono de esta barra aparece en la figura 7.3.2. Contiene los mismos elementos que hemos visto anteriormente para las búsquedas: dónde queremos buscar la información, condición a aplicar y texto a buscar. En dicha ventana, además, vemos un filtro ya definido al que hemos llamado “Tareas”. Según aparece en la parte inferior, esta condición mostrará únicamente los nodos cuyo estilo sea, precisamente “Tareas”.

Figura 7.3.2: Ventana de filtros



Como hemos dicho podemos darle un nombre a estas opciones de búsqueda. Dicho nombre aparecerá en el menú Filtro, user-defined filters y el el desplegable “Sin Filtro” de la barra de filtros. Si lo seleccionamos se aplicará el fitro que hemos guardado antes.

Podemos entender los filtros como una forma de ver el mapa desde distintas perspectivas.

Los distintos tipos de condiciones que hay son:

- Esencial, detalle o notas: Si queremos aplicar un estilo con nombre cuando el nodo, su detalle o sus notas contengan un texto determinado.
- Texto del nodo: Si queremos aplicar un estilo con nombre cuando el nodo contenga un texto determinado.
- Detalles: Si queremos aplicar un estilo con nombre cuando detalle del nodo contenga un texto determinado.
- Nota: Si queremos aplicar un estilo con nombre cuando la nota del nodo contenga un texto determinado.
- Texto de origen
- Icono: aplicamos el estilo si contiene un determinado icono.
- Hiperenlace: Si el hiperenlace del nodo es uno determinado
- Etiqueta del conector: si la etiqueta del conector del nodo es una concreta
- Conector: si tiene conector aplicamos un estilo.
- Filtro de fecha
- Prioridad
- Estilo
- Clones

- Nivel del nodo
- Recordatorio
- Filtro de script: código que tenga como resultado un booleano. Por ejemplo, usado con expresiones regulares (veremos esto en la parte avanzada):
 - `node.shortText==~ /.*(.-)$/` Se aplica el estilo si el texto del nodo acaba en “.-”
 - `node.shortText==~ /.*[!)$/` Se aplica el estilo si el texto del nodo acaba el “!”
 - `node.shortText==~ /^!.*/` Si el texto del nodo comienza con “!”
- Attribute name or value: si hay un atributo con ese nombre o con ese valor
- “nombre de atributo”: si el atributo con ese nombre contiene el valor especificado.

7.3.1. Gestión de los filtros guardados

La ventana de la figura 7.3.2 se utiliza no solo para definir filtros, sino también para gestionarlos. Vemos que en la parte derecha aparecen unos iconos que procedemos a explicar. Son en total 9 iconos cuyo significado, de arriba a abajo es el siguiente

1. Icono Añadir (con el + rojo): Añadir el filtro a nuestra lista
2. Icono No (con el símbolo \odot rojo): Añadimos otro filtro negando la condición seleccionada
3. Icono Y (con el símbolo $\&$): Seleccionando dos iconos, añadimos otro filtro aplicando ambas condiciones (condición Y, deben coincidir simultáneamente).
4. Icono O: Seleccionando dos iconos, añadimos otro filtro aplicando ambas condiciones (condición O, debe cumplirse o una condición o la otra).
5. Icono dividir: Si tenemos un filtro compuesto mediante condiciones O / Y, este icono lo divide en dos filtros.
6. Icono Borrar (con la X roja): Borra el filtro seleccionado
7. Icono nombrar: pone un nombre al filtro seleccionado
8. Icono Arriba (con la flecha hacia arriba): sube una línea el filtro seleccionado
9. Icono Abajo (con la flecha hacia abajo): baja una línea el filtro seleccionado

Hay que advertir aquí que los filtros se definen para el usuario, no para el mapa en curso. Esto significa que una vez definido un filtro y guardado tendremos acceso al mismo en cualquier mapa que abramos.

Capítulo 8

Haciendo presentaciones con Freeplane

Las presentaciones avanzadas se explican en el capítulo 11

8.1. Introducción

Contar con un apoyo visual a la hora de presentar un proyecto o incluso a nosotros mismos puede ser la diferencia entre el éxito y el fracaso. Si bien son de sobra conocidos los programas específicos para la realización de presentaciones tales como Powerpoint de Microsoft, Keynote de Apple o Impress de LibreOffice, cada vez están más de moda herramientas más originales que podemos encontrar en la web tales como Prezi.

Una presentación eficaz debe cumplir una serie características:

- Centrarse en el público y en el proyecto presentado.
- Ser simple, evitando detalles innecesarios o demasiada información.
- Emplear distintos tipos de recursos no solo texto. Incluiremos también gráficos, imágenes etc.
- Aprovechar todos los elementos gráficos disponibles. Una imagen vale más que mil palabras.

En este sentido, Freeplane se nos presenta como una herramienta ideal para cubrir todas estas condiciones. Podemos mostrar no solo texto e imágenes, sino que también se expondrán las relaciones y jerarquía de la información presentada. Además es vital conseguir mantener la atención del público y el uso de esta forma tan original redundará en nuestro beneficio.

8.2. Enseñando nuestro mapa

Si deseamos mostrar el mapa en público debemos, en primer lugar, prestar especial cuidado en su realización según decíamos en el primer capítulo. Cuando ya lo tengamos completado podremos comenzar a preparar la presentación.

Al público solo le interesa la información que tenemos que comunicar, por lo cual cuando se muestre el mapa en el proyector le ocultaremos todos los elementos que le puedan distraer, el propio Freeplane. Para hacer esto únicamente tenemos que activar Freeplane en pantalla completa pulsando F11. Se mostrará sólo el mapa y la ventana de notas si la tenemos abierta (podemos ocultarla pinchando con el botón derecho del ratón y eligiendo mostrar/ocultar ventana de notas).

Ya podremos mostrar por nuestro mapa navegando a través de los nodos. También podemos expandirlos y colapsarlos. Dependiendo de las opciones de visualización que tengamos activas el mapa se verá de una u otra forma. Particularmente útiles pueden ser las siguientes opciones, disponibles en el menú Ver, Ver configuración:

- Ver, vista en esquema: Se mostrará una representación en árbol del mapa.
- Resaltado: Aparecerá en gris todo el mapa excepto el nodo seleccionado.

- Muestra / Oculta la selección dentro de un rectángulo: Aparece un rectángulo alrededor del nodo seleccionado.
- Mueve el nodo seleccionado a la izquierda / derecha: Al pinchar en un nodo aparece en el borde izquierdo o derecho mostrando únicamente los nodos descendientes de él.
- Centrar el nodo seleccionado: Centra en medio de la pantalla el nodo seleccionado.

En la segunda parte os enseñaremos la herramienta de Freeplane para preparar presentaciones más elaboradas. Esto ya requiere más preparación, pues iremos confeccionando una serie de diapositivas con los nodos que elijamos. Esta herramienta se muestra en el panel lateral de formato eligiendo la pestaña “Presentaciones”.

Parte II

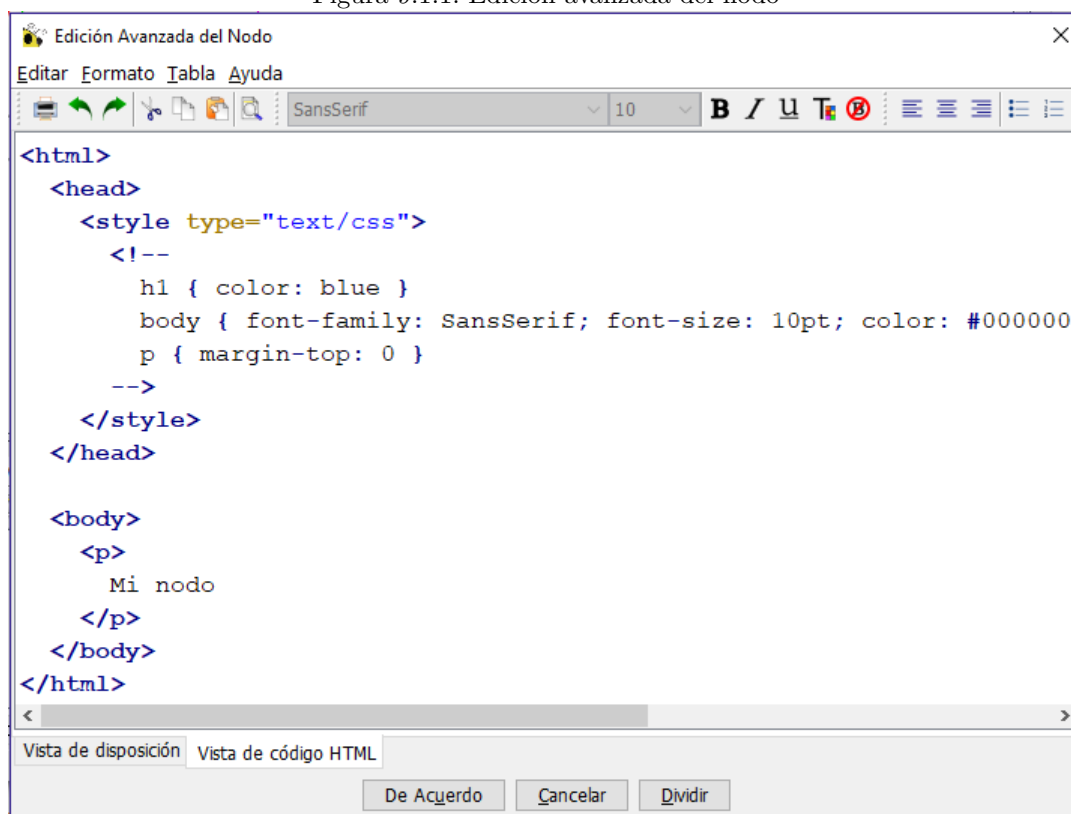
Freeplane avanzado

Capítulo 9

Texto avanzado

9.1. HTML en los nodos

Figura 9.1.1: Edición avanzada del nodo



En realidad, un nodo no es únicamente texto, sino código html. Podemos verlo si pinchamos con el botón derecho encima del nodo y elegimos “Editar el nodo”. Aparecerá la ventana de la figura 9.1.1 en la que se muestra el contenido del nodo “Mi nodo” con la pestaña “Vista de código HTML” activa en lugar de la habitual, “Vista de disposición”.

Vemos que esa ventana dispone de una barra de formato superior para poder usar formatos avanzados. Para ello abriremos la pestaña “Vista de disposición”, seleccionaremos el texto a modificar y luego el formato a usar (negrita, color del texto, etc).

También en la vista de disposición podemos insertar tablas con el menú de la ventana Tabla. Podemos así mismo añadir listas numeradas o con viñetas.

Sin embargo a veces estos formatos se nos quedan cortos. Como no deja de ser texto HTML podemos utilizar sus etiquetas para dar formatos más personalizados.

A continuación se incluye un tutorial rápido de HTML con todo lo que podemos añadir en la vista de código HTML de dicho editor avanzado.

9.2. Tutorial rápido de HTML

9.2.1. Formato básico

Las etiquetas para un formateo básico a nivel de html son las siguientes:

- ``- Negrita
- `<i>`- Cursiva
- `<s>`- Tachado
- `<sub>`- Subíndice
- `<sup>`- Superíndice
- `<p>`- Nuevo párrafo
- `
`- Salto de línea
- `<hr>`- Dibuja una línea
- `<big>`- Texto grande

Para definir con más detalle la fuente, tamaño y tipo de letra usamos el siguiente código:

```
<p>
<font size="30" face="monospace" color="green">This is some text!</font>
</p>
```

Siendo face la forma de la fuente, tal como serif, monospaced, ...

También tenemos acceso a los encabezados, títulos de orden 1, 2. etc si escribimos:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

9.2.2. Caracteres especiales

Podemos insertar caracteres especiales en la vista html del nodo o la nota. Para ello podemos escribir:

código	aparece
<code>&larr;</code>	flecha a la izquierda
<code>&rarr;</code>	flecha a la derecha
<code>&uarr;</code>	flecha arriba
<code>&darr;</code>	flecha abajo
<code>&Gamma;</code>	letra griega gamma mayúscula
<code>&#9742;</code>	muestra un teléfono

Hay cientos de símbolos de este tipo por lo que no los incluiremos todos en esta introducción. Sin embargo puedes consultar la página de w3schools para más detalles.

9.2.3. Enlaces

```
<a href="URI">linktext</a>
```

Los enlaces se identifican por el tipo de URI (uniform resource identifier), que es el atributo de el anterior comando href. Se permiten los siguientes tipos de URI:

- Ancla en el mismo documento (e.g. #anchortext)
- Otro documento (e.g. myDoc.htm)
- Un documento en otro directorio (e.g. ../directory/myDoc.htm)
- Ancla en otro documento (e.g. myDoc.htm#anchortext)
- Ancla en un documento de otro directorio (e.g. ../directory/myDoc.htm#anchortext)
- Dirección web (http://...)
- Dirección de un documento local (file://...)
- Servidor FTP (ftp://...)
- Servidor Telnet (telnet://...)
- Correo electrónico (mailto:name@domain.xy)

Para definir un ancla en un documento escribiremos: `link anchor text`.

En lugar de un texto ("linktext" en el código de arriba) podemos poner una imagen de la siguiente forma:

```
<a href="home.htm">
<imgsrc="anImageFile.jpg" width="160" height="34" border="0">
</a>
```

9.2.4. Referencia a imágenes

Para mostrar una imagen teclearemos:

```

```

Las referencias al fichero pueden ser relativas como el ejemplo superior o absolutas como en C:/data/documents/myDoc/images/picture.jpg.

9.2.5. Tablas

La estructura de un documento con una tabla es la siguiente:

```
<table border="1" bgcolor="yellow" cellpadding="10" cellspacing="10" width=
"400">
<caption>Titulo de la tabla</caption>
<tr>
_____<td>
_____<p>
_____row 1, column 1
_____</p>
_____</td>
_____<td>
_____<p>
```

```

_____row 1, column 2
_____</p>
_____</td>
</tr>

<tr>
_____<td>
_____<p>
_____row 2, column 1
_____</p>
_____</td>

_____<td>
_____<p>
_____row 2, column 2
_____</p>
_____</td>
</tr>
</table>

```

Las opciones pueden colocarse también en las etiquetas `<td>` para afectar sólo a una celda. Su significado es el siguiente:

- `border`: anchura del borde.
- `bgcolor`: color del fondo de la celda / tabla.
- `cellpadding`: espacio entre el contenido de la celda y los bordes.
- `cellspacing`: espacio entre celdas.
- `width`: anchura de la celda / tabla.

9.2.6. Listas

Podemos mostrar listas mediante este código html:

```

<ul>
_____<li>
_____<p>
_____Item 1
_____</p>
_____</li>
_____<li>
_____<p>
_____Item 2
_____</p>
_____</li>
</ul>

```

Hay dos tipos de listas. Mediante elementos `` como el caso anterior y mediante elementos `` (únicamente sustituiríamos todos los `ul` por `ol`). Las listas `ul` se muestran con un punto negro delante de los elementos y las `ol` son listas numeradas.

9.2.7. Definiciones

Podemos añadir definiciones al estilo de las entradas de un diccionario con este código:

```
<dl>
  <dt>Coffee </dt>
  <dd>Black hot drink </dd>
  <dt>Milk </dt>
  <dd>White cold drink </dd>
</dl>
```

Siendo dt el elemento que definimos y dd su definición.

9.2.8. Definiendo estilos

De momento la versión de html que implementa Freeplane no admite ficheros de estilos, .css. Incluso aunque observemos en el código html del nodo el comando `<style>` no podemos cambiarlo pues al salir del panel de edición Freeplane lo borra.

9.3. Látex en los nodos

9.3.1. Qué es latex

Latex es un sistema avanzado de creación de documentos de alta calidad tipográfica orientado a la composición de artículos, libros y documentos científicos. Se compone por un conjunto de macros de TeX que originalmente fueron desarrollados por Leslie Lamport en 1984 para simplificar el uso de este lenguaje TeX creado por Donald Knuth.

Dada la calidad tipográfica de los documentos realizados en LaTeX, se la considera adecuada a las necesidades de cualquier editorial científica de primera línea.

Los documentos escritos en LaTeX constan de instrucciones incluidas con el resto del contenido del texto y sirven para darle formato. Dichas instrucciones comienzan con una barra invertida: “\”.

Si deseamos que un nodo sea mostrado como un nodo Latex deberemos comenzar a escribir con la expresión `\latex`, como se ve a continuación. Por ejemplo, si quisiéramos escribir la frase “**L^AT_EX** es un sistema de creación de documentos **de alta calidad tipográfica** orientado a la *composición de libros*” tendríamos que poner:

```
\latex \LaTeX es un \underline{sistema} de redacción de documentos \textbf{
  de alta calidad tipográfica } orientado a la \textit{composición de
  libros}
```

En esta frase encontramos varios comandos latex:

- `\LaTeX` Este comando imprime el logo de Latex.
- `\underline{texto}` sirve para subrayar.
- `\textbf{texto}` indica negrita.
- `\textit{texto}` pone el texto en cursiva.

9.3.2. Utilizando latex en nuestros mapas

Además de escribir texto simple, también podemos añadir estos códigos Latex para añadir más vistosidad a nuestros mapas. Lo único que tenemos que hacer es comenzar el texto con “`\latex`” en el nodo. No funciona ni el en detalle ni en las notas.

Freeplane usa el editor JLatexMath. Es una API de Java usada para renderizar el resultado de los comandos latex que puedes encontrar en <https://github.com/opencollab/jlatexmath>.

Por ejemplo, cuando escribimos en un nodo `\latex my_formula: x_2 = \frac{1}{2}` veremos algo así: $x_2 = \frac{1}{2}$. Es necesario encerrar la fórmula en símbolos “\$” para que se muestre en línea.

Otra forma para insertar una fórmula látex es establecer el formato látex en el nodo. Para ello, en el apartado Texto del nodo, en formato establecer látex. Cuando pinches F2 para editar el nodo te saldrá el editor de latex. Debe encerrarse la fórmula latex entre `[` y `]`. Por ejemplo, podemos teclear `\lim_{x \rightarrow \infty} \exp(-x) = 0` y la fórmula se mostrará correctamente.

También podemos incluir los comandos incluidos en un fichero si ponemos `\latexinput{<file>}`.

9.3.3. Algunos comandos

Latex es un tema muy extenso pues incluye miles de comandos. Algunos aparecen a continuación:

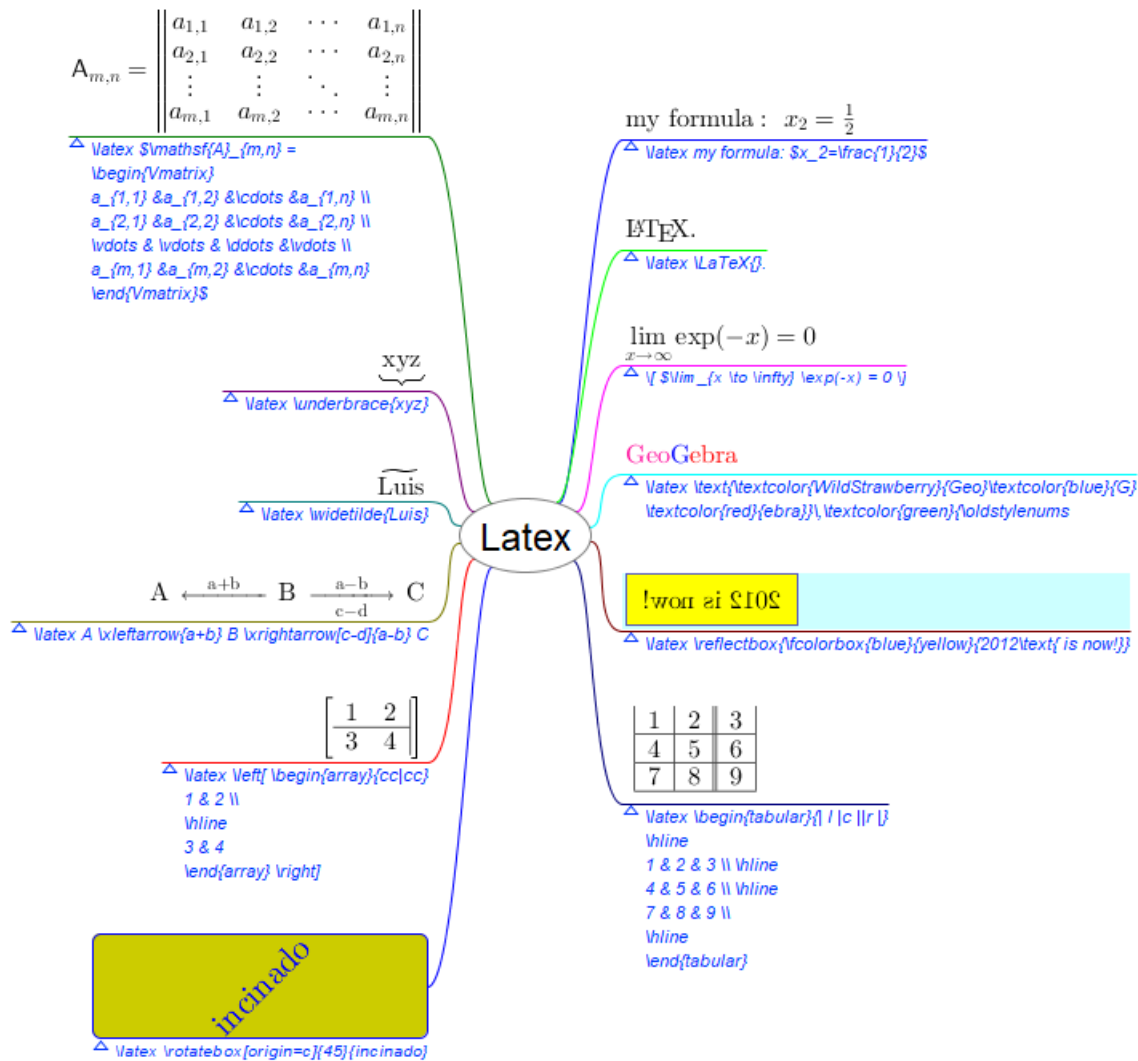
- Tamaño de letra: `\tiny` `\scriptsize` `\footnotesize` `\small` `\normalsize` `\large` `\Large` `\LARGE` `\huge` `\Huge`
- Alunas fuentes: `\mathcal{ABC}` `\mathrm{abc}` `\mathbf{abc}` `\mathsf{abc}` `\mathit{abc}`
- Color en el texto: `\textcolor{color}{...}` con colores como red, green, yellow...
- Definir recuadros de colores: `\fcolorbox{blue}{red}{2012\text{ jes ahora!}}`
- Girar estos recuadros: `\rotatebox{25}{\fcolorbox{blue}{red}{2012\text{ ya llegó}}}`
- Tablas: `\begin{tabular}{| 1 | c | r | } \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{tabular}` Otro
- Matrices: `\begin{array}{|c|c|} \hline \multicolumn{2}{|c|}{\text{Title}} \\ \hline x & y \\ \hline a & b \\ \hline c & c \\ \hline d & e \\ \hline \end{array}` Para
- funciones matemáticas:
 - Fracciones: `\frac{x}{y}`
 - Raíz cuadrada: `\sqrt[n]{x}`
 - Superíndice: `\wedge{x}`
 - Subíndice: `_ {x}`
 - Sumatorio `\sum_{k=1}^n`
 - Menor o igual: `\leq`
 - Símbolo de no igual: `\neq`
 - Símbolo dividido: `\div`
 - Puntos suspensivos: `\cdot`, `\cdots`, `\ldots`
- Letras griegas
 - `\alpha`, `\Alpha`
 - `\beta`, No igual `\Beta`
 - `\gamma`, `\Gamma`
 - `\rho`, `\Rho`
 - `\psi`, `\Psi`

Un ejemplo de L^AT_EX aparece en la figura 9.3.1 donde aparece el comando latex debajo de su renderización en el nodo.

9.3.4. Más comandos

En <http://en.wikibooks.org/wiki/LaTeX/Mathematics> podemos ver una lista de los comandos látex disponibles.

Podemos añadir nuevos comandos a Latex en Preferencias / Complementos / Macros Latex en común donde podemos escribir nuestros `\newcommand` que indica a latex la definición de un nuevo comando.

Figura 9.3.1: Ejemplo de L^AT_EX

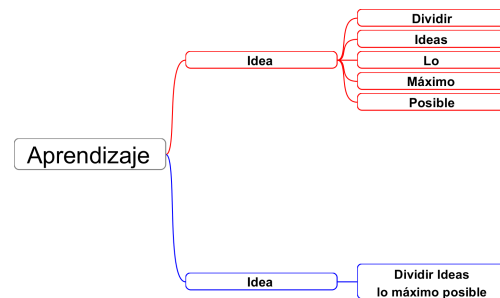
9.4. Aprendizaje mediante mapas mentales

El cerebro procesa mejor la información contenida en los mapas mentales si los nodos tienen todos el mismo ancho y si el texto está centrado en el nodo. Si procuramos poner en cada nodo una palabra ó secuencia de ideas le facilitaremos la tarea. Para ello elegiremos las palabras que provoquen que en nuestro cerebro se disparen las asociaciones de ideas. Estas palabras las denominamos “disparadores”. Esto hace que el conocimiento contenido en el mapa sea más limpio. Estas palabras “disparadores” son palabras cortas que contienen gran cantidad de conocimiento y que provocan que nuestro cerebro se llene de nuevas ideas. Podemos organizar el conocimiento del mapa utilizando estas palabras de forma que incluyamos toda la información dentro de sus nodos.

El cerebro puede dedicarse a una tarea en cada momento y es necesario hacerle las cosas lo más atractivas posible. Por todo ello dividiremos las ideas al máximo. Es mejor tener 5 nodos con el mismo padre que indiquen “Dividir”, “Ideas”, “lo”, “Máximo”, “Posible” que uno solo con la frase completa, tal y como se muestra en la parte superior de la figura 9.4.1. Para el cerebro procesar la rama superior es más sencillo que leer el párrafo completo de la rama inferior.

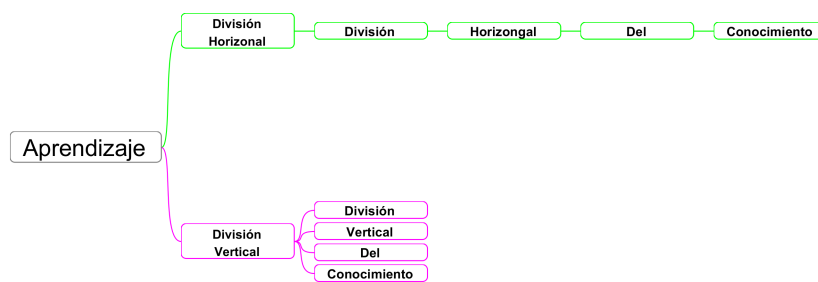
Una vez preparado el mapa tal y como se ha indicado sólo hay que abrir y cerrar las ramas del mapa para poder jugar con el conocimiento incluido en el mapa. Otra cuestión es si debemos realizar una división horizontal ó vertical (ver figura 9.4.2). Esto depende de la cantidad de información de las ideas y

Figura 9.4.1: Dividir Ideas



del espacio que tengas en la pantalla. En este sentido es más útil mantener Freeplane en pantalla completa para maximizar el área de trabajo.

Figura 9.4.2: División horizontal y vertical del conocimiento

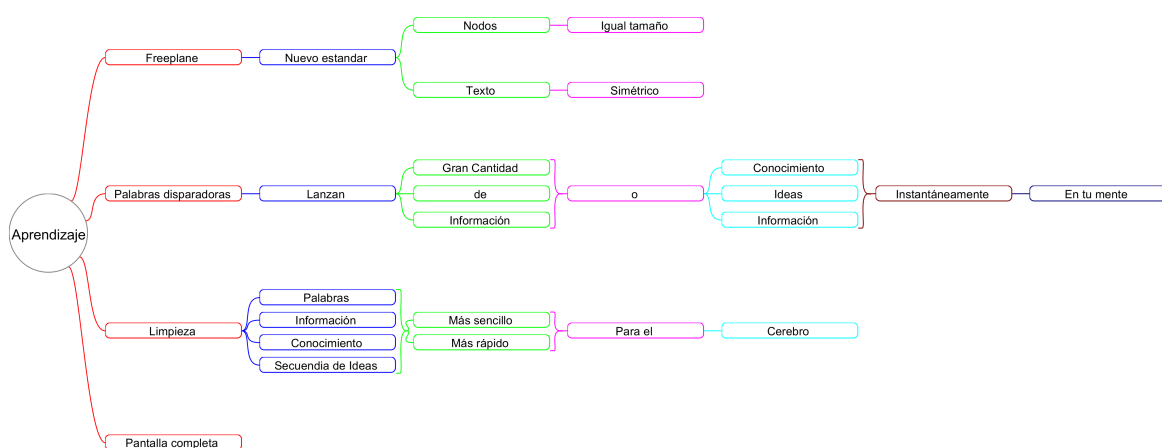


No debemos estresar al cerebro con nodos con párrafos complejos o con demasiados iconos ó imágenes. Favoreceremos el aprendizaje mejor si dividimos estos nodos en unidades más simples organizadas horizontal o verticalmente y utilizando nodos de igual tamaño y simétricos, con el texto en el centro y usando palabras que sirvan como disparadores.

Freeplane facilita esta división del conocimiento con opciones que encontramos en las preferencias de freeplane en la pestaña de “Comportamiento”, en la sección “Dividir nodo en palabras” y a la hora de hacer el mapa bajo el menú Editar / Nodo central:

- Dividir con 1 palabra en cada línea
- Dividir con 2 palabras en cada línea
- Unir nodos

Figura 9.4.3: Ejemplo de división del conocimiento



Capítulo 10

Formato avanzado

10.1. Estilos condicionales

Para poder definir estilos condicionales seguiremos el siguiente proceso:

1. Crearemos un estilo definido por el usuario al que asignaremos un nombre
2. Pincharemos en Formato / Gestionar estilos / Gestionar los estilos condicionales
3. Pincharemos en Nuevo
4. Doble click encima de la Condición. Se abrirá la ventana de Filtros (ver sección siguiente)
5. En esa pantalla estarán todas las condiciones que hemos definido. En particular permanecerá allí la condición “Siempre”.
6. De los desplegables superiores elegiremos la condición que deseamos componer
7. Una vez compuesta en los desplegables superiores, pincharemos en el icono añadir
8. Una vez la tengamos en la lista inferior la seleccionaremos. Si ya estaba dicha condición, únicamente la seleccionaremos.
9. Pincharíamos el botón “De acuerdo.
10. En la columna Estilo elegiremos entre los estilos guardados del mapa.
11. Ordenaremos nuestros estilos condicionales indicando si al aplicar alguno se detendrá la comprobación de sucesivos estilos, pues sino se harán todas de arriba a abajo.

Se pueden definir estilos en función del contenido del nodo, de algún atributo, el contenido, si tiene conector...Se accede a la ventana de definición de estilos condicionales entrando en Formato, Gestionar estilos, Gestionar estilos condicionales. La ventana que sale es una lista de estilos que se aplican en orden descendente: para cada nodo se aplica el primero y, si no tiene la marca de stop se aplicará el segundo y así sucesivamente. Sólo se detendrá cuando encuentre en un estilo la marca de parar.

Pinchando en «nuevo» creamos un nuevo estilo condicional. Aparece una línea con 4 partes:

1. Activa: indica que se aplicará a los nodos ése estilo condicional.
2. Condición: Condición que debe ser verdadera para aplicar el estilo.
3. Nombre del estilo que se aplicará
4. Parar: Si no está marcado, Freeplane continuará aplicando para ése nodo el estilo siguiente. Si está marcado se detendrá y no aplicará más estilos condicionales.

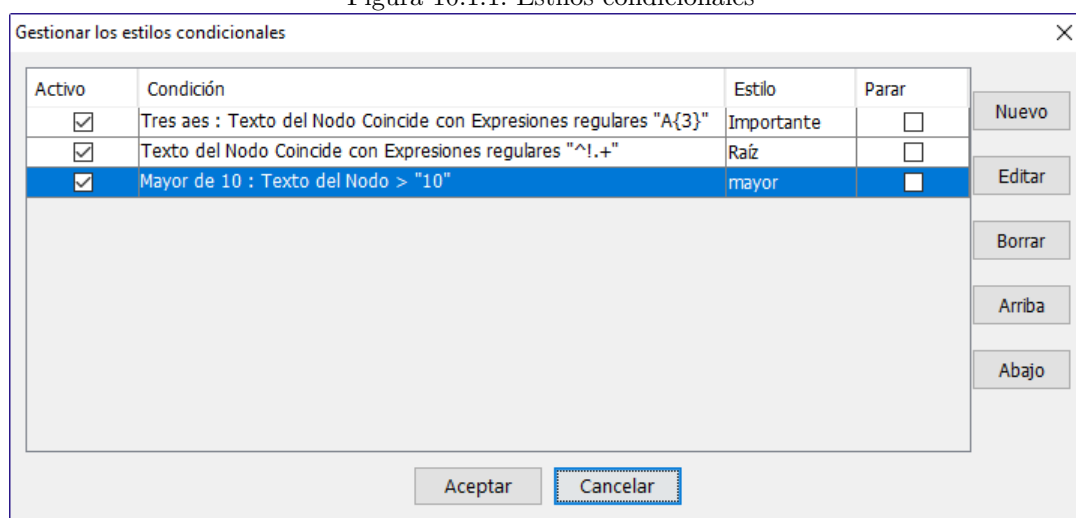
Pinchando dos veces encima de la línea que ha aparecido se abrirá la pantalla “Creador de filtros” con la lista de las condiciones disponibles (ver sección siguiente).

De esa lista elegiremos las que se aplican para esa condición y guardaremos. Debe hacerse notar en este punto que la pantalla que sale cuando pinchamos en editar (o hacemos doble click en la condición) es común para todos los nodos. Eso quiere decir que irá llenándose de condiciones a medida que definimos los estilos condicionales. Sin embargo sólo se aplicarán las condiciones seleccionadas antes de guardar.

Pinchando en la columna de estilo elegimos de la lista el estilo guardado que tomará nuestro nodo cuando dicha condición sea verdadera.

Y finalmente está la columna “Parar” que indica que después de aplicar esta condición no continuará evaluando más condiciones. Esto se usa para tener una jerarquía de condiciones con unas más fuertes que otras. Si esta condición está desmarcada se irán aplicando los formatos de arriba a abajo. De esta manera podremos combinar varios estilos en el caso de que el nodo cumpla dos ó más condiciones de la lista.

Figura 10.1.1: Estilos condicionales



En la figura 10.1.1 vemos tres estilos condicionales activos (el checkbox “Activo” está marcado). Todos se aplican de arriba a abajo (checkbox Parar sin marcar). Son los siguientes:

1. Se aplicará el estilo “Importante” si el nodo contiene tres Aes consecutivas.
2. Se aplicará el estilo “Raíz” si el estilo comienza con !.
3. Se aplicará el estilo “mayor” si el contenido del nodo es mayor de 10.

Capítulo 11

Presentaciones con Freeplane

Otra utilidad fantástica de freeplane es que podemos preparar el mapa para presentarlo con un proyector.

11.1. Opciones simples para mostrar el mapa

Lo más sencillo es usar la aplicación en pantalla completa (tecla F11). Hay tres entradas del menú Ver / configuración que nos pueden ser útiles:

- Centrar los nodos seleccionados: al seleccionar un nodo éste se moverá al centro de la pantalla.
- Spotlight: aparecerá el mapa en gris. Sólo se mostrará en claro el nodo seleccionado.
- Mostrar selección dentro de un rectángulo.

11.2. Presentaciones complejas en freeplane

En la barra lateral derecha hay una pestaña “Presentations” con la que podemos construir presentaciones más complejas. En freeplane nos referimos a presentar un mapa a ir mostrando diferentes partes de nuestro mapa de forma sucesiva. Se trata, por tanto, de ir seleccionando qué nodos van a ir apareciendo en pantalla. De esta forma podemos dar la impresión de que estamos construyendo el mapa en directo.

Las opciones de ese panel, por tanto, son opciones que nos permiten elegir qué nodos aparecen y si se muestran los nodos padre ó hijos de los seleccionados. Vamos a explicarlas.

El panel de presentaciones se muestra en la figura [11.2.1](#). Ese panel tiene cinco partes:

1. La parte superior, Presentations, nos permite tener varias presentaciones para un misma presentación.o mapa. Los botones que hay justo debajo nos permiten generar una nueva presentación, borrarla ó copiar una presentación que ya tenga el mapa.
2. La parte “Diapositivas” que está justo debajo nos permite controlar los diapositivas de la presentación actual. Como antes, tenemos debajo botones para añadir, copiar y borrar diapositivas.
3. Bajo la anterior está la parte “Slide content” que controla el contenido de la diapositiva actual. Con los botones inferiores podemos hacer que los nodos seleccionados sean los que aparecen en la presentación (primer botón) o bien añadir nuevos nodos a los que ya hay (segundo botón). Así mismo también podemos eliminar un nodo que ya hemos añadido o, con el último icono añadir los nodos a medida que los vamos pinchando en el mapa.
4. A continuación aparece una zona para ver dónde se muestran los nodos: en el borde izquierdo, centrados o en el borde derecho.
5. La zona de debajo permite hacer que se muestren sólo los nodos seleccionados. También podemos hacer que se muestren sus nodos padre ó hijos.

6. La zona que aparece bajo “Set filter” nos permite añadir nodos que cumplan los requisitos de un filtro que añadamos (por ejemplo, queremos que también se muestren los nodos que comiencen con “!” ó que contengan un icono concreto).
7. La parte “Set folding” nos permite controlar cómo se mostrarán el plegado de los nodos. Podemos hacer que se muestre el nodo con sus hijos, inicialmente plegados para que podamos desplegarlos en el momento de realizar la presentación.
8. Bajo la anterior tenemos los botones que nos permiten iniciar la presentación, avanzar de diapositiva y detenerla.
9. Con la penúltima parte podemos exportar la diapositiva actual o todas las de la presentación en imágenes. Esto nos permitiría añadirlas a un programa de presentación estándar. Sin embargo al hacerlo perdemos la posibilidad de interactuar con los nodos del mapa.
10. Al final hay un botón para cambiar las preferencias de Freeplane.

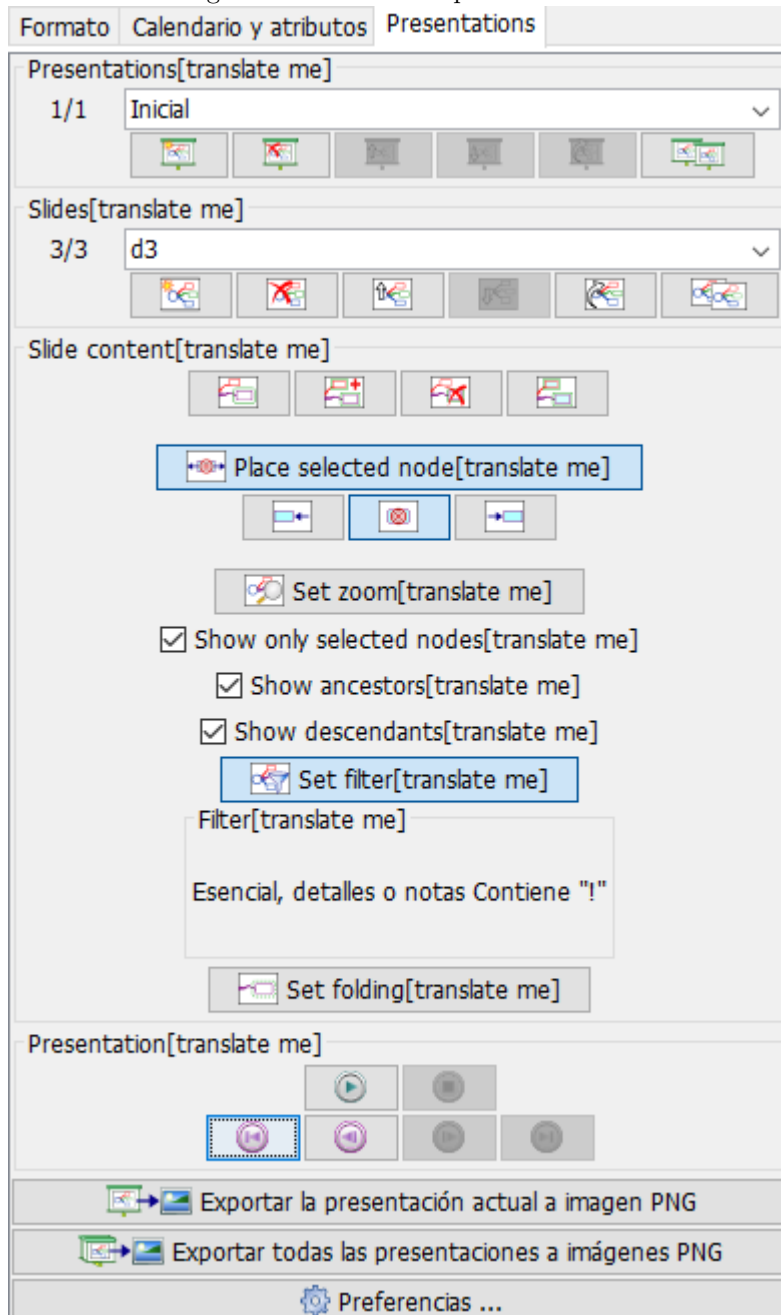
11.3. Diseño de presentaciones

Una vez sabemos las opciones que tenemos ya podemos centrarnos en diseñar nuestra presentación.

- Generar las diapositivas para mostrar cómo hemos ido construyendo el mapa
- Aplicar filtros de forma que las diapositivas se generen dinámicamente. Por ejemplo, al incluir filtros se mostrarán los nodos con un determinado icono en ellos.
- Diapositivas que muestren eventos sucesivos en una línea de tiempo

Consideramos que las opciones disponibles son muy amplias y que nos permiten una impresionante interacción con el mapa.

Figura 11.2.1: Generar presentaciones



Capítulo 12

Programando scripts

12.1. Introducción

Vamos a ver en este capítulo una de las características más destacadas de Freeplane: la capacidad de programar scripts.

Un script es como una macro en excel: un archivo de órdenes almacenado en un archivo o en los atributos de un nodo que nos servirá para automatizar las tareas más pesadas de Freeplane. En Freeplane en groovy, un lenguaje programado a objetos implementado sobre Java.

Para comenzar veremos un tutorial rápido de groovy.

12.2. Tutorial rápido de groovy

Groovy es un lenguaje de programación orientado a objetos implementado sobre la plataforma Java. Usa una sintaxis muy parecida a Java y comparte el mismo modelo de objetos, de hilos y de seguridad. Desde Groovy se puede acceder directamente a todas las API existentes en Java. El 99 % del código Java existente puede ser compilado mediante Groovy, y el 100 % del código Groovy es convertido en bytecode Java, y ejecutado en tu JVM de manera natural.

12.2.1. Datos básicos

- Booleano: `x = true` //ó *false*
- Números: `x = 1; y = 2.5; z = -8; val = "13".toInteger()`
- Cadenas: `x = "Hola_mundo"; s = "It's_time"` \\concatenación: `"hola_" + "mundo"`
- Listas: `mi_lista = [0, 1, 2, 3, 4]`
- Conjuntos: `def mi_conjunto = [0, 1, 2, 3, 4] as Set`
- Mapas: `def periodic = ['h': 'hydrogen', 'he': 'helium', 'li': 'lithium']`
- Rangos: `cuenta_atras = 10.0; minúscula = 'a'..'z' //1..3 == [1, 2, 3]`
- Bloque de código: `cod = {println "Hola_mundo"}` //Encerrado entre llaves

12.2.2. Operadores

operadores	significado
=	Asignación
!	Negación (complemento)
==, !=	Igualdad, desigualdad
+, -, *, /, %, **	Aritméticos
>, <, >=, <=	Relacionales
++, --	Incremental y Decremental
&&, , ?:	Condicionales
<<, >>, >>>, ~, &, , ^	Desplazamiento de bits
+=, -=, *=, /=, &=, =, ^=, %=, <<=, >>=, >>>=	Asignación compuesta

12.2.3. Palabras reservadas

abstract	as	assert	boolean	break
byte	case	catch	char	class
const	continue	def	default	do
double	else	enum	extends	false
final	finally	float	for	goto
if	implements	import	in	instanceof
int	interface	long	native	new
null	package	private	protected	public
return	short	static	strictfp	super
switch	synchronized	this	threadsafe	throw
throws	trait	transient	true	void volatile while

12.2.4. Sentencias de control

Expresión if

```
mayor = (x > 10)? true : false
```

Sentencia if-else

```
if(condition) {
    statement #1
    statement #2
    ...
} else if(condition) {
    statement #3
    statement #4
} else {
    statement #5
    statement #6
}
```

Sentencia switch

```
switch(expression) {  
  case expression #1:  
    statement #1  
    ...  
  case expression #2:  
    statement #2  
    ...  
  case expression #N:  
    statement #N  
    ...  
  default:  
    statement #Default  
    ...  
}
```

Bucle while

```
while(condition) {  
  statement #1  
  statement #2  
  ... }  

```

Bucle for

```
for(variable declaration; condition; Increment) {  
  statement #1  
  statement #2  
  ...  
}
```

Bucle for-in:

```
for(variable in range) {  
  statement #1  
  statement #2  
  ...  
}
```

Nota: pueden incluirse las sentencias break y continue para forzar la salida o nueva iteración del bucle.

12.2.5. Expresiones regulares

Una expresión regular es una cadena de caracteres alguno de los cuales tiene un significado especial:.

carácter	significado
.	Cualquier carácter
*	cero o más repeticiones del carácter o grupo anterior
+	uno o más repeticiones del carácter o grupo anterior
^	Comienzo de línea
\$	Fin de línea
[]	Grupo de caracteres
()	Subexpresiones
?	Coincide con el elemento anterior cero o una vez
{n}	Coincide con el elemento anterior n veces
{n,m}	Coincide con el elemento anterior al menos n y como mucho m veces
[a-z]	Cualquier carácter entre a y z
\w	Cualquier carácter de una palabra: a-zA-Z
\W	Cualquier carácter que no sea el anterior
\s	Espacio en blanco
\S	Cualquier carácter que no sea un espacio en blanco
\d	Un dígito
\D	Cualquier cosa que no sea un dígito
\t	Tabulación
\r	Retorno de carro
\n	Nueva línea
\	El carácter siguiente se trata como literal (p.e. * indica el asterisco)

Para indicar a groovy que la cadena que tenemos es una expresión regular en lugar de usar las comillas para encerrar dicha cadena usaremos `/cadena/`. De esta manera podemos escribir `/^P[\w]*r$/` representa a todas las cadenas que comienzan con P, contienen sólo letras y terminan con r. También podemos escribir `patron = ~/^P[\w]*r$/`. Esto último define la cadena patrón como de tipo Pattern.

Hay dos operadores: `find ==~` que nos permite ver si una parte de la cadena cumple con la expresión regular y `match ==~` que permite ver si la cadena completa cumple con dicha expresión. De esta forma si escribimos `cumple = mi_nombre==~/^P[\w]*r$/` la variable cumple será el booleano `true` si me llamo *Peter* ó *Podar* y `false` si me llamo *Luis* ó *Javier*.

12.2.6. Closures

Son variables que almacenan código. Ejemplos:

```
def cuadrado = {it**2}
println cuadrado(4)
//16
```

```
class Example {
    static void main(String[] args) {
        def clos = {param -> println "Hello ,_${param}"};
        clos.call('Javi');
    }
}
```

```
(10..1).step(1){println it}
```

```
def dailyRainfall = [ monday : 7,
                      tuesday : 5,
                      wednesday : 19,
                      thursday : 6,
                      friday : 11 ]
```

```
dailyRainfall.each { key, value -> println "$key_had_${value}mm_of_rain" }
```

12.2.7. Closures y listas

```
class Example {
  static void main(String[] args) {
    def lst = [11, 12, 13, 14];
    lst.each {println it}
  }
}
```

12.2.8. Métodos usados con closures

Método	Uso
<code>each()</code>	Itera sobre los elementos de la lista y se los pasa al closure uno a uno para su tratamiento
<code>find()</code>	Encuentra el primer valor de una colección que cumple el criterio
<code>findAll()</code>	Encuentra todos los valores de una colección que cumplen los criterios
<code>any()</code>	Itera sobre todos los elementos de la lista y devuelve verdadero si el predicado es válido para, al menos, un elemento
<code>every()</code>	Itera sobre todos los elementos de la lista y devuelve verdadero si el predicado es válido para todos los elemento
<code>collect()</code>	Itera sobre la colección convirtiendo cada elemento en un valor usando el closure como transformador

`each()`

```
(1..3).each {
  println "Number ${it}"
}
```

`find()`

```
class Example {
  static void main(String[] args) {
    def lst = [1,2,3,4];
    def value;
    value = lst.find {element -> element > 2}
    println(value);
  }
}
//3
```

`findAll()`

```
class Example {
  static void main(String[] args) {
    def lst = [1,2,3,4];
    def value;
    value = lst.findAll{element -> element > 2}
    value.each {println it}
  }
}
//3
//4
```

any() & every()

```

class Example {
  static void main(String[] args) {
    def lst = [1,2,3,4];
    def value;

    // Is there any value above 2
    value = lst.any{element -> element > 2}
    println(value);

    // Are all value above 2
    value = lst.every{element -> element > 2}
    println(value);
  }
}
//true
//false

```

collect()

```

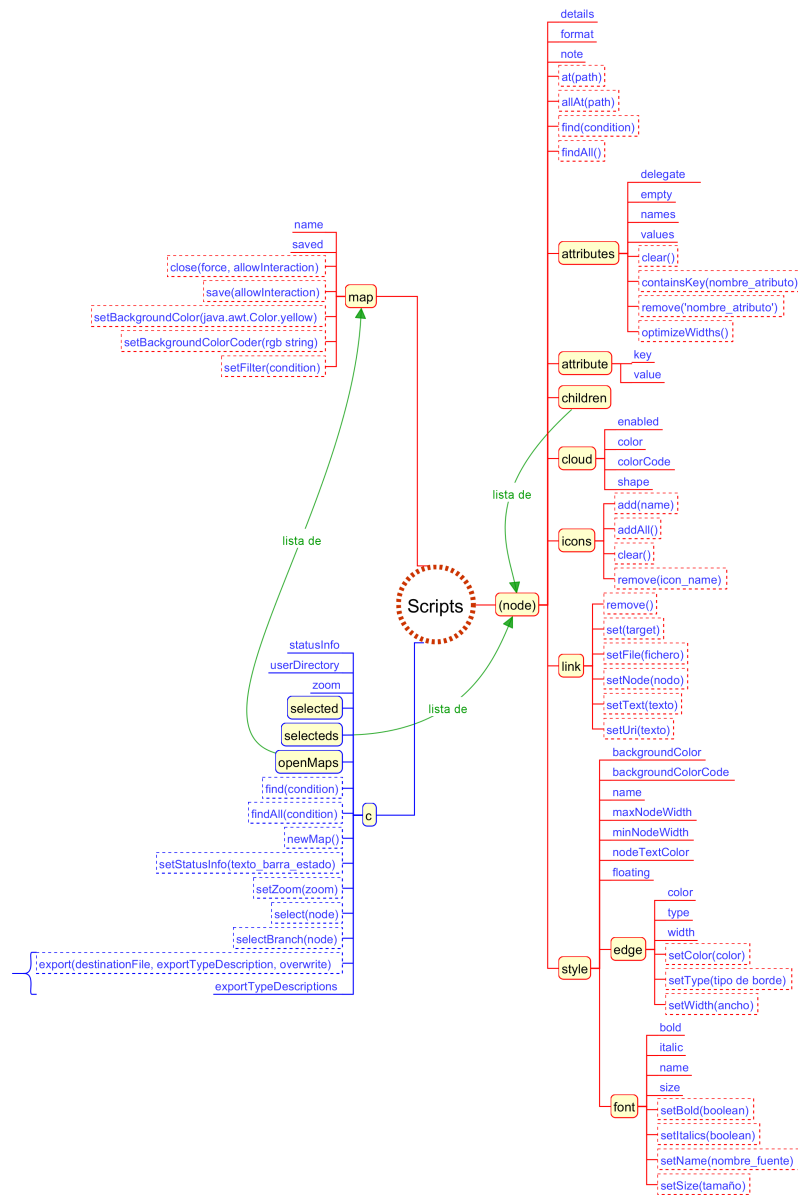
class Example {
  static void main(String[] args) {
    def lst = [1,2,3,4];
    def newlst = [];
    newlst = lst.collect {element -> return element * element}
    println(newlst);
  }
}
//[1, 4, 9, 16]

```

12.2.9. Listas

Método	Uso
lst.add()	Añade un nuevo valor al final de la lista.
lst.contains()	Devuelve true si la lista contiene el valor especificado.
lst.get()	Devuelve el elemento en la posición especificada de la lista.
lst.isEmpty()	Devuelve verdadero si la lista no contiene ningún elemento.
lst.minus()	Crea una nueva lista compuesta por los elementos de la lista original excepto aquellos especificado en la colección.
lst.plus()	Crea una nueva lista compuesta por los elementos de la lista original junto con aquellos especificados en la colección.
lst.pop()	Quita el último elemento de la lista.
lst.remove()	Quita el elemento de la posición especificada de la lista.
lst.reverse()	Crea una nueva lista con los elementos en orden inverso al de la lista original.
lst.size()	Obtiene el número de elementos de la lista.
lst.sort()	Devuelve una copia ordenada de la lista original.

Figura 12.2.1: Objetos de la API de Freeplane



12.3. Configuración de Freeplane para la ejecución de scripts

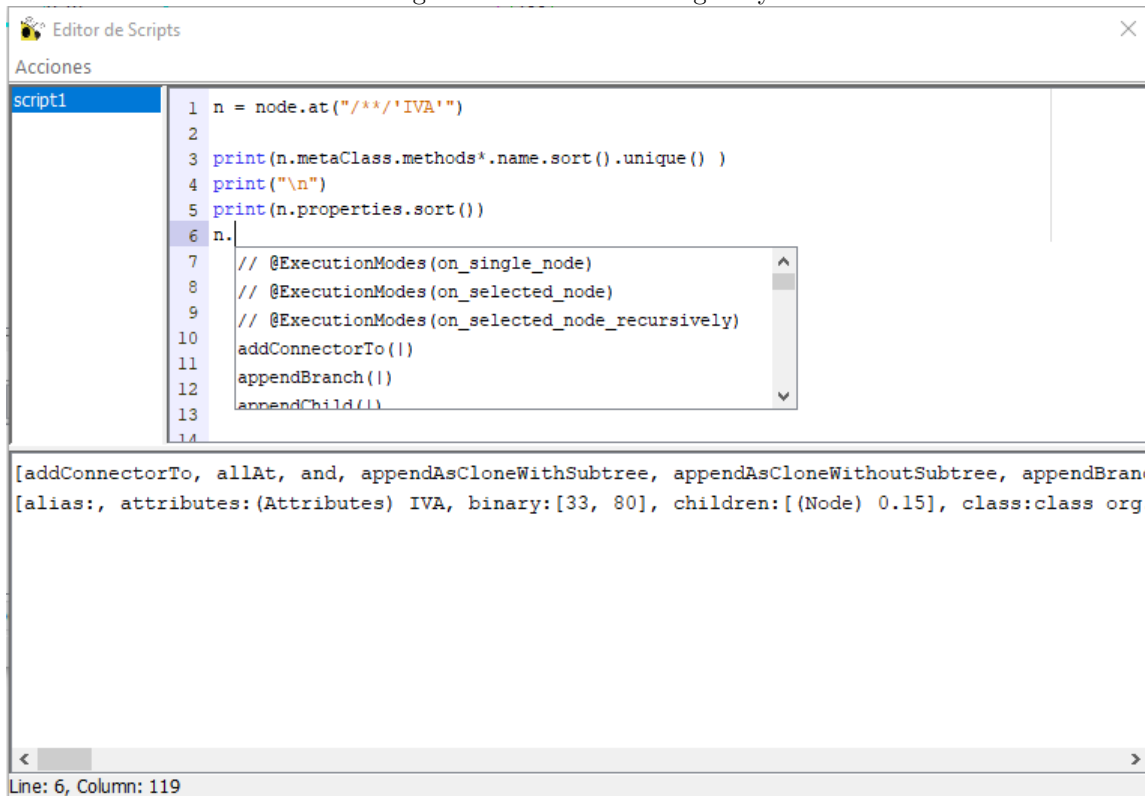
12.4. Comenzando a programar scripts

12.4.1. El editor de scripts

Podemos crear scripts de dos maneras:

- Con un editor externo creamos el código y lo guardamos en el directorio del usuario, subdirectorio "scripts". Al ejecutar Freeplane el script nos aparecerá en el menú Herramientas / Scripts.

Figura 12.4.1: Consola de groovy



- Con el editor de groovy incluido en Freeplane. Si vamos a Herramientas / Editor de scripts se abrirá la consola de groovy. Pinchamos en Acciones / Nuevo Script y ya podemos comenzar a teclear código.

Esta segunda forma nos permite acceder a todos los métodos de los objetos. Tecleando el objeto, un punto y la primera letra del método podemos pulsar las teclas `ctrl + <space>` y aparecerá una ayuda con los métodos que comienzan por esa letra. Si borramos en ese momento la letra obtendremos todos los métodos del objeto. En la imagen 12.4.1 podemos ver esta pantalla con la ayuda de los métodos para un nodo.

12.4.2. La API de freeplane

Para poder escribir scripts necesitamos conocer la estructura de objetos que se mantiene en la API de Freeplane (ver figura 12.2.1).

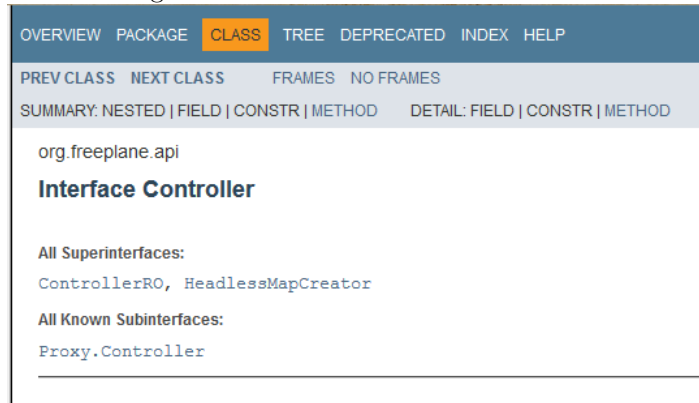
Podemos ver la documentación de la API en varios sitios:

- Abriendo el fichero `doc/api/index.html` en el directorio de instalación de Freeplane.
- Pinchando en en Ayuda → Api Generator. Se abrirá un mapa con toda la API de Freeplane.
- En la dirección <http://freeplane.sourceforge.net/doc/api/index.html>
- También es conveniente tener a mano la de groovy, accesible en internet en la dirección <http://docs.groovy-lang.org/latest/html/gapi/>.

En los scripts de Freeplane existen dos variables para facilitar la escritura de los scripts:

- `c` se refiere a `Proxy.Controler` en el mapa de la API generado por Freeplane.

Figura 12.4.2: Documentación de la API



- `node` se refiere a `Proxy.Node` en el mapa de la API generado por Freeplane.

Si accedemos a los documentos de la api en el fichero `index.html` todo esto es accesible en la clase `org.freeplane.api`. Vemos en la figura 12.4.2 la interface `Controller` con `Proxy.Controller` en el apartado All Know subinterfaces.

Lo más importante es conocer los métodos para las principales tareas de freeplane:

12.4.3. El primer script

La forma más sencilla de comenzar a realizar pruebas con groovy es entrando en el editor que tiene por defecto: Herramientas, Editor de scripts.

Si pinchamos nuevo, escribimos `node.text = "Hola"` y luego lo ejecutamos desde el menú, veremos cómo cambia el texto del nodo. Desde aquí ya podemos probar nuestro código.

12.4.4. Elementos de un nodo

`node.children` devuelve una lista con todos los hijos del nodo.

`node.parent` devuelve el padre del nodo.

`node.id` devuelve el ID único del nodo en el mapa.

`node.text` texto de un nodo. Puede ser el texto corto o en formato HTML dependiendo de si hemos añadido código html a mano.

`node.shortText` si queremos asegurarnos de no obtener el texto en formato html del nodo.

`node.details` detalle del nodo.

`node.note` las notas del nodo.

`node.attributtes.names` lista con los nombres de los atributos.

`node.attributes.values` lista con los valores de los atributos.

`node.map` es el mapa al que pertenece el nodo.

`node.next` el siguiente nodo buscando en profundidad(primeramente se va hacia los hijos antes de ir a los hermanos).

`node.previous` evidentemente, el contrario al anterior.

12.4.5. Operaciones con nodos

Buscar un nodo por su texto y seleccionarlo

```
nodo = node.find {it.shortText == "Texto_del_nodo_a_buscar"}
c.select(nodo)
```

Ir a la raíz desde cualquier nodo

```
nodo = c.find {it.isRoot()}
c.select(nodo)
```

Añadir nodos hijo

El siguiente código crea 10 hijos y los etiqueta con los números del 1 al 10

```
(1..10).each{
    node.createChild(it.toString())
}
```

Borrar nodos

El siguiente código borra todos los hijos de un nodo

```
node.children.each {
    it.delete()
}
```

Buscar un nodo y borrarle sus hijos

```
//MAL. Borra los hijos del nodo del script
nodo = c.find {it.shortText == "Nuevo_nodo"}
c.select(nodo)
node.children.each{it.delete()}
```

Añadir un enlace a un fichero en un nodo

```
node.link.file=new File("C:/Users/09767841V/AppData/Roaming/
Freeplane/1.5.x/scripts/SubirARaiz.groovy")
```

Añadir un enlace a una dirección web en un nodo

```
node.link.uri=new URI('http://www.google.com')
```

Fijar la anchura y el color de las líneas de un nodo

```
node.style.edge.setWidth(5)
node.style.edge.setClearColor("#336600")
```

O también

```
node.style.edge.setWidth(1)
node.style.edge.setColor(java.awt.Color.GREEN)
```

Pudiendo ser estos colores, según la documentación de [java.awt.Color](#) disponible: black, BLACK, blue, BLUE, cyan, CYAN, DARK_GRAY, darkGray, gray, GRAY, green, GREEN, LIGHT_GRAY, lightGray, magenta, MAGENTA, orange, ORANGE, pink, PINK, red, RED, white, WHITE, yellow, YELLOW.

Ver el nombre de los iconos de un nodo

```
println(node.icons.icons)
```

Nota: eso también funciona en las fórmulas. Si creamos un nodo y escribimos `=node.icons.icons` nos aparecerá el nombre del icono.

Mover un nodo al nivel anterior

```
node.moveTo(node.getParent().getParent())
```

Copiar un nodo y pegarlo como su hijo

```
nodo = node.text
detalles = node.details
enlace = node.link.text
iconos = node.icons

destino = node.createChild(nodo)
destino.details = detalles
destino.link.text = enlace
iconos.each{destino.icons.add(it)}
```

Nota: sólo copia el nodo con su detalle, enlace e iconos pero no copia sus hijos. Habría que crear una función recursiva para esto.

12.4.6. Configuración de Freeplane para la ejecución de scripts

Si nuestros scripts van a realizar labores con ficheros y directorios, operaciones de red o trabajar con otras aplicaciones debemos darles permisos desde Herramientas → Preferencias, en la pestaña Complementos.

Asimismo, si deseamos utilizar librerías externas también teclearemos la ruta, relativa al directorio del usuario, donde guardaremos estas librerías. Por defecto utilizaremos el directorio lib de la carpeta del usuario y escribiremos “lib” en ese cuadro.

12.5. Scripts de inicio

Si deseamos que freeplane ejecute un script cuando arranque podemos hacerlo si lo almacenamos dentro del directorio `<userdir>/scripts/init`. (ver apartado [14.1](#)).

12.6. Examinando propiedades y métodos de las clases de groovy

Podemos ver los atributos y métodos de una clase con el siguiente código:

```
n = node.at("/**/'IVA'")
print(n.metaClass.methods*.name.sort().unique() )
print("\n")
print(n.properties.sort())
```

12.7. Uso de librerías externas

Habitualmente las librerías utilizadas por Freeplane las encontramos en ficheros con la extensión `.jar`. Normalmente colocaremos dichas librerías en la carpeta `.lib` del directorio del usuario que, como sabéis, podemos mostrar si pinchamos en el menú Herramientas, Abrir el directorio del usuario.

Podemos modificar, sin embargo, la ubicación de nuestras librerías de dos maneras.

La primera si nos vamos a la sección Scripting de la pestaña Complementos en las preferencias de Freeplane. Hay una propiedad donde podemos definir la ruta para nuestros ficheros `.jar`.

La segunda es mediante el uso de la anotación de grape, `@grab` en nuestro script tal y como se muestra en el código siguiente.

```
@Grab('com.xlson.groovycsv:groovycsv:1.3')import static com.xlson.groovycsv.CsvParser.parseCsvdef
csv = '''Name,LastnameMark,AnderssonPete,Hansen'''def data = parseCsv(csv)for(line in data){ node
.createChild("$line.Name_$line.Lastname")}
```

Mediante esta anotación definimos las dependencias de nuestro código que serán automáticamente descargadas del repositorio remoto indicado.

12.8. Ejemplos

Imaginemos que queremos contar en el atributo «coste» de un nodo los valores de ese mismo atributo de sus hijos. Lo primero que tenemos que hacer es crear el nodo y uno de sus hijos con el atributo. En atributo del nodo escribimos la fórmula: `=children.sum(){it['coste'].num}`.

Sin embargo, cuando creamos un nodo hijo en esta fórmula aparece un error pues aún no existe el atributo «coste» (nota: podemos utilizar `.num0` y de esa forma no daría el error). para evitarlo añadimos el script `insertar.groovy` en el nodo seleccionando el nodo y luego herramientas, editor de scripts:

```
hijo = node.createChild('producto 1')
hijo['coste'] = 0
```

Lo cual ejecutado en el nodo crea un nodo hijo con el texto «producto 1» y un atributo «coste» de valor 0. Sin embargo vemos que queda bastante feo, pues se añade el script como otro atributo del nodo. Además, el texto y atributos creados son fijos.

Podemos mejorar el script de dos formas:

- Añadiendo el fichero `insertar.groovy` con el código del script en el directorio del usuario.
- Añadiendo un nodo INSERTAR con tres hijos (ver figura):
 - el primero con el texto del nodo
 - el segundo con el texto del atributo
 - el tercero con el valor del atributo

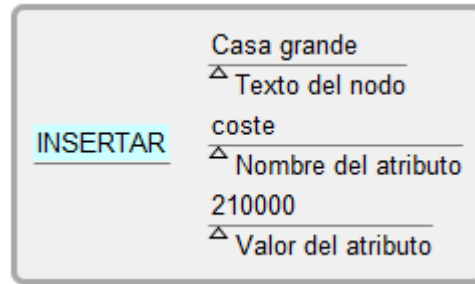


Figura 12.8.1: Nodo INSERTAR

Ahora el código del script se complica un poco, pues tenemos que acceder a los 3 hijos del nodo INSERTAR:

```

/* Inserta un nuevo nodo con los valores del nodo INSERTAR.
dicho nodo contiene:
---hijo 0: texto del nodo nuevo
---hijo 1: texto del atributo
---hijo 2: valor del atributo */

texto=c.find{it.text=="INSERTAR"}[0].children[0].value
atributo=c.find{it.text=="INSERTAR"}[0].children[1].text
valor=c.find{it.text=="INSERTAR"}[0].children[2].value

hijo=node.createChild(texto)
hijo[atributo]=valor

```

Lo único que nos queda es reiniciar freeplane y asignar un atajo a nuestro script mediante Herramientas, Asignar atajo. Primero nos pide que elijamos del menú el script (Herramientas, Scripts, Insertar) y después pulsamos la tecla de función a la que queremos asignar el script. Por ejemplo, F4.

Ahora cada vez que pulsemos F4 se insertará un nodo hijo del nodo que tengamos seleccionado cuyos valores son los del nodo INSERTAR.

12.8.1. Actualiza el atributo status de los nodos del mapa

Este script recorre todos los nodos actualizando el atributo “Status” de los nodos

```

1 // @ExecutionModes({on_single_node="main_menu_scripting/experiments"})
2
3 import org.freeplane.plugin.script.proxy.Proxy
4
5 // Get the root node of the mind map
6 Proxy.Node nodeRoot = node.map.root
7
8 /* 'findAllDepthFirst()' applied to the root node lists all nodes in the mind map
9 * sorted from the deepest to the highest level. This because you need to
10 * update the status experiments at the lowest level first, otherwise you will
11 * update the higher levels with statuses that are not up to date.
12 * 'each' means that every node in the list is processed by the statements
13 * between the curly brackets.
14 * Every 'it' between the curly bracket refers to the node from the list that
15 * is currently processed.
16 */

```

```

17 nodeRoot.findAllDepthFirst().each {
18     /* If the node has no children, it does not need updating: go to the next
19      * node in the list
20      */
21     if (it.children.size() == 0) return
22     /* If the node does not have the attribute 'Status', it is not an experiment
23      * and does not need updating: got to the next node in the list.
24      */
25     if (!(it.attributes.map.containsKey('Status'))) return
26     /* if none of the children have the attribute 'Status', there is nothing to
27      * update: go to the next node
28      */
29     if (it.children.count {it.attributes.map.containsKey('Status')} == 0) return
30     // Status is "in progress" by default
31     String strStatus = "in_progress"
32     /* If all of the children have status "not started", status changes to
33      * "not started"
34      */
35     if (blnAllChildrenHaveGivenStatus(it, "not_started")) strStatus = "not_started"
36
37     // If all of the children have status "done", status changes to "done"
38     if (blnAllChildrenHaveGivenStatus(it, "done")) strStatus = "done"
39     // Change the value of attribute 'Status' to the determined status
40     it['Status'] = strStatus
41     /* For testing purposes for every changed node, the name of the node is
42      * displayed together with the determined status. Once the script has been
43      * proven to be correct, the next 3 lines can be removed or commented out.
44      */
45     String strFeedback = 'Experiment_' + it.text +
46         '_has_been_updated_to_status_' + strStatus + '._'
47     ui.informationMessage(ui.frame, strFeedback, "Check_if_this_is_correct")
48 }
49
50 /* This method returns true if all children of the given node have the given
51  * status. Children that do not have the attribute 'Status' are ignored
52  */
53 boolean blnAllChildrenHaveGivenStatus(proxy.Node nodeParent, String strStatus) {
54     boolean blnAllHaveGivenStatus = true
55     nodeParent.children.each {
56         if (it.attributes.map.containsKey('Status')) {
57             if (it['Status'] != strStatus) {
58                 blnAllHaveGivenStatus = false
59             }
60         }
61     }
62     return blnAllHaveGivenStatus
63 }

```

Otra opción es crear la librería Experiment almacenando el siguiente código en el fichero Experiment.groovy que aparece a continuación y copiarlo en el directorio del usuario. Para usarla pondremos como valor del atributo “Status” de los nodos con hijos “=Experiment.updateStatus(node)”.

```

1  /* Library for updating the status of experiments
2  *
3  * Usage: enter the following as a value for the attribute 'Status' of an
4  * experiment node:
5  *
6  *     =Experiment.updateStatus(node)
7  *
8  * The experiment node should have one or more children with the attribute
9  * 'Status' having a value "not started", "in progress" and "done". If all

```

```
10  * children have status "not started", the value of attribute 'Status' will
11  * change to "not started". If all the children have status "done", the value
12  * of attribute 'Status' will change to "done". If the children have any other
13  * mixture of statuses, the value of attribute 'Status' will change to
14  * "in progress".
15  */
16
17  import org.freeplane.plugin.script.proxy.Proxy
18
19  def static updateStatus(Proxy.Node node) {
20      // Status is "in progress" by default
21      String strStatus = "in▯progress"
22      /* If all of the children have status "not started", status changes to
23       * "not started"
24       */
25      if (blnAllChildrenHaveGivenStatus(node, "not▯started")) {
26          strStatus = "not▯started"
27      }
28      // If all of the children have status "done", status changes to "done"
29      if (blnAllChildrenHaveGivenStatus(node, "done")) strStatus = "done"
30      return strStatus
31  }
32
33  /* This method returns true if all children of the given node have the given
34   * status. Children that do not have the attribute 'Status' are ignored
35   */
36  def private static blnAllChildrenHaveGivenStatus(Proxy.Node nodeParent,
37      String strStatus) {
38      boolean blnAllHaveGivenStatus = true
39      nodeParent.children.each {
40          if (it.attributes.map.containsKey('Status')) {
41              if (it['Status'] != strStatus) {
42                  blnAllHaveGivenStatus = false
43              }
44          }
45      }
46      return blnAllHaveGivenStatus
47  }
```

Capítulo 13

Fórmulas avanzadas

13.1. Introducción

En la sección [4.4](#) veíamos las fórmulas como expresiones matemáticas en las que aparecían algunos elementos de Freeplane tales como nodos o atributos. Sin embargo una fórmula es mucho más que una simple expresión matemática. Puede contener cualquier código de groovy que devuelva una expresión, tal y como introducíamos en dicha sección al teclear `=(parent.value=="calcular"? 2 + 3 : "2+3"`. Es esta una expresión de groovy que aparece definida en la subsección [12.2.4](#) del tutorial rápido de groovy.

Podemos encontrar numerosos ejemplos de groovy en las fórmulas. En particular son especialmente útiles los siguientes elementos:

- Operadores de groovy: `+`, `*`, `**`, `!=`, `&&...` (subsección [12.2.2](#))
- Elementos de los nodos: `children`, `parent`, `attributes...` (subsección [12.4.4](#))
- Expresión `if` (subsección [12.2.4](#))
- Closures (subsección [12.2.6](#))
- Métodos de listas: `size`, `sort`, `reverse...` (subsección [12.2.9](#)).
- Métodos que operan sobre listas: `find`, `findAll`, `each`, `collect...` (subsección [12.2.8](#))

13.2. Cómo referirnos a los elementos del mapa

Además de referirnos al nodo padre, a los hijos o a los atributos, también podemos utilizar el resto de los elementos del mapa. En la siguiente tabla podemos ver la forma de utilizar estos elementos

Descripción	Uso
Nodo con ID = ID_1907446118	=ID_1907446118
Texto del nodo padre del nodo ID_1907446118	ID_1907446118.parent.text
Valor numérico del padre del nodo	=parent.to.num
Lista con los hijos del nodo	= children.text
El primer hijo del nodo	=children[0]
Atributo Coste del nodo	=node['Coste']
Atributo Coste del padre del nodo	=parent['Coste']
Atributo Coste del primer hijo del nodo	=children[0]['Coste']
Lista con todos los iconos del nodo	=node.icons.getIcons()
Primer icono del nodo	=node.icons[0]
Detalles del nodo	node.details
Nota del nodo	=node.note
Enlace del padre	=parent.link.text

Tenemos que hacer unas puntualizaciones.

La primera: las listas en groovy se numeran comenzando por el cero. Eso quiere decir que el primer elemento de una lista es lista[0] y que lista[3] es el cuarto elemento de la lista.

Además, todos los nodos tienen un ID único al que podemos referirnos. Dicho ID podemos copiarlo pinchando con el botón derecho sobre el nodo y pinchando en “Copiar la identidad del nodo”. Otra forma de verlo es activarlo para que aparezca en la barra de estado que está en la parte inferior de la pantalla de Freeplane. Esto se activa en las preferencias, pestaña Apariencia. Si vamos a la sección Línea de Estado vemos que hay una casilla, “Mostrar la identidad del nodo” que podemos activar.

En este sentido decir que también podemos utilizar la nueva forma de referirnos a los nodos y que aparece explicada en la sección 13.8.

13.3. Funciones de conversión

En la tabla de la sección anterior vemos elementos un tanto “extraños” tales como value ó text. Vamos a proceder a explicarlos.

Cuando recuperamos un nodo mediante una fórmula tal y como =parent, Freeplane nos devuelve el texto del nodo que vemos como “(Node) 4” con la que no podremos realizar operaciones matemáticas.

Para obtener el texto del nodo, el número o la fecha que contiene tendremos que utilizar estas funciones de conversión. De esta manera teclearemos =10 * parent.to.num si queremos obtener el valor “40”.

Descripción	Uso
Texto del nodo	node.text
Texto del nodo	node.to.text
Valor numérico del nodo. Error si es un texto	node.to.num
Valor numérico del nodo. Si es un texto, cero	node.to.num0
Fecha del nodo	node.to.date

13.4. Funciones de listas

Muchas veces tendremos que hacer operaciones sobre los nodos. Por ejemplo, el número de hijos o la suma de todos los atributos coste. En esta sección se muestra cómo escribir estas expresiones.

Descripción	Uso	Ejemplo
Tamaño de una lista	<code>lista.size</code>	<code>=children.size</code>
Suma de una lista	<code>lista.sum()</code>	<code>=children.sum()</code>
Elemento máximo de una lista	<code>lista.max()</code>	<code>=[1, 2, 3, 4, 5].max()</code>
Elemento mínimo de una lista	<code>lista.min()</code>	<code>=[1, 2, 3, 4, 5].min()</code>
Contar los elementos de valor n de una lista	<code>lista.coutn(n)</code>	<code>=[1, 2, 3, 3, 3, 4].count(3)</code>
Primer elemento de una lista	<code>lista.first()</code>	<code>=children.first().value</code>
Último elemento de una lista	<code>lista.last()</code>	<code>=children.last().value</code>

13.5. Utilización de closures en las fórmulas

El hecho de que al referirnos a un nodo nos devuelva un valor de texto en lugar del valor numérico nos va a causar muchos problemas. En particular, no podremos obtener el valor máximo y mínimo de los hijos de un nodo tecleando sencillamente `=children.max()`.

En estos casos no nos queda más remedio que usar closures. Hay numerosos métodos de listas (de los que ya hemos hablado en las secciones 13.3 y 13.4) que se usan en conjunción con ellos.

La forma habitual es la siguiente:

```
=children.collect {
  _____it.to.num0
  _____}.max()
```

El método **collect** devuelve otra lista usando el closure `{it.to.num0}` como función de conversión. Esto significa que a cada elemento de la lista se le aplica esta conversión para calcular la lista que se devolverá. Una vez obtenida la lista numérica, ya podremos utilizar la función `max()` y `min()`. Como recordaréis, un closure no es más que un trozo de código que podemos usar directamente como en el ejemplo o guardar en una variable.

Esta forma de convertir una lista de valores de texto a otra con números nos va a ser útil en muchas ocasiones.

13.6. Groovy en las fórmulas

Realmente, una fórmula para Freeplane es mucho más que un conjunto de operadores y nodos. Podemos incluir cualquier código groovy, como veremos en los apartados que vienen a continuación (ver el apartado 12.2 para más detalle).

Por ejemplo, podemos buscar en el mapa el nodo con un texto determinado. Imaginemos que tenemos un mapa para calcular el precio de nuestras compras. Dicho nodo tiene tres atributos, uno con el IVA reducido, otro con el normal y otro con el especial. Para obtener en otro nodo estos IVAs podríamos teclear:

- IVA reducido: `=c.find{it.text==_IVA'}[0][reducido'].to.num`
- IVA normal: `=c.find{it.text==_IVA'}[0]['normal'].to.num`
- IVA especial: `=c.find{it.text==_IVA'}[0]['especial'].to.num`

Hay que tener en cuenta que `c.find` devuelve una lista de valores. Al poner `c.find{it.text==_IVA'}[0]` obtenemos la primera coincidencia.

Otro ejemplo de la potencia de las fórmulas de Freeplane es este. Podemos crear una variable que usaremos posteriormente en la misma fórmula.

Por ejemplo, imaginemos que en el nodo ID_790080551 se almacenan el iva. Podemos escribir una fórmula

```
=def x=_ID_790080551.to.num;_parent.to.num*(1+x/100)
```

en la que definimos la variable `x` que utilizamos posteriormente para calcular el precio final.

13.7. Ejemplos

Antes de comenzar a escribir nuestras fórmulas tenemos que hablar sobre el formato de los nodos, pues con algunos de ellos la fórmula se mostrará como texto normal:

- Estándar: si es texto, se muestra. Si es fórmula, se calcula. Si es \LaTeX , se dibuja.
- Texto: todo se muestra como texto y las fórmulas no se calculan ni el \LaTeX se dibuja.
- \LaTeX : el nodo contiene código \LaTeX

Para elegir el formato, desplegamos la barra de la derecha. En el tercer apartado, texto del nodo aparece un campo etiquetado Formato con una lista desplegable.

- Sumar en un nodo el valor de todos sus hijos: `= "hijos: " + children.sum()`
- Calcular el número de hijos de un nodo: `= "hijos: " + children.size()`
- Sumar los valores de los atributos coste de todos los nodos hijos: `=children.sum(0){ it['coste'].num0 }`.
- Obtener el triple del valor del nodo padre: `=parent.to.num*3`.
- Obtener en el nodo el valor del parámetro “precio” de nuestro nodo: `=node['precio']`.
- Sumamos todos los valores del atributo “precio” de todos los hijos de nuestro nodo: `=children.sum(0){ it['precio'].to.num }`.
- Calculamos 12^3 : `=Math.pow(12,3)`.
- Buscar en el mapa un nodo con un nombre determinado y obtener el valor del atributo “Coste”: `= c.find{ it.text=='MiNodo' }.collect{ it['Coste'].to.num }.first()`
- Calcular el día siguiente de la fecha del nodo padre: `=parent.to.date + 1`
- Obtenemos el máximo del atributo ‘val’ de los nodos hijos: `=children.collect{ it['val'].to.num0 }.max()`
- Suma de los cuadrados de los nodos: `=children.collect{ it*2 }.sum()`

13.8. Nueva forma de referirse a los nodos

13.8.1. Conceptos previos

Normalmente al añadir nuevos nodos a un mapa lo que hacemos es teclear el texto que deseamos que aparezca. Posteriormente también podemos añadir iconos, enlaces e imágenes. Y si deseamos referirnos al nodo copiamos su ID en el menú contextual con la opción “Copiar la identidad del nodo” y la ponemos en la fórmula de otro nodo.

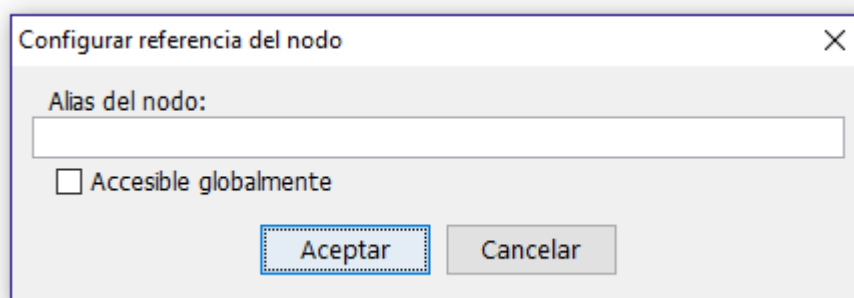
Sin embargo también podemos configurar la forma de la referencia de dicho nodo. Para ello en el menú contextual accesible pinchando en el nodo con el botón derecho elegimos la opción “Configurar la referencia del nodo” (También accesible en el menú Editar / Propiedades del nodo / Configurar referencia del nodo). Aparecerá la figura 13.8.1.

En esa ventana podemos:

- Configurar un nodo como “Globalmente accesible” o simplemente **global** cuando la casilla está marcada. En su creación todos los nodos la tienen desmarcada.
- Crear un **alias** para el nodo

Vemos que hay alias globales y no globales. Si marcamos la casilla de “Accesible globalmente” podremos acceder al nodo mediante la fórmula `=node.at("/:~mi_alias")` siendo el alias global “mi_alias”. Sin embargo si no marcamos dicha casilla deberemos dar toda la ruta. Por ejemplo, si el nodo con el alias “mi_alias” cuelga directamente de la raíz tendremos que escribir `=node.at("/~mi_alias")`.

Figura 13.8.1: Configurar la referencia del nodo



13.8.2. Acceso a nodos en fórmulas y scripts

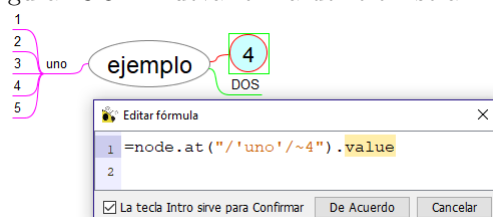
A partir de la versión 1.7, cuando escribimos una fórmula, además de usar su ID podemos referirnos a los nodos de otra manera.

Existen dos funciones de Freeplane que podemos utilizar para buscar los nodos: **node.at()** y **node.allAt()**. Con la primera obtenemos un nodo, provocando un error si no se obtiene ninguno o si encuentra varios nodos. La segunda nos proporciona una lista con todos los nodos encontrados. El uso de ambas es idéntico. Teclearemos `=node.at("referencia")`. “Referencia” puede incluir los siguientes elementos:

Referencia	Explicación
'texto del nodo'	Nodo cuyo contenido es el “texto del nodo”.
'texto...'	Nodo cuyo contenido comienza por “texto”.
'...'	Cualquier nodo hijo.
:~alias	Nodo cuyo alias global es “alias”. “Alias” no puede ser un número.
~alias	Nodo cuyo alias es “alias”. Debemos dar la ruta completa hasta el nodo con el alias.
/	Es la raíz del mapa.
~n	Siendo n un número, indica el hijo número n del nodo.
..	Indica el nodo padre
**/	Especifica cualquier nodo hijo. El nodo buscado puede estar en cualquier rama de los nodos hijo, a cualquier profundidad.
/**/	Colocado al comienzo de la referencia especifica cualquier subarbol de la raíz.

Explicación

Figura 13.8.2: Nueva forma de referirse al nodo



Significado: Estos son los elementos de la ecuación `=node.at("/'uno'/~4").value`:

- `=node.at(` : Iniciamos la ecuación
- `/` : Desde la raíz
- `'uno'` : Vete al nodo denominado uno
- `/~4` : Vete al cuarto nodo hijo de “uno”.
- `.value` : Pásame su valor.

13.8.3. Otras fórmulas

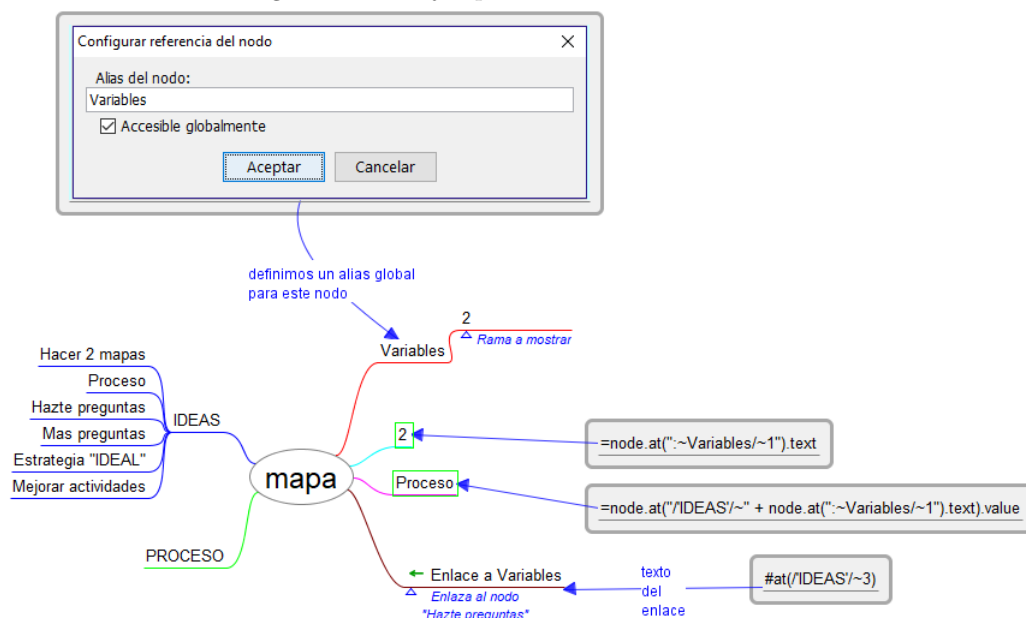
- `=node.at("~2").value` : segundo nodo hijo del nodo de la fórmula
- `=node.at("../'uno'/~5").value` : Vamos al nodo padre del que contiene la fórmula y buscamos el quinto nodo de “uno”.
- `=node.at("/'uno'/~4")` : Ir a la raíz, buscar el nodo cuyo texto es “uno” y obtener el valor del cuarto hijo (contado desde arriba).
- `=node.at("/**/IVA/~1").to.num` : buscar el nodo 'IVA' en todo el mapa y obtener su primer hijo (el valor numérico contenido en el nodo). Si no es numérico obtendremos un error. Para evitarlo podemos usar `to.num0` que devuelve el valor cero en ese último caso.
- `=node.at("..").value` : Valor del padre
- `=node.at("~1").value` : Valor del primer hijo
- `=node.at("/").value` : Valor de la raíz
- `=node.at("../'house'/**/~chairs")` : Nos vamos al nodo padre del actual y accedemos a un nodo con alias “chairs” que puede estar en cualquier rama de “house”.
- `=node.at("/**/'IVA'")['coste']` : Obtenemos el valor del atributo “coste” del nodo “IVA” ubicado en cualquier sitio del mapa.

13.8.4. Acceso a los nodos en los hiperenlaces

También podemos poner enlaces a otros nodos de esta manera. Así si queremos que en uno de los nodos haya un enlace al tercer hijo del nodo IDEAS cuyo padre es el nodo raíz pincharíamos en ese nodo con el botón derecho del ratón, enlaces y Añadir o modificar enlace (escribe). Luego escribiríamos `#at('/IDEAS'/~3)`.

En la figura 13.8.3 se ve un ejemplo gráfico de estas nuevas referencias.

Figura 13.8.3: Ejemplo de las nuevas referencias



Capítulo 14

Configurando Freeplane

14.1. El directorio del usuario

Es la ubicación donde residen todas las configuraciones y otros elementos particulares del usuario. Podemos acceder a él si desde freeplane pinchamos en Herramientas / Abrir directorio del usuario. Dentro de ese directorio podemos encontrar:

auto.properties: Fichero que contiene la configuración del usuario.

accelerator.properties: Fichero que almacena las teclas rápidas que ha definido el usuario.

auto.mmfilter: Filtros del usuario. Cuando creas un filtro para un mapa freeplane lo guarda en este fichero.

logs: Directorio donde están los archivos de registro. Son archivos en los que freeplane informa de situaciones comprometidas.

templates: Directorio para guardar las plantillas personales.

icons: Directorio para guardar los iconos personales.

scripts: Directorio para almacenar los scripts del usuario.

scripts/init: Directorio para guardar los scripts que freeplane ejecutará cuando arranque. Son todos los archivos con extensión .groovy.

libs: Las librerías externas se almacenan en este directorio

backup: Freeplane almacena periódicamente en este directorio copias de seguridad de los mapas que estamos editando.

resources: Directorio para almacenar otro tipo de recursos tales como plantillas personales de exportación o ficheros de idioma.

14.2. Opciones de configuración

No vamos a entrar en el detalle de cada una de las opciones de configuración disponibles pues son muchas. Únicamente explicar la pantalla de configuración.

Si accedemos a ella desde el menú Herramientas / Preferencias aparecerá una ventana con varias pestañas en la parte superior:

- Entorno
 - Instancia única de programa: si abre una ventana de freeplane por mapa o todos en la misma
 - Idioma

- Archivos: cantidad de archivos recordados, abrir último mapa al iniciar freeplane, plantilla usada para nuevos mapas...
- Cargar: acciones al abrir un mapa
- Guardar: acciones al guardar el mapa.
- Guardado automático: tiempo entre guardados, directorio...
- Exportar: resolución, usar tabuladores...
- Tipos de hiperenlaces: relativos ó absolutos
- Caché
- Programar actualizaciones
- Ayudas (surveys): mostrar consejos
- Informar de errores
- Apariencia
 - Estilo y apariencia: aspecto de la ventana de freeplane.
 - Línea de estado: mostrar la identidad del nodo ó no
 - Colores por defecto
 - Colores de la selección
 - Apariencia del nodo raíz. desde dónde comienzan las líneas de sus nodos hijo
 - Alisado
 - Iconos SVG
 - Límites de tamaño: máximas y mínimas longitudes de nodos, plegados, etc.
 - Conectores: configuración de los conectores
 - Clones: mostrar marcas en los clones o no
 - Ventana de edición avanzada del nodo: tamaño y botones
 - Vista de esquema: distancias
 - Iconos: mostrar iconos
- Por defecto
 - Nube: configuración de la nube por defecto
- Comportamiento
 - Comportamiento: como se comporta freeplane al plegar, desplegar y mover el ratón
 - Opciones de recordatorio: Al añadir una alarma en el calendario cómo nos la muestra
 - División del nodo en palabras
 - Unión de nodos
 - Formateo y análisis de los datos: reconocer la entrada de números y fechas, formatos etc.
 - Búsqueda: comportamiento al buscar
 - Editor de nodos en línea: mostrar edito y apariencia
 - Opciones del corrector ortográfico
 - Confirmaciones: al borrar, cortar o quitar nodos
 - Método de selección: si freeplane selecciona un nodo al pasar por encima o al hacer click
 - Rueda del ratón
 - Barra de desplazamiento
 - Tooltips: mostrar ayuda emergente
 - Deshacer: número de niveles del comando

- Editor de texto enriquecido: pegar como html
- HTML
 - Navegador: cuál usamos
 - Exportación de html
 - Importación de html
- Presentación
 - Configuración de la presentación
- Atajos y combinaciones del teclado
- Complementos
 - Fórmulas: activarlas o desactivarlas
 - LaTeX: nuevos comandos de latex, fuente de letra etc
 - Scripting: dónde ejecutar los scripts

14.3. Opciones de la línea de comandos

Podemos ejecutar freeplane desde la ventana de comandos utilizando esta sintaxis: `freeplane [options] [file1 [file2 ...]]`. Las opciones disponibles son las siguientes:

- `h|--help`: lista las opciones disponibles
- `-X<menukey>`: ejecuta el elemento del menú con la clave `<memkey>`. Usa el complemento `devtools` Use `devtools` para encontrar las claves de los menús
- `-S`: se detiene después de ejecutar los elementos del menú.
- `-N`: Fija la propiedad del sistema “`nonInteractive`” a “verdadero”
- `-U<userdir>`: Fija el directorio de configuración del usuario

14.4. Variables de entorno

`DEBUG=true freeplane.sh` Activa el modo debug de freeplane.

`FREEPLANE_JAVA_HOME=C:\Program Files\Java\jre1.8.0_131` Directorio de java de freeplane

`JAVA_HOME` Directorio raíz de java

`JAVA_OPTS=-Dorg.freeplane.userfpdir=$HOME/blah freeplane.sh` Fija las propiedades de java

Parte III

Desarrollando Freeplane

Capítulo 15

Colaborando en el desarrollo de Freeplane

Freeplane es un proyecto de programación complejo. Para su desarrollo utiliza numerosas tecnologías que han demostrado suficiente madurez.

15.1. Conceptos

Java: Lenguaje de programación de Freeplane.

Groovy: Lenguaje de programación utilizado para los scripts de Freeplane.

Git: Sistema de control de versiones.

Github: Repositorios para Git.

Gradle: Sistema de automatización para el desarrollo de programas.

Knoplerfish: Implementación libre de OSGi utilizada en Freeplane

OSGi: Sistema de publicación de componentes basado en la arquitectura de servicios.

Swing: Framework para implementar interfaces de usuario.

15.2. Configuración de un entorno de desarrollo de scripts.

Aunque podemos desarrollar scripts directamente con Freeplane, es mucho más fácil si utilizamos un entorno de desarrollo como eclipse, netbeans o jetbrains (idea). Las ventanas son evidentes: tendremos acceso a toda la potencia del entorno tal como resaltado de sintaxis, completado de código, detección de errores, depuración y navegación dentro del código. Para construir este sistema necesitaremos:

- Instalar groovy.
- Instalar gradle.
- Instalar nuestro entorno de programación preferido.
- Instalar los plugins necesarios para trabajar con estos elementos: groovy y gradle en el entorno elegido.
- Descargarnos una copia actualizada del código.
- Importar el código a nuestro entorno. Podemos importarlo como proyecto de gradle.
- Crear un proyecto de groovy para nuestro script.

- Vincular el proyecto anterior, Freeplane a este proyecto de groovy.
- Programar el script.

Puede ser que al hacer un build de gradle aparezcan errores indicando la imposibilidad de descargar los ficheros externos de los que depende el proyecto. Eso puede deberse a que en algunos sistemas no puede importar los certificados de algunos sitios https a los que se conecta gradle para resolver las dependencias. Para solventar el problema copiaremos los certificados de todos los sitios que informen del problema (error “peer not authenticated”) previamente exportados con el navegador, a la instalación de java que usemos, típicamente en el directorio `JDK_HOME/jre/lib/security`.

No es necesaria toda la complicación del control de versiones imprescindible para poder convertirnos en un desarrollador de Freeplane (ver a continuación.). Únicamente una copia del código de la versión de Freeplane que tengamos instalada importada en el entorno de desarrollo.

15.3. Desarrollar add-ons

Aún por escribir.

15.4. Desarrollando Freeplane

Freeplane es una aplicación libre de código abierto para la creación de mapas mentales que está programada en java usando OSGi (utiliza Knoplerfish como plataforma OSGi) y Java Swing como framework para su interfaz de usuario. Utiliza gradle como sistema para generar el código definitivo. Es un framework libre que permite automatizar las tareas tediosas de generar los ejecutables.

Freeplane fue lanzado por [Dimitry Polivaev](#) en 2007 como una rama del proyecto Freemind al que había contribuido durante mucho tiempo. Utiliza git como sistema de control distribuido de versiones. El código más actualizado está alojado en [github](#) y todos los distintos [ficheros](#) con versiones y plugins en [sourceforge](#). Asimismo, el [foro de discusión](#) y la [comunicación de bugs](#) también se realizan en sourceforge. Aunque el código también se publica en sourceforge, la versión más actualizada es la de github.

Tiene un ciclo de lanzamiento de 6 a 9 meses que comienza con el desarrollo de una versión alfa de la que sale una versión beta que genera ya la nueva versión definitiva. El desarrollo de la fase alfa dura de 3 a 6 meses y se publica en sourceforge como “freeplane preview”. Tras la publicación de las versiones previas se abre un plazo para fijar errores de unos 3 meses al final del cual se publica el RC que ya genera la versión definitiva.

Cuando se publica una versión beta de Freeplane es necesario probarla. Para ello la descargaremos y tras instalarla podremos realizar todas las pruebas que deseemos. Insertar nodos, notas, atributos, generar iconos de progreso o escribir fórmulas pueden generar algunos errores. Todos estos errores que hemos detectados tenemos que enviarlos a los desarrolladores en la página de [bugs](#) incluyendo la siguiente información.

- Qué estabas haciendo.
- El comportamiento observado.
- Los mensajes de error o advertencia obtenidos.
- Versiones de Freeplane y de java.
- Sistema operativo.
- Secuencia de pasos para poder reproducir el error.
- Resultado obtenido de esos pasos.
- Resultado esperado.

Los principales componentes tienen su propio directorio dentro del código fuente. Así podemos encontrar:

- freeplane

- freeplane_ant
- freeplane_api
- freeplane_debughelper
- freeplane_framework
- freeplane_mac
- freeplane_plugin_bugreport
- freeplane_plugin_formula
- freeplane_plugin_jsyntaxpane
- freeplane_plugin_latex
- freeplane_plugin_openmaps
- freeplane_plugin_script
- freeplane_plugin_script_test
- freeplane_plugin_svg

15.5. Qué necesitaremos

Si tenemos claro que queremos meternos de lleno en las tripas de Freeplane necesitaremos los siguientes elementos:

- Un entorno de desarrollo como eclipse, netbeans o IntelliJ Idea.
- Un entorno de desarrollo de java, JDK.
- Git.
- Gradle.
- Cuenta de Github.
- Diversos plugins para nuestro entorno de desarrollo (groovy, git, gradle...).

Los pasos básicos comprenden:

- Instalación de todo esto, incluyendo el entorno de desarrollo, JDK, git, gradle y demás elementos.
- Descarga del código más reciente del repositorio general.
- Crear una rama para nuestro código.
- Realizar el desarrollo.
- Mezclarlo con la rama general y subirlo al repositorio.

15.6. Obteniendo el código más reciente.

El equipo que desarrolla Freeplane utiliza git como sistema de control de versiones y trabajo colaborativo. Por lo tanto es imprescindible tener un conocimiento básico de este sistema para poder comenzar el proceso pues es la puerta de acceso al código así como la base de todo el desarrollo. Lógicamente, para trabajar de forma colaborativa con el resto de los desarrolladores es imprescindible el uso ágil de este sistema de versiones. Los pasos básicos de todo el desarrollo son los siguientes:

1. Instalación de git en nuestro equipo. Podemos obtenerlo de [git](#).
2. Configurar nuestro usuario y correo globales que se usarán cuando comencemos a utilizarlo.
3. Crear un repositorio local.
4. Descargarlos una copia del repositorio remoto en github.
5. Hacer un build desde gradle para generar el proyecto.
6. Crear con git una rama local para trabajar nosotros.
7. Realizar el desarrollo que deseemos. Configurar en git qué ficheros añadimos o borramos.
8. Enviar nuestra rama al repositorio global.
9. Comprobar si hay desarrollos posteriores de otros compañeros en el repositorio y descargarlos en nuestra copia local.
10. Cuando finalices el desarrollo local debes unir tu rama con la rama principal y enviarla al repositorio remoto.

Existe una guía oficial en español sobre git que puedes obtener en su sitio web dentro de documentation, book y seleccionando Español como idioma. Podrás descargarlo en formato pdf o leerla directamente de su sitio web.

15.7. Importándolo

Según el [mensaje del foro](#), las instrucciones para importarlo como proyecto de gradle son las siguientes:

- Download Freeplane source and extract to a directory freeplane_src
- Click/select File >Import >Gradle Project >Next >Next
- Specify the path to the freeplane_src directory in 'Project root directory' and click Finish. All projects are now imported as Gradle projects
- Delete freeplane_ant and freeplane_mac, close all other imported projects except for freeplane_src (i.e. the project for the main freeplane_src directory)
- Click/select Run >Run configurations
- Create a new Gradle project, e.g. 'freeplane_src - clean build cleanEclipse eclipse'
- In Gradle Tasks enter: clean build cleanEclipse eclipse'
- Click/select Workspace... >freeplane_src >OK
- Click Run and wait till Gradle build has finished
- Now open all closed projects and guess what: they are now Java projects.

Además, otras ideas del foro: Specifically for Freeplane development you can generate eclipse projects using gradle command gradle clean build cleanEclipse eclipse

and use preconfigured launch configurations for debugging. Unfortunately we have not created such projects and configurations for IntelliJ, so eclipse is the only tool of the choice. Yes, you have a burden of downloading and installing the tools preferable in the right versions (for eclipse I recommend "eclipse for eclipse developers"), but using them you follow standard path and can get support from big communities.

15.8. Comenzando el desarrollo.

El proyecto utiliza como constructor gradle que es un de automatización de construcción de código. Ha sido diseñado para proyectos grandes y nos permite modificaciones incrementales. Determina qué parte del código ha cambiado y sus dependencias para determinar qué tareas son necesarias para la construcción del sistema.

Lo primero que tenemos que hacer una vez descargado el código del repositorio principal es lanzar el builder de gradle para generar el proyecto.

Al principio se recomienda tratando de resolver bugs sencillos y desde ahí ir aumentando la complejidad. Sin embargo, a la hora de mezclar nuestra rama con la master conviene consultar a algún desarrollador experimentado, sobre todo al principio.

Los desarrolladores utilizan una [lista de correo interna](#). Si deseamos colaborar activamente tendremos que registrarnos y presentarnos. Es un buen sitio para decir en qué estamos trabajando.

Parte IV

Apéndices

Capítulo 16

Glosario

Alias: Se ha especificado un alias para el nodo en la ventana de configurar referencia del nodo. Figura: [13.8.1](#).

Atributos: Características o variables que podemos añadir a un nodo. Tienen un nombre y un valor. Figura [2.5.1](#).

Borde: Conexión entre una idea y las subideas que parten de ella. Figura [2.5.1](#).

Colapsar un nodo: Pinchar en el círculo de expansión de un nodo para que se oculten sus hijos. Figuras [2.4.1](#) y [2.4.2](#).

Conector: Conexión entre nodos arbitrarios del mapa. Figura [2.5.1](#).

Enlace: Unión a un recurso interno o externo que se añade a un nodo. En particular, los hiperenlaces se añaden a los nodos para que, al pinchar sobre ellos nos lleven a recursos de la Web. Figura [2.5.1](#).

Estilo: Conjunto de formatos a los que ponemos un nombre.

Estilo predefinido: Estilo que viene incluido en la aplicación

Estilo definido por el usuario: Estilo creado por el usuario.

Expandir un nodo: Pinchar en el círculo de expansión que muestra freeplane en los nodos de forma que se muestren sus hijos.

Exportar: Guardar el mapa en un formato distinto al de Freeplane para que pueda ser utilizado por otras aplicaciones.

Filtro: Condición de búsqueda que crea el usuario para buscar nodos coincidentes con dichos criterios.

Fórmula: Expresión que calcula freeplane de forma dinámica. Puede incluir referencias a otros nodos, operadores aritméticos, funciones matemáticas y funciones de listas como las del apartado [12.2.8](#).

Formato: Apariencia del elemento del mapa que podemos configurar de forma estática.

Formato condicional: Apariencia dinámica del nodo que Freeplane modifica en función de otros elementos.

Formato directo: Aquel formato que aplicamos a un nodo de forma manual

Formato automático: Formato que asigna Freeplane en función de la ubicación de un nodo en el mapa

Groovy: Lenguaje de programación.

Icono: Pequeño dibujo que añadimos a los nodos y que aparece antes de su texto. Figura: [2.6.1](#).

Idea: Palabras que muestran los conceptos que deseamos añadir a nuestros mapas.

Idea central: Concepto ó idea principal que se coloca en el centro del mapa que muestra el asunto que deseamos analizar.

Importar: Introducir información en un mapa procedente de otra aplicación.

Latex: Sistema de composición de textos orientado a la creación de documentos de alta calidad tipográfica.

Mapa condeptual: Diagrama en forma de red de conceptos que muestra la relación que existe entre ellos.

Mapa mental: Diagrama circular que parte de una idea central a partir de la cual se van añadiendo otras radialmente.

Nodo: Cada uno de los puntos fijos en el mapa mental y que se deriva de la idea central. Figura [2.5.1.](#)

Nodos clon: Nodos que realmente son el mismo nodo. Cualquier cambio en uno de ellos modifica todos los demás. Alterar la estructura o borrar uno de ellos lo haría en el resto. Figura [2.5.1.](#)

Nodo gobal: Nodo marcado como global en la ventana de configurar referencia del nodo (ver figura [13.8.1](#))

Nodos hermanos: Aquellos nodos que comparten el nodo padre. Figura [2.5.1.](#)

Nodo hijo: Nodo que parte de otro más importante. Figura [2.5.1.](#)

Nodo padre: Aquel nodo del que parten nuevos enlaces. Figura [2.5.1.](#)

Nodo raíz: Nodo ó idea central a partir de la que surgen todas las demás. Tema fundamental del mapa. Figura [2.5.1.](#)

Nodos sumario: Agrupan varios nodos hermanos en uno Figura [2.5.1.](#)

Nota: Texto complejo que podemos añadir a un nodo. Contiene explicaciones, aclaraciones o textos largos en formato HTML. Figura [2.5.1.](#)

Nube: Recuadro sombreado alrededor de un grupo de nodos. Figura [3.5.1.](#)

Plantilla: Conjunto de estilos, nodos, bordes y formatos de un mapa que guardamos en el directorio del usuario.

Presentación: Conjunto de diapositivas que contienen información sobre un tema. Figura [11.2.1.](#)

Ramas: Líneas que conectan una serie de ideas dependientes.

Script: Código que podemos incluir en Freeplane para automatizar tareas. El lenguaje nativo de Freeplane es Groovy aunque también podemos añadir librerías para poder utilizar otros lenguajes.

Nota: No se incluye en el glosario los conceptos de groovy tales como listas, sentencias o closures.