

Anteproyecto

Editor de mapas mentales online

con

HTML5 y Javascript

Autor: José Luis Molina Soria

Fecha: 16 de noviembre de 2012

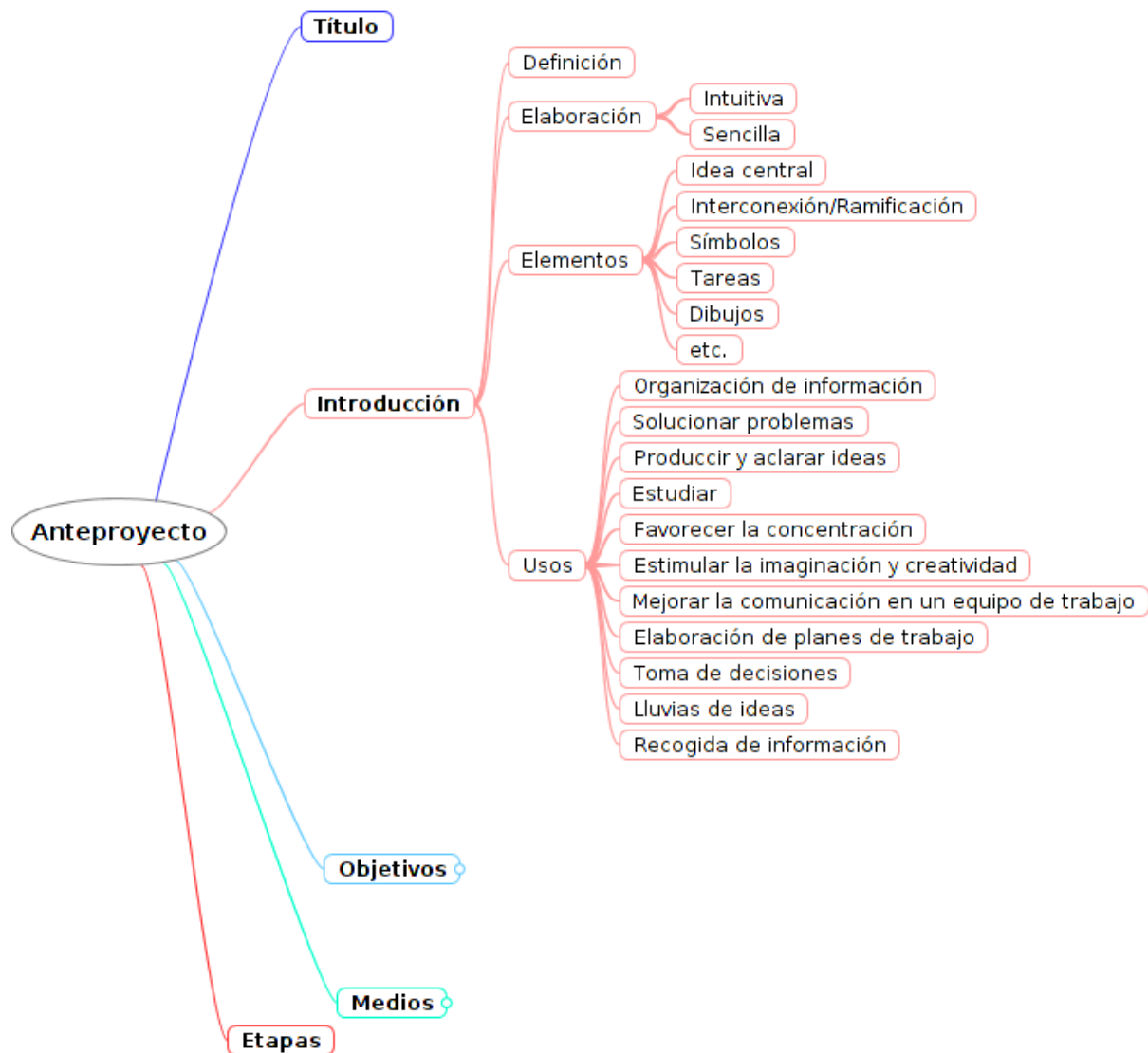
Introducción.

Popularizado por el psicólogo británico Tony Buzan en la segunda mitad del siglo XX, los mapas mentales han demostrado, a lo largo de este tiempo, su eficacia y versatilidad. Para extraer, memorizar, anotar, expresar y/o cartografiar ideas. Los mapas mentales permiten generar nuevas ideas, conectar o expandir otras. Contribuyen al desarrollo de la memoria, ayuda a organizar, analizar, entender, pensar, anotar, conocer, aprender, facilita el recuerdo y la comprensión de ideas complejas.

Entre otras cosas los mapas mentales son utilizados para:

- Organizar información.
- Solucionar problemas.
- Producir y aclarar ideas.
- Estudiar.
- Favorecer la concentración.
- Estimular la imaginación y creatividad.
- Mejorar la comunicación en un equipo de trabajo.
- Elaboración de planes de trabajo.
- Toma de decisiones.
- Lluvias de ideas.
- Recogida de información.
- Elaboración de sitios webs.

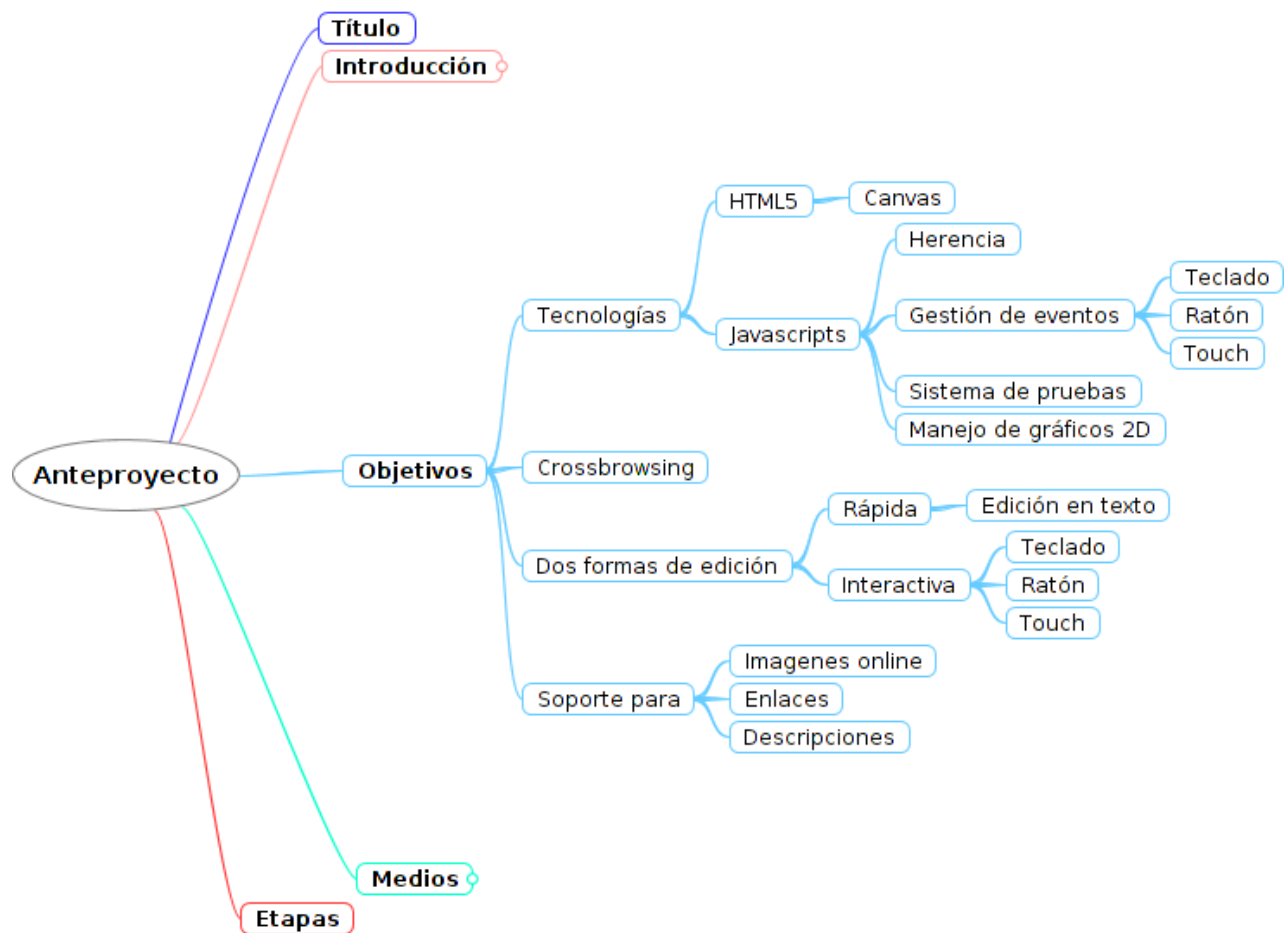
La elaboración de un mapa mental debe ser una actividad sencilla e intuitiva, sólo deberíamos necesitar arrancar de una idea central, de la cual vamos ramificando asociando o interconectando símbolos, palabras, tareas o dibujos.



En definitiva, se trata de un diagrama radial que nos permite a una persona, o grupo de ellas, plasmar su percepción sobre un tema, o idea, mediante la asociación de conceptos palabras y/o imágenes. En cierta forma son una extensión de los clásicos diagramas de llaves pero de forma más libre, creativa y fácil de modificar, orientada a una mayor uso del hemisferio derecho, el de la creatividad: color, formas redondeadas, facilidad para la colocación espacial de los elementos... usando siempre palabras clave o frases muy cortas, sugerencias.

Objetivos.

El principal objetivo de este proyecto es la creación de un editor de mapas mentales online. El editor, *frontend*, debe ejecutarse completamente en el navegador cliente. Para ello, vamos a utilizar como lienzo de dibujos el canvas de HTML5 y Javascript como lenguaje de desarrollo.



El editor soportará principalmente dos modos de trabajo:

- Edición rápida: mediante un cuadro de edición, se podrá ir introduciendo los nodos de forma que, cada nodo estará en una línea y que la indentación indicará el nivel de profundidad en el mapa. El mapa se actualizará con cada nodo introducido de forma automática.
- Edición interactiva: el usuario podrá navegar por el diagrama con los cursores partiendo desde la idea central. Interactuará con el diagrama de forma que, dependiendo del nodo

en el que se encuentre y la acción que realice podrá insertar, modificar, anotar, incluir imágenes o enlaces.

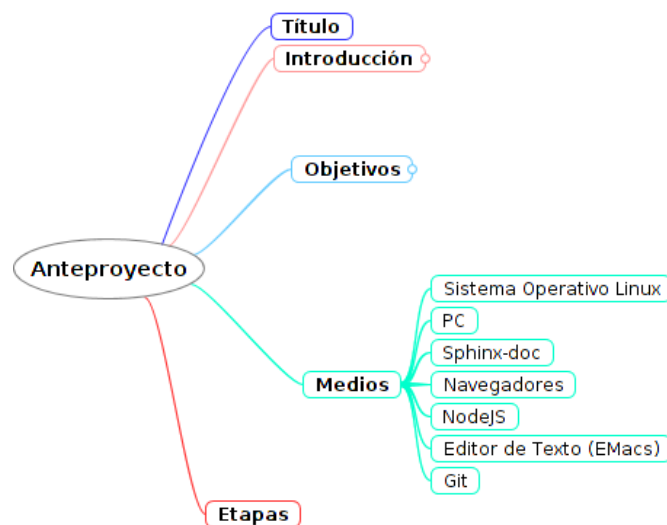
Esta fuerte interacción, provoca que dentro de los objetivos del proyecto, se encuentre la elaboración de una extensa librería JavaScript.

En todo momento, y en pos de una aplicación lo más standard posible, se seguirá las especificaciones de la World Wide Web Consortium (W3C).

Como objetivo principal está pues, la universalidad, independencia de sistemas y la inmediatez de uso, sin instalación, siempre actualizada, e incluso la posibilidad de uso en forma local con cualquier navegador actual que sigue el estándar HTML5. Entre las posibles plataformas de uso se tratará de incluir las plataformas táctiles, especialmente los tablets.

Medios.

- Sistemas operativos Linux, iOS, OSX, Windows.
- Org-Mode para gestión de *Todo's*.
- Navegadores web estándares para pruebas en todas las plataformas y servidor web.
- NodeJS a través de CLI para soporte al desarrollo Javascript.
- Sphinx o JSDoc como sistema de documentación de código fuente.
- Sistema de gestión de versiones git y github.



Metodología de análisis (UML-WAE).

Como metodología de análisis utilizaré el estándar de facto UML y su extensión al desarrollo e interfaz web, WAE. UML desde el punto de vista estructural o estático en el cual se plasmarán las estructuras estáticas compuesta por objetos, atributos, operaciones y relaciones y el otro punto de vista, dinámico o de comportamiento que nos permitirá establecer y especificar las colaboraciones entre las distintas partes del sistema. Este último punto de vista, además nos proporcionará estados internos de nuestro sistema. UML aporta métodos de diagramación estándar para modelar (diagramas de estructura, que se corresponden con el punto de vista estático –diagrama de clases, diagrama de componentes y diagrama de paquetes; y diagramas de comportamiento, que se corresponden con el punto de vista dinámico: diagrama de casos de usos, diagrama de actividad y diagrama de secuencia)

UML 2 es extensible. Los mecanismos para personalizar y extender el UML son los perfiles y los estereotipos. UML-WAE utiliza los dos.

UML-WAE

Hoy en día, la mayoría de las aplicaciones se desarrollan entorno a la web, es decir, son aquellas aplicaciones que en su arquitectura y elementos principales están el navegador y el protocolo HTTP. Fue Jim Conallen, en 1988, quien definió una extensión del estándar UML para aplicaciones webs. Esta extensión pasó a llamarse WAE (Web Application Extension) y se trata de la convención o extensión más aceptada por lo que podríamos hablar de un estándar dentro de las distintas extensiones webs de UML.

En un principio las aplicaciones webs se basaban en un conjunto bien definidos de tecnologías. Con un servidor web, un cliente web capaz de comunicarse, vía protocolo HTTP, y de renderizar HTML. Con el tiempo se fueron sumando otras tecnologías como CSS, JavaScripts, XML, Ajax, Comet, WebGL, etc... WAE lo que nos ofrece, a los desarrolladores, es la posibilidad de modelar elementos como HTML, CSS, JavaScript, páginas de servidor, etc ... WAE ofrece a los desarrolladores la posibilidad de modelar estos elementos mediante estereotipo, valores etiquetados y restricciones.

Metodología y etapas del desarrollo.

Metodología de desarrollo ágil.

Las etapas de desarrollo de la metodología Ágil (planificación, análisis de requerimientos, diseño, codificación, revisión y documentación) se adaptan muy bien al desarrollo web. Por esto, y por las características que presenta este paradigma, el proyecto seguirá este modelo de desarrollo.

Etapas del desarrollo.

Viendo la dependencia entre librerías a implementar, lo más apropiado, es seguir un diseño ascendente (bottom-up). Siempre que el estadio anterior haya sido verificado y comprobado su completud, se podrá afrontar con éxito la siguiente etapa. Dicho de otra forma, cada etapa es dependiente de la etapa inmediatamente anterior. Siempre siguiendo la metodología ágil se ha decidido afrontar el proyecto en dos fases:

- Primera fase: se encargará de llevar a buen término la implementación de las librerías javascripts necesarias para la aplicación. En cada ciclo tendremos que realizar una planificación, análisis de riesgos, implementación, pruebas unitarias y documentación de cada librería. La primera fase constará pues, de seis ciclos bien definidos. El orden de los ciclos es el que sigue:
 - 1 Librería base con soporte para herencia. Esta librería debe tener toda la funcionalidad básica (bindings, curryings, etc) y debe estar muy optimizada ya que el perfecto funcionamiento de la aplicación dependerá en buena medida de ella.
 - 2 Librería para manejo de grafos. Constará principalmente, de un conjunto de clases con las cuales podremos modelar un grafo. Aunque un mapa mental, en principio, tiene una forma arbórea, árbol-n, no es descartable la posibilidad de realizar ciclos y generar nuevas aristas o enlaces a otros conceptos o ideas.
 - 3 Librería para el manejo de ficheros. Será la encargada de manejar ficheros, a partir de ella, realizaremos las clases de exportación e importación de mapas

mentales de la aplicación.

- 4 Librería gráfica. Ciñéndonos al contexto 2D, necesitamos un wrapper sobre la librerías propias del canvas. Esta librería, nos debe permitir pintar, cada uno de los elementos de nuestro árbol. Además de configurar, atributos visuales tales como color del trazo, relleno, etc. No se descarta el uso de alguna librería estándar. Para ello, se realizará una pruebas de concepto sobre ellas.
 - 5 Librería para el manejo de eventos del canvas. El canvas debe reaccionar tanto al teclado, ratón y touch. El canvas viene desprovisto de eventos sobre los elementos pintados en él y es aquí donde entra en juego esta librería.
 - 6 Por último, las librerías propias del mapa y prototipo o primera versión.
- Segunda fase: una vez implementadas todas las librerías necesarias y una primera versión (inoperativa), nos encontramos en disposición de ir elaborando la aplicación. En esta fase, desde mi punto de vista, el modelo iterativo incremental se adapta a la perfección al proceso de refinamiento de la aplicación. Cada iteración con llevará la implementación de la funcionalidad, en la que nos enfoquemos, y su posterior evaluación y prueba.

Bibliografía.

- [1] Software for mindmapping and information organisation <http://www.mind-mapping.org/>
- [2] Mind Map. Wikipedia. http://en.wikipedia.org/wiki/Mind_map
- [3] Tony Buzan. Wikipedia. http://es.wikipedia.org/wiki/Tony_Buzan
- [4] Sitio oficial de FreeMap. http://freemind.sourceforge.net/wiki/index.php/Main_Page
- [5] UML Resource page de OMG <http://www.uml.org/>
- [6] Building Web Applications With UML. Second Edition. Jim Conallen. Addison Wesley, 2002. ISBN 0-201-73038-3
- [7] Modeling Web Application Design with UML. <http://www.itmweb.com/essay546.htm>. Autor Jim Conallen.
- [8] Web Application Extensions. Documentación de IBM. http://publib.boulder.ibm.com/infocenter/rsysarch/v11/index.jsp?topic=/com.ibm.sa.oomethod.doc/topics/c_Web_app_Extensions_WAE.html
- [9] Manifiesto ágil. <http://www.agilemanifesto.org/iso/es/manifesto.html>
- [10] Principios del manifiesto ágil. <http://www.agilemanifesto.org/iso/es/principles.html>
- [11] Agile Alliance. <http://www.agilealliance.com/>