

CHROME WEBFILTERING TOOLBAR

MIGRACIÓN DE UN ADDON DE FIREFOX A UNA EXTENSIÓN DE GOOGLE CHROME

Autor: José Vicente Beneyto Ruiz
Director: Josep Francesc Silva Galiana

Curso 2010/2011

Índice de contenidos

1. INTRODUCCIÓN	3
1.1. OBJETIVOS Y MOTIVACIÓN	3
1.2. METODOLOGÍA	3
1.3. ORGANIZACIÓN DE LA MEMORIA	4
2. ESPECIFICACIÓN DE REQUISITOS	5
2.1. INTRODUCCIÓN	5
2.1.1. Propósito	5
2.1.2. Ámbito	5
2.1.3. Definiciones, acrónimos y abreviaturas	5
2.1.4. Referencias	5
2.2. DESCRIPCIÓN GENERAL	6
2.2.1. Perspectiva del producto	6
2.2.2. Funciones del producto	6
2.2.3. Características del usuario	6
2.2.4. Dependencias y limitaciones	7
2.3. REQUERIMIENTOS ESPECÍFICOS	7
2.3.1. Interfaces Externas	7
2.3.2. Requisitos Funcionales	7
2.3.2.1. Funcionalidad ‘Filter’	7
2.3.2.2. Funcionalidad ‘Reload’	8
2.4. APÉNDICES	8
2.3.1. Restricciones de las Extensiones	8
3. ANÁLISIS Y DISEÑO	13
3.1. CHROME EXTENSIONS	13
3.2. ESTRUCTURA DE UNA EXTENSIÓN	13
3.2.1. Archivo de Manifiesto	14
3.2.2. Archivos HTML	15
3.2.3. Archivos JavaScript	16
3.2.4. Archivos de Recursos	16
3.3. CONTENT SCRIPTS	16
3.4. COMUNICACIÓN ENTRE PÁGINAS	17
3.5. ALMACENAJE DE OPCIONES - LOCAL STORAGE	18
4. IMPLEMENTACIÓN	21
5. GUIA DE USO DE LA EXTENSIÓN	64
6. CONCLUSIONES	67
6.1. CONCLUSIONES DEL PROYECTO	67
6.2. TRABAJO FUTURO Y AMPLIACIONES	68
7. BIBLIOGRAFIA	68

1. INTRODUCCIÓN

El presente Proyecto de Fin de Carrera culmina y completa la formación que el alumno ha estado recibiendo durante estos últimos años en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Politécnica de Valencia.

Con la realización del presente Proyecto de Fin de Carrera, el autor pretende poner en práctica los conocimientos y técnicas adquiridos durante estos años de estudio, aplicados a un problema concreto. Para ello será decisiva la capacidad de análisis y de síntesis, así como el ser capaz de resolver problemas prácticos de una forma rápida, eficaz y profesional. No menos importante es la labor de documentación e investigación para profundizar en determinados aspectos clave para el desarrollo eficaz del Proyecto de Fin de Carrera. Cabe destacar también la figura del Tutor de Proyecto, como guía indispensable en la elaboración del mismo.

Mediante el presente Proyecto de Fin de Carrera se pretende estudiar los pasos a seguir para conseguir migrar una aplicación diseñada y pensada para un entorno de ejecución determinado a otros para los cuales inicialmente no se planteó.

1.1. OBJETIVOS Y MOTIVACIÓN

Dado que hoy en día, tener instalado un navegador web, medianamente actualizado y moderno, se ha convertido en algo indispensable para cualquier persona que haga un uso medio de la red, necesitamos que el uso e interacción con éste nos sea lo más cómodo y útil posible, para ello el interfaz de usuario debe ser fácil de manejar e intuitivo y que a ser posible tenga el mayor número de opciones y herramientas posibles para agilizar nuestro trabajo. Es por esto que complementar a un navegador con alternativas de instalación personalizables, ya se llamen módulos, extensiones, complementos, plugins o addons, se convierte en algo imprescindible.

En nuestro caso, partimos de un complemento (addon) para el navegador Mozilla Firefox[1], que se hospeda en el interfaz de este a modo de barra de herramientas y nos permite eliminar de una página web toda la información que no está relacionada con un criterio determinado de filtrado. A su vez, permite aumentar o disminuir la cantidad de información que se muestra al usuario por medio de un control de la tolerancia.

Tras valorar varias opciones, se tomó la decisión de intentar migrar dicho complemento a otro navegador como es el caso de Google Chrome[2], un navegador relativamente nuevo y distinto, pero que parte de un API para crear componentes similar al de Firefox a priori.

Esta memoria tratará pues de resumir los aspectos fundamentales de la migración de un plugin escrito para Mozilla Firefox a Google Chrome, demostrando la efectiva portabilidad del mismo.

1.2. METODOLOGÍA

Durante la ejecución de este Proyecto Final de Carrera se han empleado diversos métodos para llegar a resolver el problema de migrar un complemento de un navegador a otro.

En primer lugar se planificó el uso de una herramienta que posibilitara y agilizara el proceso de desarrollo. Para ello se usó Subversion[3] como herramienta de control de versiones, y aunque a nivel de producción entre los distintos alumnos que concurrentemente estamos en el mismo proyecto global, particularmente para mi propio desarrollo he usado Git[4].

No hace falta explicar en detalle en qué consiste este sistema de control de versiones, pero sí cabe destacar que al ser distribuido posibilita trabajar y hacer las modificaciones (*commits*) en una rama local para luego finalmente poder empujarlos (*push*) a la rama origen en el servidor.

Por otro lado, hubo que lidiar con las diferencias entre ambos sistemas de gestión de complementos, el de Mozilla Firefox y el de Google Chrome. Buscar la mejor manera para adaptar la estructura de archivos de la que partíamos en nuestro complemento. Ver la forma en la que el navegador trabaja con los objetos que se han instanciado y la manera de recuperar propiedades de los mismos. Y por último, encontrar la manera de substituir el interfaz de usuario actual y adaptarlo al nuevo encontrado en Chrome, lo que supone estudiar dónde alojar el menú de opciones, o de entrada de datos por parte del usuario. Todo esto se abordó paso a paso, implementando de manera horizontal cada faceta o característica requerida, haciendo incursiones más profundas de implementación de código cuando el estudio realizado daba garantías para abordar el problema.

La primera prueba de implementación fue determinante. Se tenía que poder montar una extensión mínima para tomarla como punto de partida. Esta extensión de partida debía ser capaz de poder alterar el DOM[5] de la página visitada, pues sólo así se podría llevar a cabo el objeto de este proyecto. Una vez comprobado este requisito, ya se podría ir agregando código y funcionalidades a la extensión, adaptar el interfaz de botones que interactúa con el usuario, etc.

1.3. ORGANIZACIÓN DE LA MEMORIA

La presente memoria está estructurada en las siguientes bloques o partes fundamentales:

- ♣ *Introducción.*
En esta primera parte se introduce al lector las causas que motivaron la realización de este proyecto final de carrera, así como los principales objetivos que ello conlleva y las metodologías empleadas para su realización.
- ♣ *Requisitos para la migración y portabilidad.*
Como en todo proyecto de ingeniería, se debe especificar los requisitos siguiendo en todo momento el estándar IEEE830-1984[14]. Durante este bloque se irán detallando cada uno de los requerimientos funcionales que afectarán al desarrollo del producto.
- ♣ *Análisis y diseño.*
A lo largo de este bloque se detallará todo lo referente al análisis y diseño del producto partiendo de las premisas que se dieron en la especificación de requisitos.
- ♣ *Implementación y guía de uso.*
En este otro bloque se tratará de exponer e ilustrar toda la implementación que se ha llevado a cabo en el proyecto, mostrando partes sustanciales de código fuente y explicando el funcionamiento de las mismas. También se incluye una breve guía de uso.
- ♣ *Conclusiones y evaluación del resultado obtenido.*
Por último se analizará el producto resultante y se plantearán una serie de conclusiones y posibles trabajos futuros.

Además, esta memoria se acompaña de capturas de pantalla con el objetivo de poder ilustrar y aclarar ciertos aspectos en el momento de tratarlos. Así como también se provee la bibliografía necesaria para referenciar aquellos documentos que han servido de apoyo y guía para elaborar la misma.

2. ESPECIFICACIÓN DE REQUISITOS

2.1. INTRODUCCIÓN

En este apartado se va a realizar una especificación de requerimientos siguiendo el estándar IEEE 830 destinado a la especificación de requerimientos software para proyectos de ingeniería. En nuestro caso se enfocará a la migración de un addon de Firefox a una extensión de Google Chrome.

2.1.1. Propósito

El propósito de este documento es realizar la especificación completa de todos los requisitos necesarios para satisfacer la migración de un addon, escrito originalmente para el navegador Firefox, a su nuevo entorno dentro del navegador Chrome. Para ello se realizará una exposición clara y concisa de requerimientos software con los que contar para su migración y se abordarán sus posibles limitaciones y funcionalidades.

2.1.2. Ámbito

La aplicación resultante se orienta a ser utilizada en un ámbito personal y doméstico. Pudiendo ser también enmarcada dentro de un contexto de aprendizaje para el posterior desarrollo de más extensiones para Chrome.

La aplicación será de uso individual y el usuario deberá obtener el producto a instalar mediante descarga o generación desde el código fuente.

2.1.3. Definiciones, acrónimos y abreviaturas

En este punto se va a detallar todos los términos y abreviaturas utilizadas en el desarrollo de la especificación de requisitos.

ERS: Especificación de Requisitos Software. Se puede definir como el proceso de estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos.

Requisito: se define como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Esta definición extiende y se aplica a las condiciones que debe cumplir o poseer un sistema para satisfacer un contrato, una norma o una especificación.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos. Es una asociación técnico-profesional mundial dedicada a la estandarización. Su trabajo es promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para el beneficio de los profesionales

2.1.4. Referencias

Para la elaboración del presente documento se ha seguido el estándar IEEE para la especificación de requisitos software.

IEEE Std 830-1984 - IEEE Guide to Software Requirements Specifications. IEEE Standards Board. 345 Eas 47th Street. New York, NY 10017, USA.1998.

2.2. DESCRIPCIÓN GENERAL

Las siguientes secciones describen los factores generales que afectan al producto y determinan sus requerimientos. A su vez, proveen el contexto necesario para esos requerimientos, los cuales serán definidos en detalle más adelante.

2.2.1. Perspectiva del producto

La aplicación (un complemento a migrar para el navegador Chrome) debe permitir al usuario poder filtrar la información obtenida por el navegador web de manera similar a como se comportaba en su entorno original en Firefox, y partiendo de las opciones seleccionadas por el usuario se comportará de una manera u otra.

Deberá funcionar en todas las versiones de Chrome, salvo en el caso de versiones tan recientes que cambien drásticamente el API interno y ello conlleve el tener que reimplementar gran parte de la aplicación o toda ella.

2.2.2. Funciones del producto

El producto final deberá satisfacer las siguientes funcionalidades:

Filtrado por palabra y Filtrado inverso por palabra: muestra únicamente aquellos nodos que contienen (o no contienen, en el caso del filtro inverso) la palabra clave.

Además se podrán habilitar o deshabilitar las siguientes opciones:

Filtrado manteniendo el árbol original de nodos (Keep Tree): cuando un nodo en el árbol representa un documento que debe incluirse en el resultado final del filtrado, es posible incluir también todos los nodos del camino desde ese nodo a la raíz, es decir, los antecesores del nodo en cuestión. Para conseguir esto, la opción keep tree debe estar seleccionada, por el contrario, si esta opción no está activada, únicamente se mostrará el nodo que forme parte del filtrado.

Filtrado manteniendo la estructura original: (Keep Structure): si el usuario escoge mantener la estructura de la página Web, cuando éste filtre la página, la información se mostrará en la posición que ocupa dentro de la página Web original. Si por el contrario, el usuario decide no mantener la estructura, cuando aplique un filtrado, se mostrará toda la información agrupada al inicio de la página, quedando una página más corta, y en muchos casos desapareciendo la barra de desplazamiento vertical.

Filtrado con nivel de tolerancia seleccionable: el usuario puede decidir aumentar o disminuir la cantidad de información que desea ver, una vez aplicado el filtrado, es decir, si solamente desea ver la frase que contiene la palabra, el párrafo, etc. También se puede definir el nivel de tolerancia antes de filtrar, de este modo ya se consigue ver la información de la página con el nivel deseado.

Filtrado con formateo de iframes (Format Size Iframes): si el usuario decide mantener el formato de los iframes que aparecen en la página Web, cuando éste aplique un filtro, si la palabra que busca se encuentra dentro de un iframe, el tamaño del mismo se mantendrá tal cual aparece en la página original, manteniendo, si el iframe la posee y si es necesario, la barra de desplazamiento vertical. Si por el contrario el usuario decide no mantener el tamaño de los iframes, cuando la palabra que busca se encuentre dentro del mismo, desaparecerá la barra de desplazamiento vertical, visualizando toda la información de un solo vistazo.

2.2.3. Características del usuario

Esta aplicación va dirigida a cualquier tipo de usuario que utilice el navegador Chrome.

El usuario tan solo deberá tener conocimientos básicos acerca de que es una extensión, para qué sirven, cómo se consiguen y cómo se instalan.

El uso de la aplicación es sencillo e intuitivo, por lo que un usuario con conocimientos básicos sobre el uso de navegadores e Internet puede utilizarla sin ningún tipo de problema.

2.2.4. Dependencias y limitaciones

Es imprescindible que cumpla todos los requerimientos especificados por los desarrolladores de Chrome para la implementación de extensiones. Entre ellas, y como se verá más adelante, destaca el uso de llamadas asíncronas para el paso de mensajes entre partes de la aplicación.

El usuario debe tener instalado la última versión estable del navegador Chrome y tener acceso a internet.

2.3. REQUERIMIENTOS ESPECÍFICOS

Esta sección contiene todos los requisitos software en un nivel de detalle suficiente como para permitir a los desarrolladores satisfacer esos requerimientos, y también a los testadores realizar las pruebas pertinentes. Cada requerimiento describirá un mínimo de cada interfaz de entrada/salida al sistema software y todas las funciones implementadas por el sistema en respuesta a eventos de entrada/salida.

2.3.1. Interfaces Externas

La interacción con el usuario se llevará a cabo a través de los periféricos del computador, esto es: ratón y teclado o elementos de accesibilidad alternativa para personas de movilidad reducida, luego podemos concluir que no hay requisitos que afecten a la interfaz de usuario.

2.3.2. Requisitos Funcionales

2.3.2.1. Funcionalidad 'Filter'

Descripción

Esta función se encarga de filtrar los componentes de la página web que coincidan con el criterio de búsqueda (introducido en el cuadro de texto de entrada), ocultando o borrando los demás componentes no coincidentes. Los componentes de una página reciben el nombre de nodo. Una página web se puede ver como un árbol de nodos de etiquetados. Este árbol se representa internamente con el estándar DOM [5].

Requisitos funcionales

Se requiere de un criterio de filtrado que introducirá el usuario por medio del cuadro de texto que incluye el interfaz de la aplicación.

Excepciones que genera:

1. No se ha introducido una palabra en el cuadro de texto.

En este caso, la aplicación muestra una ventana de alerta indicando al usuario que el criterio de búsqueda es nulo.

No se realiza ninguna acción sobre el DOM [5] de la página.

2. La palabra introducida como criterio de búsqueda no se encuentra en la página.

Este caso también se mostrará una ventana de alerta para indicar al usuario que el criterio de búsqueda no se encuentra en la página.

De igual modo, no se realiza ninguna acción sobre el DOM [5] de la página.

Por otro lado, la aplicación requiere de unas preferencias o valores que deben ser definidos por el usuario para poder comenzar la tarea de filtrado. Estas preferencias deben tomar un valor por defecto o inicial para evitar valores vacíos en caso de que el usuario no las defina. Las preferencias se muestran en la pantalla de

'Opciones', e inicialmente mostrarán el valor por defecto o inicial, siendo posible modificarlas de forma temporal o permanente por el usuario.

Estas preferencias son:

Keep Structure: los nodos se mantendrán en la misma posición que en el árbol original y se mostrarán huecos en blanco ocultando aquella información no relevante para el criterio introducido.

Format Size iFrames: los nodos iFrames se modificarán para ocupar únicamente el tamaño del fragmento de página que abarcan, en caso contrario se mantendrá el tamaño original . Hay que recordar que normalmente la página web de un iFrame es más grande que el área reservada para tal iFrame.

Keep Tree: se incluirá en el árbol de nodos resultante los antecesores de los nodos relevantes, desde el nodo raíz o 'root' hasta el nodo en cuestión.

Por último hay otra preferencia no incluida en la pantalla de 'Opciones', y que se incluye en el interfaz principal:

Tolerance: el valor por defecto es 0 y a medida que aumenta su valor hace que los nodos que estén directamente relacionados con nodos relevantes pasen a ser relevantes también.

Una vez apliquemos el filtrado, obtendremos como resultado final, una nueva página que contendrá un árbol de nodos distinto al original y que se ajustará a nuestras preferencias.

2.3.2.2. Funcionalidad 'Reload'

Descripción

Esta función se encarga de recargar la página original. Para ello restablecerá el árbol original previamente guardado.

Requisitos funcionales

El usuario dispondrá de un botón para este fin y no requiere de preferencias ni requerimientos extras.

2.4. APÉNDICES

2.3.1. Restricciones de las Extensiones

Las restricciones para la migración del add-on original escrito para Firefox a extensión de Google Chrome vienen determinadas por las diferencias que existen entre ambas, tanto de entorno de usuario, como de características de la programación de ambas, etc.

Lo primero y más esencial es la similitud (a priori) del API que proveen ambos navegadores para el desarrollo de complementos. Tanto Firefox como Chrome aceptan un desarrollo basado en código JavaScript[6] aunque con diferencias de diseño que más adelante se abordarán. Quizás la más notable diferencia se produce en la interfaz de presentación al usuario, teniendo aquí que recurrir a definiciones de interfaces XUL para Firefox, ya que este hace uso más extenso del interfaz del navegador pudiendo reservar una barra de herramientas específicamente para nuestro complemento. Por contra, en Chrome tenemos un ámbito más cerrado para las extensiones, siendo éstas representadas en el interfaz del navegador por un mero icono que se sitúa en la única barra de herramientas que posee el navegador. Ahora bien, este icono será el encargado de lanzar las acciones que para él definamos.

Por otro lado, algunas de las funcionalidad originales del complemento, como son el resaltado por palabras clave, han sido descartadas ya que además de no aportar nada extra que el navegador no tuviese ya, éste lo hacía mejor, más completo y rápido. También veremos, como ya se ha comentado, que algunos otros elementos han tenido que ser adaptados debido a las restricciones que impone el nuevo interfaz.

A continuación se describe las diferencias más destacables entre ambos navegadores y la relación que tienen con el desarrollo de componentes adicionales, ya sea en forma de addon o extensión, y sus requerimientos y/o restricciones.

Facilidad de instalación

Lo que más gusta acerca de las extensiones de Chrome en general es la capacidad de instalar y desinstalar extensiones sin necesidad de reiniciar el navegador. Por otro lado, Firefox tiene que reiniciarse cada vez que se instala una extensión, o incluso un tema.

Chrome además posee un interfaz de fácil manejo a modo de página web, donde podemos, obteniendo una vista para desarrolladores, gestionar todos los aspectos relacionados con la instalación y desinstalación, así como de las opciones que para la extensión hayan sido programadas. Todo esto hace de Chrome un navegador atractivo para los creadores de extensiones, ya que esto lo hace muy ágil a la hora de desarrollar, testear y mantener extensiones.

Un ejemplo de instalación podría ser el siguiente:



Imagen 1: Cuadro de diálogo de instalación

Rendimiento del navegador con las extensiones / complementos

Desde la introducción de extensiones de Google Chrome, muchos usuarios se mostraron escépticos ya que pensaban que al instalar muchas extensiones frenaría la velocidad con la que se mueve el navegador, una de sus principales cualidades. Pero se puede demostrar que nada de esto ocurre. Chrome ejecuta cada extensión, pestaña, etc. en procesos independientes, por lo que una extensión no puede afectar al rendimiento global del navegador. Esta es una tecnología ha hecho de Chrome uno de los más rápidos navegadores de los hoy en día disponibles y mucho más manejable que sus rivales.

Por otra parte, Firefox es conocido por ralentizar el propio equipo en el que se ejecuta cuando se alcanza un número determinado de complementos instalados. Seguramente no sea por culpa del propio navegador, pero si se instalan de 10 a 15 o más complementos y a continuación se intenta abrir a la vez unas 10 pestañas y notará el consumo de memoria RAM que experimenta su equipo, además de la perdida de rendimiento que ello ocasiona en Firefox y en su equipo.

Elección de las extensiones y características

Firefox tiene una inmensa cantidad de desarrolladores y el número de addons disponibles es enorme, además de que hasta la fecha son en general más útiles y complejas que las que se pueden encontrar por el momento para Chrome. Esto es lógico ya que Firefox es un navegador maduro comparado frente a Chrome. Sin embargo, ya van apareciendo algunas extensiones muy avanzadas creadas para Chrome, incluyendo la utilidad 'Speed Tracer' para realizar trazas de código creada para facilitar la vida a los desarrolladores. Así que

podemos asumir y sólo cabe esperar que con el tiempo y en un futuro cercano, Chrome pueda estar a la altura de la gran base de complementos que posee Firefox.

A continuación se muestra una captura de 'chrome web store', la página web de Chrome desde donde podemos acceder a la mayoría de recursos disponibles para el navegador, incluyendo extensiones, apps, themes, etc.

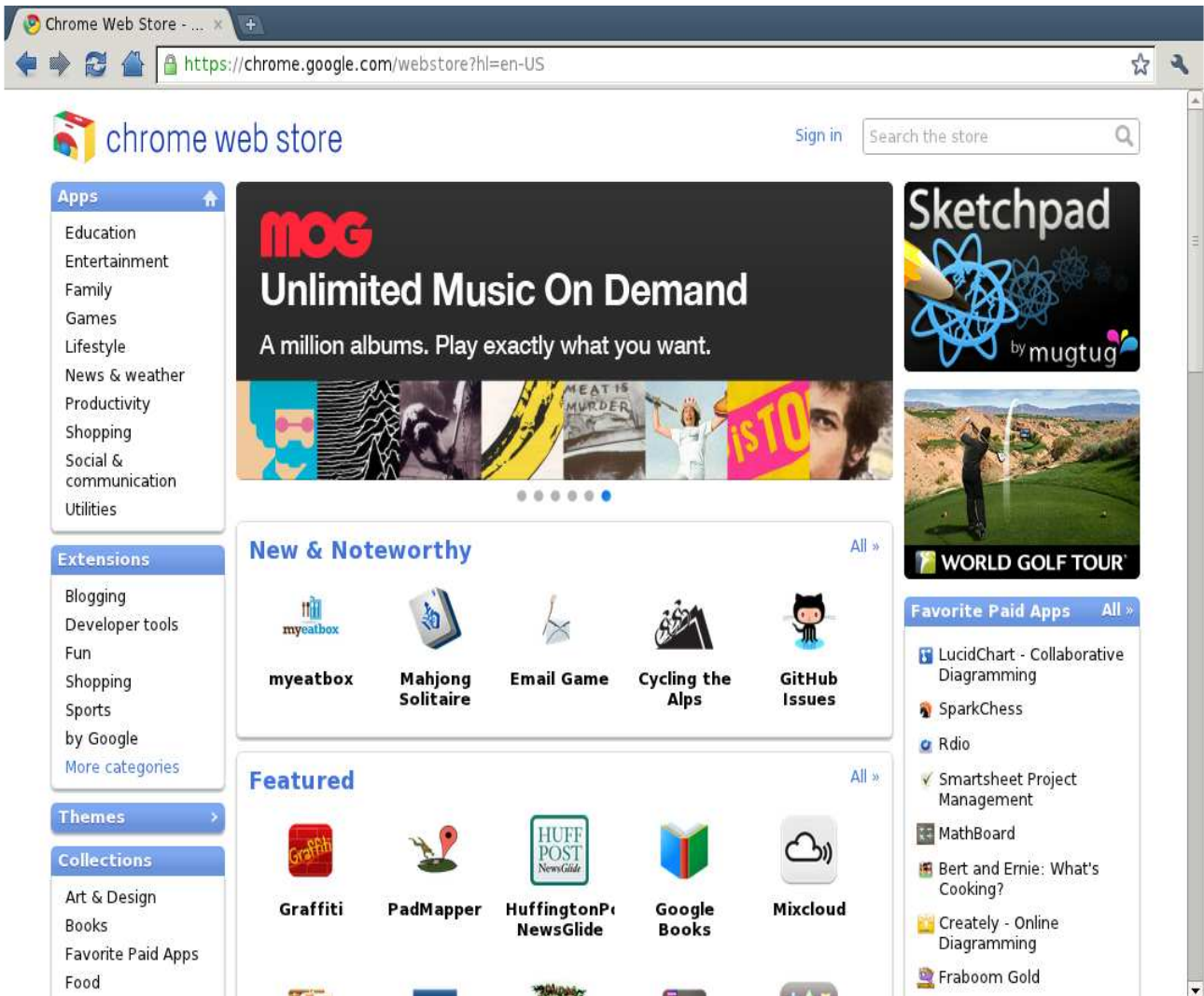


Imagen 2: Chrome web store

Facilidad de Gestión de Extensión

Tanto Firefox y Google Chrome tienen ambos una interfaz para la administración de extensiones. Firefox revisa cada complemento de los que tenemos instalados y nos avisa de que ha localizado una nueva actualización, invitándonos a llevar a cabo la acción. Posteriormente nos pide un reinicio del navegador para poder hacer uso de ella. Sin embargo, las actualizaciones de Chrome se gestionan de manera automática encuentra se encuentra una actualización disponible y no requiere de reinicios del navegador. Esto puede ser algo bueno, pero también choca y puede ser molesto para los usuarios que, por lo que sea, prefieran permanecer con la versión en uso y no desean actualizar a una versión más reciente de la extensión.

En la siguiente figura se muestra una captura de pantalla de la página que provee Chrome para administrar las extensiones que tengamos instaladas. En este caso aparecen unas pocas extensiones entre las que se encuentra nuestra extensión Web Filtering, acompañada de las acciones que nos permite Chrome y las muestra a modo de enlaces, estas son: Reload, Disable, Uninstall y Options. Por otro lado podemos ver más acciones destinadas al 'modo desarrollador'. Y por último una sección que enlaza con el Chrome Web Store y nos permite encontrar más extensiones y complementos para instalar.

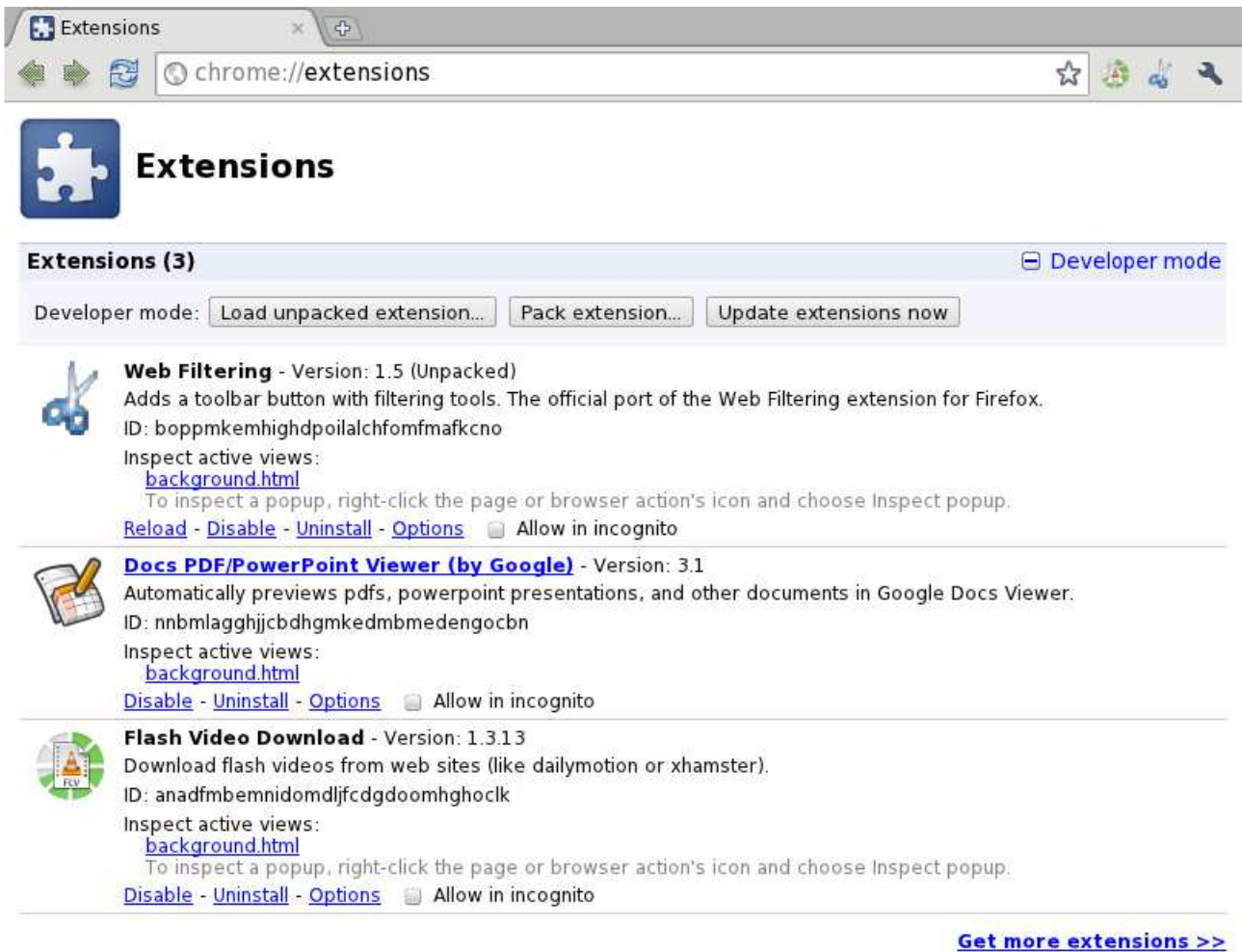


Imagen 3: Página de administración de extensiones

Seguridad

Como antes se mencionó, Google Chrome ejecuta sus extensiones en un proceso independiente, lo que lleva a un entorno de ejecución separado. Chrome adopta una arquitectura multi-proceso como parte de la estrategia de seguridad de Google "principio de mínimo privilegio", por lo que un problema con una extensión no debe afectar a las demás ni al propio navegador. En esencia, el entorno en el que se moverá la extensión es limitado ya que aísla a los procesos del acceso a cualquier cosa dentro del sistema, incluidos los archivos, otros procesos, y objetos internos.

Otro punto importante es la separación de entornos de ejecuciones distintos dentro de la propia extensión, diferenciando partes que trabajan con el DOM de la página, partes que interactúan con el usuario, etc. haciendo más seguro al navegador ante ataques de cross-site scripting, y código pernicioso en general. Sin embargo, Firefox no tiene esta característica. Como anécdota, hace poco, Firefox tubo que deshabilitar algunos addons de Microsoft debido a los problemas de seguridad que causaban en el navegador.

El siguiente diagrama ilustra la arquitectura multi-proceso de Chrome que se explicará más adelante.

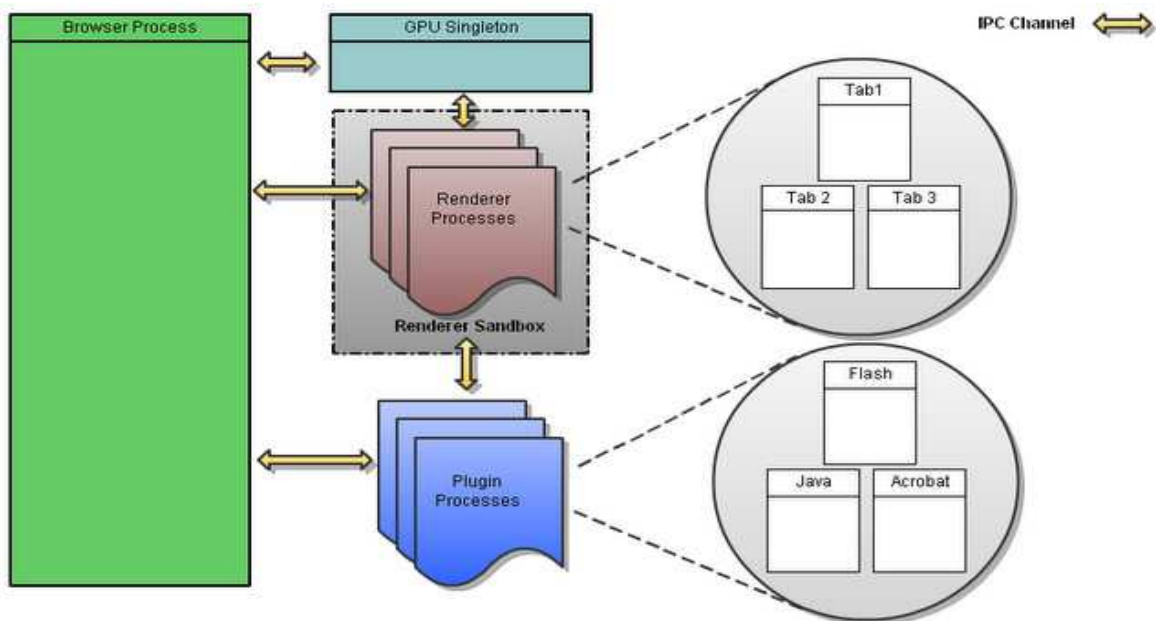


Imagen 4: Diagrama multi-proceso de Chrome

En el diagrama anterior sólo se han representado los procesos que tienen lugar en la sesión de navegación, por lo tanto otros potenciales procesos que pudieran existir han sido descartados. Por ejemplo, un proceso de instalación puede ser generado cuando las extensiones de Chrome se descargan. Estos procesos auxiliares se han omitido en el diagrama por simplicidad.

Cada uno de los principales componentes se describen brevemente a continuación.

Browser Process o *Proceso Navegador* - Un único proceso navegador, con privilegios sobre el sistema, existe durante la duración de toda la sesión de navegación. El proceso del explorador es responsable de generar otros procesos cuando sea necesario, las tareas de aprovisionamiento a ellos, y la realización de operaciones con privilegios, en nombre de los procesos que no tienen este acceso al sistema, necesario para realizar las propias operaciones.

Renderer Process o *Proceso de Render* - Un proceso de render se genera cada vez que una nueva pestaña o ventana se abre, y es responsable de realizar el análisis y la representación de la página web que se visualiza. Esto incluye el análisis de HTML, la funcionalidad de JavaScript, y procesamiento de imágenes, etc.

Plugin Process o *Proceso de Plugins* - Los plugins son alojados dentro de su propio proceso privado y se comunican con los procesos de render y del navegador. Este es un proceso único para cada uno de los diferentes plugins que tengamos instalados, independientemente del número de diferentes instancias de objetos incrustados de ese plugin. Los procesos de plugins realmente se ejecutan con privilegios por defecto (como lo hace el proceso del navegador), aunque podemos esperar que Chrome cambiará esto cuando determinadas cuestiones de compatibilidad se resuelven.

GPU Process o *Proceso de GPU* - Este proceso existe para realizar tareas relacionadas con la GPU en nombre de los procesos de render.

IPC Channels o *Canales IPC* - Los canales de la IPC constituyen el marco de mensajería de Chrome, que ofrece servicios para los diversos procesos de Chrome para interactuar entre sí sobre los mecanismos locales de transporte específicas del sistema operativo.

Categorización Extensión/Add-on

Actualmente el número de complementos que posee Firefox no es comparable al de Chrome, en mayor parte debido a que éste último está todavía en su infancia y no hay una clasificación tan clara y bien categorizada como la que Mozilla posee para su navegador, cosa que lo hace fácil para encontrar complementos para necesidades específicas. (Ver Imagen 2)

3. ANÁLISIS Y DISEÑO

En esta sección se explican detalladamente las decisiones tomadas en las fases de análisis y diseño. En primer lugar se introducen los lenguajes y tecnologías utilizados tanto en ambas fases como en el desarrollo final.

3.1. CHROME EXTENSIONS

¿Qué son las extensiones?

Las extensiones son pequeños programas de software que se proveen a modo de complemento y puede modificar y mejorar la funcionalidad del navegador Chrome. Son por tanto una manera de personalizarlo a medida que requerimos cierta utilidad, etc.

Las extensiones son programadas, por lo tanto tienen código fuente, pero este es una abstracción a un nivel muy distinto de lo que sería el núcleo interno del navegador, y para ello se apoyan de tecnologías web como HTML, JavaScript y CSS.

Las extensiones tienen todos sus archivos fuente agrupados en un sólo archivo que el usuario descarga e instala, este archivo cuya extensión es .crx es un paquete comprimido que contiene todos esos archivos HTML, CSS, JavaScript, imágenes, y todo lo que necesita. En su esencia son meras páginas web que pueden utilizar el API proporcionado por Chrome para interactuar con otras páginas web o servidores. Las extensiones también pueden interactuar mediante programación con las características del navegador, como los marcadores, pestañas, etc.

3.2. ESTRUCTURA DE UNA EXTENSIÓN

Cada extensión está estructurada según los siguientes archivos:

- ^ Un archivo de manifiesto
- ^ Uno o más archivos HTML
- ^ Uno o más archivos de JavaScript. (opcional)
- ^ Archivos de recursos necesarios para la extensión, como por ejemplo archivos de imagen, etc. (opcional)

Una extensión está formada por al menos un archivo de manifiesto y alguna página más, y puede ser tan grande y compleja como queramos, ahora bien, debe albergar todos estos archivos bajo una carpeta única. Cuando distribuya su extensión, el contenido de esta carpeta se empaquetará en un archivo especial comprimido ZIP que tiene un sufijo .crx.

Lo bueno es que Chrome tiene una opción para facilitar este trabajo a los desarrolladores de extensiones, ya que provee un interfaz para cargar extensiones en base a carpetas que en ellas tengan un manifest.json, y en definitiva contengan una extensión, sin la necesidad de tener que empaquetar en un .crx para instalar, etc.



Extensions

Extensions (3) ☰ Developer mode

Developer mode:


 **Web Filtering** - Version: 1.5 (Unpacked)
Adds a toolbar button with filtering tools. The official port of the Web Filtering extension for Firefox.
ID: boppmkemhighdpoilalchfomfmafkcno
Inspect active views:
[background.html](#)
To inspect a popup, right-click the page or browser action's icon and choose Inspect popup.
[Reload](#) - [Disable](#) - [Uninstall](#) - [Options](#) Allow in incognito

Imagen 5: Ejemplo de desarrollo de extensión en base a carpetas

En referencia a los archivos, se puede poner cualquier tipo de archivo en una extensión, pero ¿cómo se usa? Por lo general, se puede hacer referencia al archivo mediante una dirección URL relativa, tal como lo haría en una página HTML. He aquí un ejemplo de referencia a un archivo denominado `myimage.png` que está en una subcarpeta denominada `images`.

```

```

Aunque también es posible acceder por dirección URL absoluta:

```
chrome-extension://<extensionID>/<pathToFile>
```

En esta última dirección, el `<extensionID>` es un identificador único que genera el sistema de extensiones de chrome para cada extensión. Además, se puede ver el ID de todas las extensiones cargadas tecleando directamente `chrome://extensions/` en la barra de url del navegador.

Por otro lado, el `<pathToFile>` es la ubicación del archivo en la carpeta de la extensión, que es la misma que la dirección URL relativa.

3.2.1 Archivo de Manifiesto

El archivo de manifiesto, llamado `manifest.json`, da información y define la extensión sobre cada uno de los archivos que ésta usará.

JSON^[7] (acrónimo de JavaScript Object Notation) es un formato ligero para el intercambio de datos.

Este formato es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador semántico de JSON.

En JavaScript, un texto JSON se puede analizar fácilmente usando el procedimiento `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML.

Un ejemplo típico de archivo manifiesto para una extensión podría ser el siguiente:

```

{
  "name":          "myextension",
  "version":       "1.0",
  "description":   "Example extension."
  "icons":        {"128": "icon_128.png"},
  "background_page": "background.html",
  "permissions":  ["http://*.*/*", "https://*.*/*"],
  "Browser_action": {
    "Default_title": "mytitle",
    "Default_icon":  "icon_64.png",
    "Default_popup": "popup.html"
  }
}

```

Como podemos ver se especifica el nombre, versión y una descripción detallada de la extensión, así como los iconos usados y las páginas html adicionales (background y popup). Por otro lado debemos indicar con la directiva 'permissions' las combinaciones de URLs válidas para interactuar con nuestra extensión, que en nuestro caso será cualquier URL http o https. Estos permisos ayudan a establecer un marco de seguridad en el cual se ejecutará nuestra extensión, protegiéndola así de páginas que pudieran vulnerar nuestro código.

De las directivas anteriores, solamente son obligatorias 'name' y 'version', el resto son opcionales pudiendo dar multitud de combinaciones y posibilidades en función de lo que necesitemos para crear nuestra extensión.

3.2.2. Archivos HTML

La mayoría de las extensiones tienen una página de fondo (o background), esto es, una página invisible que sostiene la lógica principal de la extensión. Pero la página de fondo o background no es la única página HTML que puede tener una extensión. Por ejemplo, si necesitamos cierta interacción con el usuario podríamos tener una ventana emergente (o popup), o incluso una página de opciones y ambas también se realizan mediante un archivo HTML.

La siguiente figura muestra la arquitectura de una acción emergente del navegador. El emergente de contenidos es una página web, que se define en un archivo HTML (popup.html). Además, en la ventana emergente no es necesario duplicar el código que está en la página de fondo (background.html) debido a que puede invocar funciones de la página de fondo.

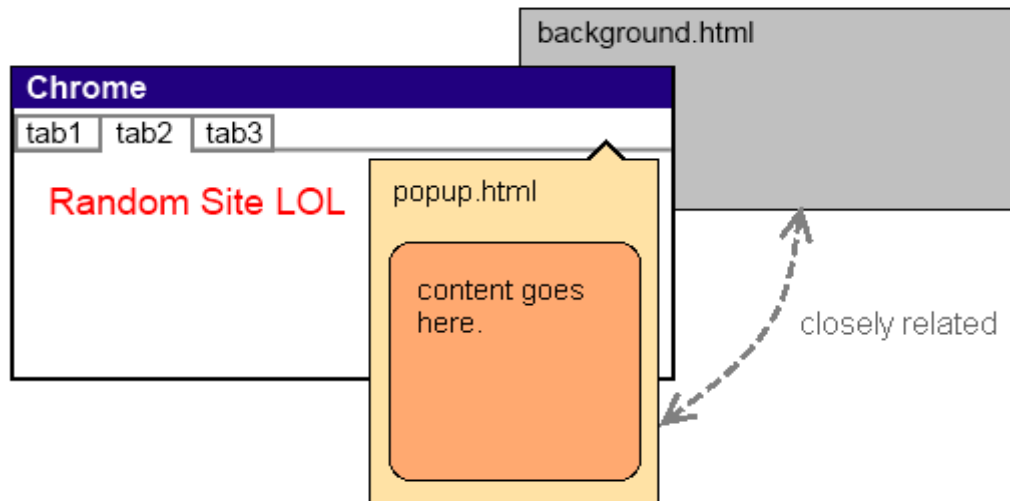


Imagen 6: Acción emergente popup

3.2.3. Archivos JavaScript

Una extensión, además de archivos HTML puede necesitar código para realizar ciertas tareas, y lo más seguro es que esto sea así. Para ello se emplea código JavaScript.

El intérprete de JavaScript será el encargado de interpretar este código que podrá ir de manera separada a los archivos HTML o incluyéndose en ellos de manera embebida.

Por claridad y como regla general, en este proyecto se ha optado por la modularidad en su código, por tanto y en medida de lo posible se separará el código en archivos individuales. Por ejemplo, para el popup tendremos un fichero popup.html y otro popup.js.

También se ha optado por albergar todos los archivos JavaScript bajo un directorio llamado 'js' y por incluir un fichero de librería para la abstracción de los objetos DOM, esta librería recibe el nombre de JQuery[8].

3.2.4. Archivos de Recursos

Los archivos de recursos pueden ser imágenes, texto plano, o cualquier otro tipo de archivo que no contenga HTML y JavaScript para ser interpretado. Este tipo de archivo puede acompañar a la extensión y normalmente son usados por las páginas HTML que requieren mejorar el aspecto gráfico (en el caso de imágenes), etc.

A su vez también pueden suponer recursos necesarios para la extensión y definidos en el archivo de manifiesto, como el icono que representa a la extensión y es mostrado en el navegador como parte del popup.

En el caso de imágenes para poder mantener la modularidad del proyecto se ha optado por tenerlos en una carpeta llamada 'images'.

3.3. CONTENT SCRIPTS

Si su extensión tiene que interactuar con las páginas web, entonces es necesario un archivo 'content script'. Este no es más que un script de JavaScript que se ejecuta en el contexto de una página que ha sido cargada en el navegador. Para hacernos una mejor idea, un 'content script' es una parte más de la página cargada y no parte de la extensión cargada. De esta forma, el 'content script' se sitúa en el mismo contexto que la página web visitada y por tanto podrá acceder a esta.

Los 'content scripts' pueden leer los detalles de las páginas web visitadas por el navegador, y también pueden hacer cambios sobre ellas. Por tanto pueden leer y modificar el DOM de la página y es esta cualidad la que lo hace adecuado para nuestro proyecto.

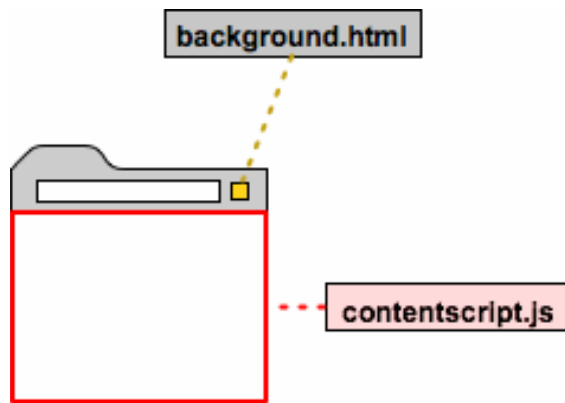


Imagen 7: Interaccion del ContentScript

Por otro lado, los 'content scripts' no están aislados de la extensión padre, por lo que pueden intercambiar mensajes con otras páginas de la extensión. Por ejemplo, un 'content script' puede enviar un mensaje cada vez que encuentra una fuente RSS en una página del navegador. O una página de fondo podría enviar un mensaje pidiendo un 'content script' para cambiar la apariencia de la página web.

En la figura siguiente se muestra mediante el sentido de las flechas las direcciones que pueden tomar los mensajes dentro del contexto de la extensión.

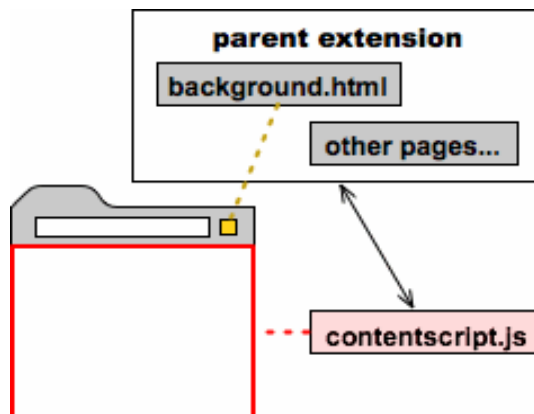


Imagen 8: Interacción del ContentScript con el resto de la extension

3.4. COMUNICACIÓN ENTRE PÁGINAS

Para intercambiar datos, las páginas HTML dentro de una extensión a menudo necesitan comunicarse. Debido a que todas las páginas de una extensión se deben de ejecutar en el mismo hilo del mismo proceso, las páginas pueden hacer llamadas directas a funciones entre sí.

Además, para encontrar las páginas dentro de una extensión, Chrome provee de ciertos métodos tales como `getViews()` y `getBackgroundPage()`. De esta forma, una vez que la página tiene una referencia a otras páginas

dentro de la extensión, la primera página puede invocar funciones en las otras páginas, y puede manipular su DOM.

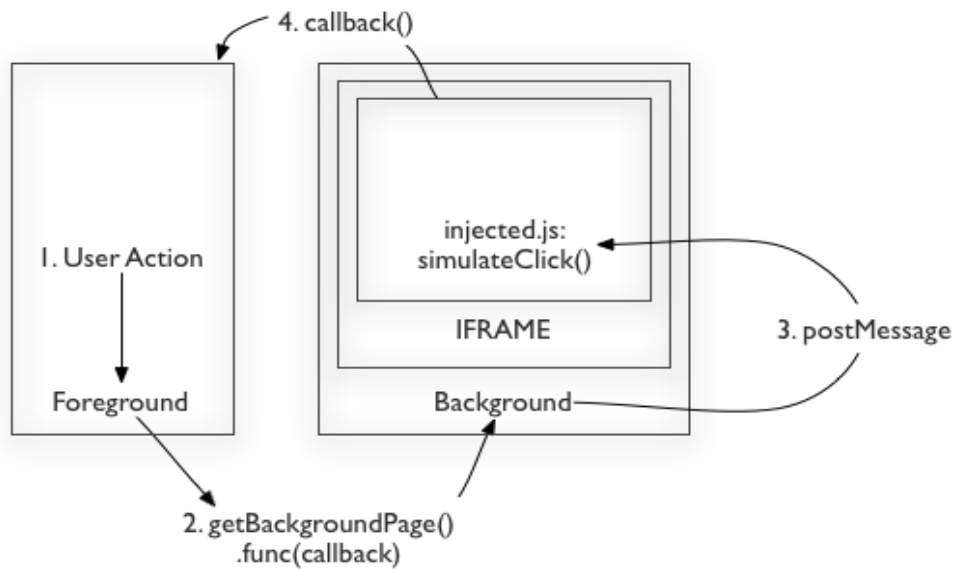


Imagen 9: Llamadas a funciones entre páginas

3.5. ALMACENAJE DE OPCIONES - LOCAL STORAGE

Un aspecto importante para toda extensión es el hecho de ser capaz de almacenar los datos del usuario y poder recuperarlos más tarde. Para esto, las extensiones pueden hacer uso del objeto LocalStorage, una interfaz de almacenamiento derivado del API web HTML5[12].

Veamos a continuación una ligera muestra de lo que nos puede proveer este objeto LocalStorage.

Guardando Opciones

Para guardar una cadena con un valor en el LocalStorage, podemos asignar el valor a la propiedad del objeto. Se podría usar el siguiente código:

```
localStorage["mysetting"] = "myvalue";
```

O de manera equivalente:

```
localStorage.mysetting = "myvalue";
```

Recuperando Opciones Guardadas

Para poder recuperar opciones previamente guardadas, directamente podemos recuperar un atributo del objeto LocalStorage:

```
myvar = localStorage["mysetting"];
```

O de manera equivalente:

```
myvar = localStorage.mysetting;
```

Una manera más eficaz de obtener los atributos guardados en el objeto LocalStorage pasaría por abstraer este proceso a una función que además validase el atributo a recuperar antes de retornarlo al usuario. Veamos un ejemplo de ello mediante la función 'loadFavColor', en donde si el color recuperado del LocalStorage no es válido, se devolverá el color por defecto:

```
var defaultColor = "blue";

function loadFavColor() {
  var favColor = localStorage["favColor"];
  // los colores validos son red, blue, green and yellow
  if (favColor == undefined || (favColor != "red" && favColor != "blue" && favColor != "green" && favColor != "yellow")) {
    favColor = defaultColor;
  }
  return favColor;
}
```

Borrando Opciones Guardadas

De igual forma que se pueden recuperar y guardar opciones, se pueden borrar éstas completamente del objeto localStorage.

Para ello podríamos hacer algo así:

```
localStorage.removeItem("mysetting");
```

La página de Opciones

Lo más cómodo a la hora de interactuar con opciones es tener una página HTML enteramente dedicada a ello. Para poder usar una página de opciones deberemos definirlo en el archivo manifiesto de la siguiente manera:

```
{
  "name": "My Extension",
  ...
  "options_page": "options.html"
  ...
}
```

Ejemplo de Página de Opciones

Siguiendo con los ejemplos anteriores de selección de color favorito, veamos como podría ser una página de opciones:

```
<html>
<head>
  <title>Options for Color Chooser</title>
  <script type="text/javascript" src="options.js"></script>
```

```

</head>
<body onload="loadOptions()">
  <h1>Favorite Color</h1>
  <select id="color">
    <option value="blue">Blue</option>
    <option value="red">Red</option>
    <option value="green">Green</option>
    <option value="yellow">Yellow</option>
  </select>
  <br />
  <button onclick="saveOptions()">Save</button>
  <br />
  <button onclick="eraseOptions()">Restore default</button>
</body>
</html>

```

Ahora para que la página quede completa deberemos agregarle el código necesario para que se implementen las funciones *loadOptions*, *saveOptions* y *eraseOptions*.

Este código JavaScript se podría insertar de forma embebida, pero se ha preferido separarlo en un fichero *options.js* (como bien se define en la cabecera del fichero HTML).

Veamos como queda el fichero con el código:

```

var defaultColor = "blue";

function loadOptions() {
  var favColor = localStorage["favColor"];
  // valid colors are red, blue, green and yellow
  if (favColor == undefined ||
      (favColor != "red" && favColor != "blue" && favColor != "green" && favColor != "yellow")) {
    favColor = defaultColor;
  }
  var select = document.getElementById("color");
  for (var i = 0; i < select.children.length; i++) {
    var child = select.children[i];
    if (child.value == favColor) {
      child.selected = "true";
      break;
    }
  }
}

function saveOptions() {
  var select = document.getElementById("color");
  var color = select.children[select.selectedIndex].value;
  localStorage["favColor"] = color;
}

function eraseOptions() {
  localStorage.removeItem("favColor");
  location.reload();
}

```

4. IMPLEMENTACIÓN

Lo primero para llevar a cabo la migración del plugin fue establecer una estructura clara, sencilla e intuitiva de directorios y ficheros, a la vez que debería ser escalable pensando en futuras ampliaciones.

Para ello se separó por directorios cada familia o tipo de ficheros. Por ejemplo, los recursos como iconos u hojas de estilo y los archivos JavaScript.

Así la estructura quedó de la siguiente forma:

```
|-- background.html
|-- css
| |-- options.css
| `-- popup.css
|-- icons
| |-- 128.png
| |-- 16.png
| |-- 32.png
| |-- 48.png
| |-- 64.png
| |-- filter.png
| |-- highlight.png
| |-- next.png
| |-- previous.png
| `-- reload.png
|-- js
| |-- background.js
| |-- jquery.js
| |-- options.js
| |-- popup.js
| |-- popup.slicetoolbar.js
| `-- slicetoolbar.js
|-- manifest.json
|-- options.html
`-- popup.html
```

A continuación pasaré a enumerar y describir brevemente cada una de las partes más importantes de la extensión.

Archivo de Manifiesto

Nuestro archivo de manifiesto, como no podría ser de otra forma, debe llamarse *manifest.json*.

Su aspecto es el siguiente:

```
{
  "name": "Web Filtering",
  "version": "1.5",
  "description": "Adds a toolbar button with filtering tools. The official port of the Web Filtering extension for Firefox.",
  "icons": {
    "16": "icons/16.png",
    "32": "icons/32.png",
    "48": "icons/48.png",
    "64": "icons/64.png",
    "128": "icons/128.png"
  }
}
```

```

},

"browser_action": {
  "default_icon" : "icons/32.png",
  "default_title" : "Web Filtering",
  "popup" : "popup.html"
},

"options_page": "options.html",
"background_page": "background.html",

"permissions": [ "tabs", "http://*/*", "https://*/*" ]
}

```

Página de Fondo o Background

Nuestra página de fondo o background constará de 2 partes, por un lado el fichero *background.html* y por el otro el código necesario en *background.js*.

Para la página HTML no habrá que extenderse mucho ya que este no será presentado en ninguna parte del interfaz del usuario y su existencia se limita a especificar donde estará el código JavaScript asociado. Eso sí, este archivo es esencial ya que es el contenedor principal de la extensión.

La página *background.html* queda así:

```

<!DOCTYPE html>
<html id="web-filtering-background">

<head>
  <meta charset="utf-8">
  <title>Web Developer Background</title>
  <script src="js/background.js"></script>
</head>

<body>

</body>

</html>

```

Y la página *background.js* que contiene el código JavaScript asociado:

```

var WebFiltering = {};

WebFiltering.Background = {};

WebFiltering.Background.debug = false;

WebFiltering.Background.setStorageItem = function(key, value)
{
  window.localStorage.removeItem(key);
  window.localStorage.setItem(key, value);
  if (WebFiltering.Background.debug == true)
  {
    console.log("set-storage-item(" + key + ", " + value + ")");
  }
};

WebFiltering.Background.getStorageItem = function(key)
{
  var result = window.localStorage.getItem(key);
  if (WebFiltering.Background.debug == true)

```

```

    {
        console.log("get-storage-item(" + key + ") = " + result);
    }
    return result;
};

WebFiltering.Background.clearStorageItems = function()
{
    window.localStorage.clear();
};

WebFiltering.Background.request = function(request, sender, sendResponse)
{
    if (request.type == "get-storage-item")
    {
        sendResponse(WebFiltering.Background.getStorageItem(request.item));
    }
    else if (request.type == "set-storage-item")
    {
        sendResponse(WebFiltering.Background.setStorageItem(request.item, request.value));
    }
    else
    {
        sendResponse({});
    }
};

chrome.extension.onRequest.addListener(WebFiltering.Background.request);

```

Página emergente o popup

La página emergente o popup constará también de varias partes: por un lado el fichero popup.html que será el visible y el que da forma al interface de usuario (cuadros de texto, botones, etc.) y por el otro lado separaremos el código JavaScript en un primer fichero popup.js que tendrá llamadas de biblioteca a otros ficheros de código secundarios donde estarán implementadas la mayoría de funcionalidades de nuestra extensión, sino todas.

La página *popup.html* queda así:

```

<!DOCTYPE html>
<html id="web-filtering-popup">

<head>
  <meta charset="utf-8">
  <title>Web Filtering Popup</title>
  <link href="css/popup.css" rel="stylesheet">
  <script src="js/jquery.js"></script>
  <script src="js/popup.js"></script>
  <script src="js/popup.slicetoolbar.js"></script>
  <script src="js/slicetoolbar.js"></script>
</head>

<body>

<div id="toolbar">
  <form>

    Criteria:&nbsp;<input id="cuadroblanco" type="text" size="16" />

    <span><input id="botonslice" type="button" value="Filter" /></span>

    <input id="inverse" type="checkbox" checked />Inverse&nbsp;&nbsp;&nbsp;

```



```

    {
        window.close();
    }
};

// Returns the selected tab
WebFiltering.Popup.getSelectedTab = function(callback)
{
    chrome.tabs.getSelected(null, callback);
};

// Returns the selected window
WebFiltering.Popup.getSelectedWindow = function(callback)
{
    chrome.windows.getCurrent(callback);
};

// Returns true if this is a valid tab
WebFiltering.Popup.isValidTab = function(tab)
{
    var url = tab.url;

    // If this is a chrome URL
    if(url.indexOf("chrome://") === 0)
    {
        WebFiltering.Popup.showSimpleNotification('The Web Filtering extension does not work on internal browser pages. Please read the FAQ for more information.');
```

return false;

```
    }
    else if(url.indexOf("file://") === 0)
    {
        WebFiltering.Popup.showSimpleNotification('The Web Filtering extension does not work on local files. Please read the FAQ for more information.');
```

return false;

```
    }
    else if(url.indexOf("https://chrome.google.com/extensions/") === 0)
    {
        WebFiltering.Popup.showSimpleNotification('The Web Filtering extension does not work on the Chrome Extension Gallery. Please read the FAQ for more information.');
```

return false;

```
    }

    return true;
};

// Handles the popup loading
WebFiltering.Popup.load = function()
{
    WebFiltering.Popup.getSelectedTab(function(tab)
    {
        // TODO
    });
};

// Shows the notification
WebFiltering.Popup.showNotification = function()
{
    $("#notification").fadeIn(500);
};

// Clears a notification
WebFiltering.Popup.clearNotification = function()
{
    var notification = $("#notification");
    notification.hide();
    notification.children().remove();
};

// Shows a simple notification
WebFiltering.Popup.showSimpleNotification = function(message)

```

```

{
    WebFiltering.Popup.clearNotification();
    $("

</p>").html(message).appendTo($("#notification"));
    WebFiltering.Popup.showNotification();
};

// *****
// Handles any popup requests
WebFiltering.Popup.request = function(request, sender, sendResponse)
{
    // If the request type is a notification
    if(request.type == "notification")
    {
        WebFiltering.Popup.showSimpleNotification(request.notification);
    }
    else
    {
        sendResponse({});
    }
};
// *****

chrome.extension.onRequest.addListener(WebFiltering.Popup.request);


```

Como se ha comentado anteriormente, a lo largo de este proyecto se ha pretendido hacer un código fuente cuanto más modular mejor, por lo tanto se ha separado toda la funcionalidad destinada a disparar eventos en una nueva página llamada *popup.slicetoolbar.js*, que queda de la siguiente forma:

```

var WebFiltering = WebFiltering || {};

WebFiltering.Popup.SliceToolbar = WebFiltering.Popup.SliceToolbar || {};

// document load
$(function()
{
    // input text: load last previous search if not empty
    chrome.extension.sendRequest({type: "get-storage-item", item: "inputtext"}, function(response) {
        $("#cuadroblanco").val(response);
    });

    // filter button
    $("#botonslice").click(WebFiltering.Popup.SliceToolbar.doCentral);

    // checkbox buttons
    chrome.extension.sendRequest({type: "get-storage-item", item: "inverse"}, function(response) {
        if (response == "false") $("#inverse").get(0).checked = false;
        else $("#inverse").get(0).checked = true;
    });
    chrome.extension.sendRequest({type: "get-storage-item", item: "continuous"}, function(response) {
        if (response == "false") $("#continuous").get(0).checked = false;
        else $("#continuous").get(0).checked = true;
    });
    /* TODO: for now the implementation of this feature is not required
    chrome.extension.sendRequest({type: "get-storage-item", item: "followlinks"}, function(response) {
        if (response == "false") $("#followlinks").get(0).checked = false;
        else $("#followlinks").get(0).checked = true;
    });
    */

    // tolerance: load last previous tolerance
    chrome.extension.sendRequest({type: "get-storage-item", item: "tolerance"}, function(response) {
        if (response != null) $("#cuadrotolerancia").val(response);
        else $("#cuadrotolerancia").val(0);
    });

    // reload button
    $("#botonreload").click(WebFiltering.Popup.SliceToolbar.doCentral);

```

```

});

// document ready
$(document).ready(function()
{
    // input text: on keypressed event
    $('#cuadroblanco').keypress(function(event)
    {
        if (event.keyCode == '13')
        {
            chrome.extension.sendRequest({type: "set-storage-item", item: "inputtext", value: $('#cuadroblanco').val()},
function(response) {
                //alert("set-storage-item(inputtext): " + $('#cuadroblanco').val());
            });
            // TODO: if possible, then we have to start the slice event
            //$("#botonslice").trigger('click');
        }
    });

    // tolerance: on change event
    $("#cuadrotolerancia").change(function()
    {
        chrome.extension.sendRequest({type: "set-storage-item", item: "tolerance", value: $("#cuadrotolerancia").val()},
function(response) {
                //alert("set-storage-item(tolerance): " + $("#cuadrotolerancia").val());
            });
    });

    // checkboxes: on change event
    $("#inverse").change(function()
    {
        chrome.extension.sendRequest({type: "set-storage-item", item: "inverse", value: $("#inverse").get(0).checked},
function(response) {
                //alert("set-storage-item(inverse): " + $("#inverse").get(0).checked);
            });
    });
    $("#continuous").change(function()
    {
        chrome.extension.sendRequest({type: "set-storage-item", item: "continuous", value:
$("#continuous").get(0).checked}, function(response) {
                //alert("set-storage-item(continuous): " + $("#continuous").get(0).checked);
            });
    });
    /* TODO: for now the implementation of this feature is not required
    $("#followlinks").change(function()
    {
        chrome.extension.sendRequest({type: "set-storage-item", item: "followlinks", value:
$("#followlinks").get(0).checked}, function(response) {
                //alert("set-storage-item(followlinks): " + $("#followlinks").get(0).checked);
            });
    });
    */
});

WebFiltering.Popup.SliceToolbar.doCentral = function()
{
    var featureItem = $(this);

    WebFiltering.Popup.getSelectedTab(function(tab)
    {
        // If the tab is valid
        if(WebFiltering.Popup.isValidTab(tab))
        {
            WebFiltering.Popup.addFeatureOnTab(
                featureItem,
                tab,
                "js/slicetoolbar.js",
                "WebFiltering.SliceToolbar.doCentral(document, "
                + featureItem.val() + ", "
                + $("#cuadroblanco").val()
                + ", " + $("#cuadrotolerancia").val() + ");"
            );
        }
    });
}

```

```

    );
  }
});
};

```

En el código anterior se puede ver cómo se llama, mediante la función doCentral, a un 'content script' llamado *slicetoolbar.js*. Este 'content script' es el que tiene la mayor parte de las funcionalidad de nuestra extensión y será el encargado de interactuar con el DOM de la página a filtrar, etc.

El fichero consta del siguiente código:

```

var WebFiltering = WebFiltering || {};

WebFiltering.SliceToolbar = WebFiltering.SliceToolbar || {};

WebFiltering.SliceToolbar.webfiltering_contentDocument;

WebFiltering.SliceToolbar.webfiltering_sliceAplicado = WebFiltering.SliceToolbar.webfiltering_sliceAplicado || 0;
WebFiltering.SliceToolbar.webfiltering_AplicadoResaltar = WebFiltering.SliceToolbar.webfiltering_AplicadoResaltar || 0;
WebFiltering.SliceToolbar.webfiltering_AplicadoSiguiente = WebFiltering.SliceToolbar.webfiltering_AplicadoSiguiente || 0;

WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes = WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes ||
new Array ();
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes = WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes || new
Array ();
WebFiltering.SliceToolbar.webfiltering_arrayVisibleNodes = WebFiltering.SliceToolbar.webfiltering_arrayVisibleNodes ||
new Array();
WebFiltering.SliceToolbar.webfiltering_arrayRelevants = WebFiltering.SliceToolbar.webfiltering_arrayRelevants || new
Array();
WebFiltering.SliceToolbar.webfiltering_microformatosBuscados =
WebFiltering.SliceToolbar.webfiltering_microformatosBuscados || new Array();
WebFiltering.SliceToolbar.webfiltering_match_nodos = WebFiltering.SliceToolbar.webfiltering_match_nodos || new Array ();

WebFiltering.SliceToolbar.webfiltering_iglobal = WebFiltering.SliceToolbar.webfiltering_iglobal || 0; // keynode
WebFiltering.SliceToolbar.webfiltering_jglobal = WebFiltering.SliceToolbar.webfiltering_jglobal || 0; // hidden
WebFiltering.SliceToolbar.webfiltering_vglobal = WebFiltering.SliceToolbar.webfiltering_vglobal || 0; // visible
WebFiltering.SliceToolbar.webfiltering_Reglobal= WebFiltering.SliceToolbar.webfiltering_Reglobal || 0;

WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo =
WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo || 0;

WebFiltering.SliceToolbar.webfiltering_tolerance_level = WebFiltering.SliceToolbar.webfiltering_tolerance_level || new
Array();
WebFiltering.SliceToolbar.webfiltering_tolerance_level[0] = WebFiltering.SliceToolbar.webfiltering_tolerance_level[0] || new
Array(); // keynode
WebFiltering.SliceToolbar.webfiltering_tolerance_level[1] = WebFiltering.SliceToolbar.webfiltering_tolerance_level[1] || new
Array(); // hidden
WebFiltering.SliceToolbar.webfiltering_tolerance_level[2] = WebFiltering.SliceToolbar.webfiltering_tolerance_level[2] || new
Array(); // visible

WebFiltering.SliceToolbar.webfiltering_tolerancia = WebFiltering.SliceToolbar.webfiltering_tolerancia || 0; // valor inicial de la
tolerancia
WebFiltering.SliceToolbar.webfiltering_tolerancia_antigua = WebFiltering.SliceToolbar.webfiltering_tolerancia_antigua || 0;

WebFiltering.SliceToolbar.webfiltering_mantenerestructura = WebFiltering.SliceToolbar.webfiltering_mantenerestructura ||
1; // 1=true, 0=false
WebFiltering.SliceToolbar.webfiltering_colapsareestructura = WebFiltering.SliceToolbar.webfiltering_colapsareestructura || 0;
// 1=true, 0=false
WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes =
WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes || 0; // 1=true, 0=false
WebFiltering.SliceToolbar.webfiltering_mantenerarbol = WebFiltering.SliceToolbar.webfiltering_mantenerarbol || 1; //
1=true, 0=false
WebFiltering.SliceToolbar.webfiltering_inverso = WebFiltering.SliceToolbar.webfiltering_inverso || 0; // 1=true, 0=false
WebFiltering.SliceToolbar.webfiltering_inverso_antiguo = WebFiltering.SliceToolbar.webfiltering_inverso_antiguo || 0; //
sirve para deshacer la tolerancias cuando cambia varias veces entre inversa y normal.
WebFiltering.SliceToolbar.webfiltering_slicing_continuo = WebFiltering.SliceToolbar.webfiltering_slicing_continuo || 0;

WebFiltering.SliceToolbar.webfiltering_microformato = WebFiltering.SliceToolbar.webfiltering_microformato || "";

```

```

WebFiltering.SliceToolbar.webfiltering_texto = WebFiltering.SliceToolbar.webfiltering_texto || "";
WebFiltering.SliceToolbar.webfiltering_texto_innerHTML_antiguo =
WebFiltering.SliceToolbar.webfiltering_texto_innerHTML_antiguo || "";
WebFiltering.SliceToolbar.webfiltering_palabra_guardada = WebFiltering.SliceToolbar.webfiltering_palabra_guardada || "";

WebFiltering.SliceToolbar.webfiltering_posicion = WebFiltering.SliceToolbar.webfiltering_posicion || -1;

WebFiltering.SliceToolbar.webfiltering_textos = WebFiltering.SliceToolbar.webfiltering_textos || new Array ();
WebFiltering.SliceToolbar.webfiltering_textos_cuantos = WebFiltering.SliceToolbar.webfiltering_textos_cuantos || 0;
WebFiltering.SliceToolbar.webfiltering_numPalabrasFiltradas =
WebFiltering.SliceToolbar.webfiltering_numPalabrasFiltradas || 0;
WebFiltering.SliceToolbar.webfiltering_palabrasBuscadas = WebFiltering.SliceToolbar.webfiltering_palabrasBuscadas ||
new Array();

WebFiltering.SliceToolbar.debug = false;

// central function
WebFiltering.SliceToolbar.doCentral = function(contentDocument, method, inputtext, tolerance)
{

// save global vars to use them along all WebFiltering.SliceToolbar file
WebFiltering.SliceToolbar.webfiltering_contentDocument = contentDocument;
WebFiltering.SliceToolbar.webfiltering_tolerancia = tolerance;

if (inputtext == "")
{
// si no hay texto introducido pero se llama a resaltar anterior o siguiente
// se debe recuperar de localStorage
if ((method != "previous") || (method != "next"))
{
// envio un request a background
chrome.extension.sendRequest({type: "get-storage-item", item: "inputtext"}, function(response) {
//alert("get-storage-item(inputtext): " + response);
WebFiltering.SliceToolbar.webfiltering_texto = response;
});

// si no puede recuperar ningun valor del localStorage
if (WebFiltering.SliceToolbar.webfiltering_texto == null)
{
alert("Can't load a valid value from localStorage. Null search criteria");
return;
}
}
// si no, el criterio es nulo
else {
alert("Null search criteria");
return;
}
}
else
{
WebFiltering.SliceToolbar.webfiltering_texto = inputtext;
// envio un request a background
chrome.extension.sendRequest({type: "set-storage-item", item: "inputtext", value: inputtext}, function(response) {
//alert("set-storage-item(inputtext): " + inputtext);
});
}

switch(method)
{
case "Filter":
WebFiltering.SliceToolbar.doFilter();
break;
case "Reload":
WebFiltering.SliceToolbar.webfiltering_verpagentera(true);
break;
default:
// TODO: internationalize alerts?
alert ("Method " + method + " not found");
break;
}
}

```

```

}
};

WebFiltering.SliceToolbar.doFilter = function()
{
  /* RESUMEN:
  * Este metodo se ejecuta tras apretar el boton Filtrar.
  * Dependiendo de si la opcion mantener arbol esta
  * abierto o no se ejecuta:
  * 1.-webfiltering_nokeeptree (no mantiene arbol)
  * 2.-webfiltering_ejecucionfunc (mantiene arbol)
  * Posteriormente se guarda la palabra utilizada.
  */

  // El siguiente metodo inicializa los parametros (keeptree,keepstructure,inverse...)
  // y les asocia el respectivo valor a las variables.

  WebFiltering.SliceToolbar.webfiltering_inicializar_opciones();

  if (WebFiltering.SliceToolbar.debug == true) {
    console.log("(( doFilter ))\n\n" +
      "keepstructure   " + WebFiltering.SliceToolbar.webfiltering_mantenerestructura + "\n" +
      "formatsizeiframes " + WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes + "\n" +
      "keeptree         " + WebFiltering.SliceToolbar.webfiltering_mantenerarbol + "\n" +
      "inverse          " + WebFiltering.SliceToolbar.webfiltering_inverso + "\n" +
      "continuous       " + WebFiltering.SliceToolbar.webfiltering_slicing_continuo + "\n" +
      "tolerance        " + WebFiltering.SliceToolbar.webfiltering_tolerancia + "\n"
    );
  }

  // ***** OJO *****
  // *****
  // problema de asincronicidad
  alert(
    "Este alert evita un problema de asincronicidad, al intentar evaluar una propiedad " +
    "cuyo valor se encuentra pendiente de ser actualizado en callback.\n\n" +
    "Para más información visite:\n"+
    "http://code.google.com/p/chromium/issues/detail?id=54257"
  );
  // *****

  WebFiltering.SliceToolbar.webfiltering_texto_particionar();

  if (WebFiltering.SliceToolbar.webfiltering_texto == "")
  {
    if (WebFiltering.SliceToolbar.webfiltering_microformato == "") // No se ha especificado ningun criterio ni microformato
    {
      alert("Tanto el microformato como el criterio de busqueda estan vacios, debe introducir al menos, una palabra en alguno de ellos");
      return;
    }
    else // Se ha especificado un microformato pero ningun criterio
    {
      alert("Voy a filtrar por microformato");
      WebFiltering.SliceToolbar.webfiltering_filtrarmicroformatos(content.document);
      WebFiltering.SliceToolbar.webfiltering_numMicroformatosFiltrados++;
      WebFiltering.SliceToolbar.webfiltering_guardarhistorico("microformatos");
      alert("Voy a mostrar...");
      WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia();
      alert("Voy a mostrar tambien el path...");
      WebFiltering.SliceToolbar.webfiltering_showpathstobody();
      return;
    }
  }

  if (WebFiltering.SliceToolbar.webfiltering_microformato != "") // Se ha especificado un microformato y un criterio
  {
    alert("Voy a filtrar por microformato y despues por el criterio");
    WebFiltering.SliceToolbar.webfiltering_filtrarmicroformatos(content.document);
    WebFiltering.SliceToolbar.webfiltering_numPalabrasFiltradas++;
    WebFiltering.SliceToolbar.webfiltering_numMicroformatosFiltrados++;
  }
}

```

```

// TODO: guardarhistorico debe reemplazarse por los accesos a localStorage
//WebFiltering.SliceToolbar.webfiltering_guardarhistorico("palabras");
//WebFiltering.SliceToolbar.webfiltering_guardarhistorico("microformatos");
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia();
return;
}

// Se ha especificado un criterio pero ningun microformato
WebFiltering.SliceToolbar.webfiltering_numPalabrasFiltradas++;

// Funcionalidad de Harvester.js

// TODO: Harvester (obtener de localStorage como le resto de opciones)
//if(document.getElementById("checkbox_multiple_webpage").checked)
if (false)
{
webfiltering_HarvesterMain();
}
else
{
var pagina = WebFiltering.SliceToolbar.webfiltering_contentDocument;

if (WebFiltering.SliceToolbar.webfiltering_mantenerarbol == 0)
{
if (WebFiltering.SliceToolbar.debug) alert("nokeepree");
WebFiltering.SliceToolbar.webfiltering_nokeepree(pagina);
if (WebFiltering.SliceToolbar.debug) alert("nokeepree done");
}
else
{
if (WebFiltering.SliceToolbar.debug) alert("ejecucionfunc");
WebFiltering.SliceToolbar.webfiltering_ejecucionfunc(pagina);
if (WebFiltering.SliceToolbar.debug) alert("ejecucionfunc done");

if(WebFiltering.SliceToolbar.webfiltering_inverso==1)
{
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia("inverso", "creciente", 0,
WebFiltering.SliceToolbar.webfiltering_tolerancia);
}
else
{
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia("normal", "creciente", 0,
WebFiltering.SliceToolbar.webfiltering_tolerancia);
}
}

WebFiltering.SliceToolbar.webfiltering_sliceAplicado = 1; //ya se ha aplicado slice

//if(webfiltering_plugin) //Se encarga de anyadir una palabra al dropbox de cuadro en blanco.
//{
// webfiltering_guardarhistorico("palabras");
// webfiltering_numPalabrasFiltradas++;
//}
};

WebFiltering.SliceToolbar.webfiltering_texto_particionar = function ()
{
// RESUMEN:
// Metodo que se encarga de coger la frase que hay
// en el cuadro de texto y particionarlo siguiente reglas
// AND, OR y literal.

//Trim del inputtext
WebFiltering.SliceToolbar.webfiltering_texto_trim();

//Inicializar variables
var las_palabras = new Array();
var las_palabras_cuantos = 0;
var palabra = "";
WebFiltering.SliceToolbar.webfiltering_textos = new Array ();

```

```

WebFiltering.SliceToolbar.webfiltering_textos_cuantos = 0;

var i;
if(WebFiltering.SliceToolbar.webfiltering_texto.indexOf("") != -1)
{
    var las_palabras_intermedio = WebFiltering.SliceToolbar.webfiltering_texto.split(" ");
    //ahora tengo un vector, pero tengo que comprobar donde hay comillas y unir las
    for(var i=0; i<las_palabras_intermedio.length; i++)
    {
        if(las_palabras_intermedio[i].indexOf("")==0) // La ' esta al principio
        {
            palabra=las_palabras_intermedio[i].substring(1,las_palabras_intermedio[i].length);
            var continuar=true;
            for(var j=i+1;j<las_palabras_intermedio.length && continuar;j++)
            {
                //alert("palabra "+las_palabras_intermedio[j]+" posicion comilla:"+las_palabras_intermedio[j].indexOf("").toString()+"
                tamanyo:"+las_palabras_intermedio[j].length.toString());
                if(las_palabras_intermedio[j].toString().indexOf("")==las_palabras_intermedio[j].toString().length-1) //La ' esta al final
                {
                    palabra += " " + las_palabras_intermedio[j].toString().substring(0,las_palabras_intermedio[j].toString().length-1);
                    continuar = false;
                }
                else
                {
                    palabra += " " + las_palabras_intermedio[j];
                }
                i = j;
            }
            //Anyadir palabra
            las_palabras[las_palabras_cuantos] = palabra;
            las_palabras_cuantos++;
        }
        else
        {
            las_palabras[las_palabras_cuantos] = las_palabras_intermedio[i];
            las_palabras_cuantos++;
        }
    }
}
else
{
    las_palabras = WebFiltering.SliceToolbar.webfiltering_texto.split(" ");
    las_palabras_cuantos = las_palabras.length;
}
//alert("las_palabras: " + las_palabras);

//Formatear las_palabras
WebFiltering.SliceToolbar.webfiltering_textos[0] = new webfiltering_textos_clase();
WebFiltering.SliceToolbar.webfiltering_textos[0].operator = "AND";
WebFiltering.SliceToolbar.webfiltering_textos[0].palabra = "PRUEBA_ELIMINAR";
WebFiltering.SliceToolbar.webfiltering_textos[0].resultado = true;
WebFiltering.SliceToolbar.webfiltering_textos[0].resultado_atributos = true;
WebFiltering.SliceToolbar.webfiltering_textos[0].resultado_innerHTML = true
WebFiltering.SliceToolbar.webfiltering_textos_cuantos++;

for(i=0; i<las_palabras_cuantos; i++)
{
    //alert("WebFiltering.SliceToolbar.webfiltering_textos_cuantos: " +
    WebFiltering.SliceToolbar.webfiltering_textos_cuantos);
    palabra = las_palabras[i].toLowerCase();
    if(palabra == "and" || palabra == "&&" || palabra == "or" || palabra == "|")
    {
        WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos] = new
webfiltering_textos_clase();
        if(palabra == "and" || palabra == "&&")
        {
            WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos].operator =
"AND";
        }
        else
        {

```



```

        WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos].operator = "OR";
    }
}
else
{
    if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos] == null)
    {
        WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos] = new
webfiltering_textos_clase();
    }
}

WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos].palabra=las_palabras[
].toLowerCase();
    WebFiltering.SliceToolbar.webfiltering_textos_cuantos++;
}
}
};

WebFiltering.SliceToolbar.webfiltering_texto_trim = function()
{
    // RESUMEN:
    // Funcion que se encarga de quitar espacios (del final y principio) del
    // inputtext (que ahora esta copiado en webfiltering_texto)

    //Empieza por espacio
    while(WebFiltering.SliceToolbar.webfiltering_texto.charAt(0) == " " &&
WebFiltering.SliceToolbar.webfiltering_texto.length>0)
    {
        WebFiltering.SliceToolbar.webfiltering_texto =
WebFiltering.SliceToolbar.webfiltering_texto.substring(1,WebFiltering.SliceToolbar.webfiltering_texto.length-1);
    }

    //Termina por espacio
    while(WebFiltering.SliceToolbar.webfiltering_texto.charAt(WebFiltering.SliceToolbar.webfiltering_texto.length-1) == " " &&
WebFiltering.SliceToolbar.webfiltering_texto.length>0)
    {
        WebFiltering.SliceToolbar.webfiltering_texto =
WebFiltering.SliceToolbar.webfiltering_texto.substring(0,WebFiltering.SliceToolbar.webfiltering_texto.length-2);
    }
};

WebFiltering.SliceToolbar.webfiltering_ejecucionfunc = function(pagina)
{
    // El ejecucionfunc el primer metodo para limpiar la pagina

    /* RESUMEN:
    * Este metodo se encarga de realizar un filtro slice,
    * dependiendo de si ya se ha ejecutado o no.
    * Si no se ha ejecutado se realizara entre el codigo
    * html pero si ya se ha ejecutado, se realizara solo
    * entre los nodos del vector arraymuestra.
    * A ambos metodos le pasa un parametro a true que oculta
    * los nodos que no coinciden con la busqueda.
    */

    if(WebFiltering.SliceToolbar.webfiltering_sliceAplicado == 1)
    {
        //alert("Slice sobre slice aplicado");
        // HORROR: aqui nunca entraba --> a != a
        //if(WebFiltering.SliceToolbar.webfiltering_inverso_antiguo != WebFiltering.SliceToolbar.webfiltering_inverso_antiguo)
        //{}
        // WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia();
        //}
        var resultado_slicing = WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_arraymuestra(true);
        if(!resultado_slicing)
        {
            alert("No se ha encontrado ninguna coincidencia");
            WebFiltering.SliceToolbar.webfiltering_verpagentera(true);
            return;
        }
    }
}
}

```

```

else
{
  if(WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes == 1)
  {
    //Para que nos convierta los iframes en divs.
    alert("Voy a formateariframes de webfiltering_contentDocument");
    WebFiltering.SliceToolbar.webfiltering_formateariframes(WebFiltering.SliceToolbar.webfiltering_contentDocument);
    //WebFiltering.SliceToolbar.webfiltering_formateariframes(pagina);
    alert("Hecho");
  }

  //alert("pagina.getElementsByTagName(\"FRAME\").length: " + pagina.getElementsByTagName("FRAME").length);
  if(pagina.getElementsByTagName("FRAME").length > 0)
  {
    for(var i = 0; i < pagina.getElementsByTagName("FRAME").length; i++)
    {
      WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina.getElementsByTagName("FRAME")[i].contentDo
cument.body, true);
    }
  }
  else
  {
    WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina.body, true);
  }
}

if(WebFiltering.SliceToolbar.webfiltering_iglobal == 0)
{
  // TODO: translate?
  alert("ejecucion_func: La palabra buscada no se encuentra");
  if(WebFiltering.SliceToolbar.webfiltering_slicing_continuo == 0)
  {
    WebFiltering.SliceToolbar.webfiltering_verpagentera(true); // reload
  }
}
};

WebFiltering.SliceToolbar.webfiltering_ejecucionfunc_sinocultar = function()
{
  var pagina = WebFiltering.SliceToolbar.webfiltering_contentDocument;

  // RESUMEN:
  // Metodo que se encarga de buscar los nodos que coinciden con
  // los criterios de busqueda. Se diferencia de webfiltering_ejecucionfunc
  // en que los nodos que no coinciden no los oculta.

  WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes = new Array ();
  WebFiltering.SliceToolbar.webfiltering_iglobal = 0;
  WebFiltering.SliceToolbar.webfiltering_arrayVisibleNodes = new Array();
  WebFiltering.SliceToolbar.webfiltering_posicion=-1;

  // obtener el valor almacenado de la propiedad inverso, para ello uso llamadas al localStorage
  // y no lo obtengo del document del popup (de donde no puedo obtenerlo al no ser el document activo)
  //WebFiltering.SliceToolbar.webfiltering_inverso=document.getElementById("checkbox_slice_inverso").checked;
  chrome.extension.sendRequest({type: "get-storage-item", item: "inverse"}, function(response) {
    if (response != null)
    {
      if (response == "false") WebFiltering.SliceToolbar.webfiltering_inverso = 0;
      else if (response == "true") WebFiltering.SliceToolbar.webfiltering_inverso = 1;
    }
  });

  if(WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes == 1)
  {
    WebFiltering.SliceToolbar.webfiltering_formateariframes(pagina); // para que nos convierta los iframes en divs.
  }

  if(pagina.getElementsByTagName("FRAME").length > 0)
  {
    for(var i=0;i<pagina.getElementsByTagName("FRAME").length;i++)

```

```

{
WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina.getElementsByTagName("FRAME")[i].contentDo
cument.body, false);
}
}
else
{
WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina.body, false);
}

/* TODO: implement, habra que usar accesos tirando de localStorage
if(WebFiltering.SliceToolbar.webfiltering_iglobal != 0 && webfiltering_plugin)
{
WebFiltering.SliceToolbar.webfiltering_guardarhistorico("palabras");
}
*/

};
WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_arraymuestra =function (ocultar)
{
// RESUMEN:
// El metodo se encarga de buscar entre todos los nodos
// del arraymuestra aquellos que coinciden con los criterios
// de busqueda. Si coinciden se guardan en el arraymuestra de
// nuevo o sino se guardan al final del arrayocultos.

var i = 0;
var ArrayAuxiliar = new Array();
var ArrayAuxiliarContador = 0;
var resultado;

//Quitamos la tolerancia y la dejamos a 0.
var opcion="";
if(WebFiltering.SliceToolbar.webfiltering_inverso==1)
{
opcion="inverso";
}
else
{
opcion="normal";
}

for(i=0; i<WebFiltering.SliceToolbar.webfiltering_iglobal; i++)
{

WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_hasta(WebFiltering.SliceToolbar.webfiltering_tolerancia,
0,opcion,WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i],WebFiltering.SliceToolbar);
}
WebFiltering.SliceToolbar.webfiltering_tolerancia=0;

for(i=0;i<WebFiltering.SliceToolbar.webfiltering_iglobal;i++)
{
//Le aplico la tolerancia al elemento
var nodo = WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i];
for(var j=1; j<WebFiltering.SliceToolbar.webfiltering_tolerancia+1; j++)
{
nodo=nodo.parentNode;
if(nodo.tagName=="BODY" || nodo.tagName=="HTML") break;
}

resultado = WebFiltering.SliceToolbar.webfiltering_contiene_coincidencia(nodo);

if(resultado!=-1 && resultado!=-2) // Lo encuentra
{
//Los voy ya introduciendo en array_oculto
if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{

WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i],oc
ultar,WebFiltering.SliceToolbar);

```

```

//COMPROBACION VISIBLES.
if (WebFiltering.SliceToolbar.webfiltering_mantenerestructura==1 ||
WebFiltering.SliceToolbar.webfiltering_colapsareestructura==1)
{

WebFiltering.SliceToolbar.webfiltering_quitar_VisibleNode_asociado_a_KeyNode(WebFiltering.SliceToolbar.webfiltering_arryKeyNodes[i],i,ocultar,WebFiltering.SliceToolbar);
}
}
else
{
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i].setAttribute("Z","K"+ArrayAuxiliarContador.toString());
WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][ArrayAuxiliarContador] =
WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][i];
ArrayAuxiliar[ArrayAuxiliarContador] = WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i];
ArrayAuxiliarContador++;
}
}
else //No encuentra coincidencia
{
if(WebFiltering.SliceToolbar.webfiltering_inverso == 0) // false
{

WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i],ocultar,WebFiltering.SliceToolbar);
if (WebFiltering.SliceToolbar.webfiltering_mantenerestructura==1 ||
WebFiltering.SliceToolbar.webfiltering_colapsareestructura==1)
{
//Hay que ocultar a los padres si no tienen mas hijos.

WebFiltering.SliceToolbar.webfiltering_quitar_VisibleNode_asociado_a_KeyNode(WebFiltering.SliceToolbar.webfiltering_arryKeyNodes[i],i,ocultar, WebFiltering.SliceToolbar);
}
}
else
{
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i].setAttribute("Z","K"+ArrayAuxiliarContador.toString());
WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][ArrayAuxiliarContador] =
WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][i];
ArrayAuxiliar[ArrayAuxiliarContador] = WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i];
ArrayAuxiliarContador++;
}
}
}

if(ArrayAuxiliarContador>0)
{
//alert("Entra teniendo el iglobal_"+this.webfiltering_iglobal+" y el arrayauxiliar:"+ArrayAuxiliarContador);
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes = new Array();
WebFiltering.SliceToolbar.webfiltering_iglobal = 0;
for(i=0;i<ArrayAuxiliarContador;i++)
{
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]=ArrayAuxiliar[i];
WebFiltering.SliceToolbar.webfiltering_iglobal++;
}
return true;
}
else
{
for(i=0;i<ArrayAuxiliarContador;i++)
{
ArrayAuxiliar[i].style.visibility="";
}
return false;
}
};

WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html = function(pagina, ocultar)
{
// RESUMEN:
// Este metodo se encarga en encontrar sobre los hijos del parametro 'pagina' los hijos

```

```

// que coinciden con la busqueda.
// El parametro ocultar es para ocultar o no los parametros que coinciden con la busqueda.
// Este metodo es utilizado desde el boton filtrar (ocultar=true) pero tambien desde el
// boton siguiente, anterior y resaltar (ocultar=false).
// Hay que tener en cuenta que si la opcion slicing inverso se encuentra activa, se guarda
// en arraymuestra los nodos no coinciden mientras si esta desactivada es al revés.

var resultado = 0;
var nodo, nodo_result, nodo_TextNode;
var cuerpo = "";
var pagina_recurso;

for(var i=0; i<pagina.childNodes.length; i++)
{
if(WebFiltering.SliceToolbar.webfiltering_contiene_tagNameValido(pagina.childNodes[i]))
{
if(pagina.childNodes[i].tagName == "IFRAME" || pagina.childNodes[i].tagName == "FRAME")
{
//cuerpo = pagina.childNodes[i].contentDocument.body.innerHTML.toLowerCase();
pagina_recurso = pagina.childNodes[i].contentDocument.body;
}
else
{
pagina_recurso = pagina.childNodes[i];
}

nodo_result = WebFiltering.SliceToolbar.webfiltering_contiene_coincidencia(pagina.childNodes[i]);
//alert("nodo_result: " + nodo_result);

switch(nodo_result)
{
case -2: // Nada coincide y encima es de tipo NodeText
nodo_TextNode = WebFiltering.SliceToolbar.webfiltering_convertir_textNode(pagina.childNodes[i]);

if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
if(pagina.childNodes.length>1)
{
resultado = 1;
WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(nodo_TextNode, ocultar);
}
}
else
{
WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(nodo_TextNode, ocultar);
}
break;

case -1: // Nada coincide
if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
resultado = 1;
WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(pagina.childNodes[i], ocultar);
}
else
{
WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
}
break;

case 4:
// resultado=true ademas significa que es un simple texto, osea hay que modificarlo y cambiarlo por otro nodo
nodo_TextNode = WebFiltering.SliceToolbar.webfiltering_convertir_textNode(pagina.childNodes[i]);

if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
if(pagina.childNodes.length>1)
{
WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(nodo_TextNode, ocultar);
}
}
}
}
}

```

```

else
{
  if(pagina.childNodes.length>=1)
  {
    resultado = 1;
    WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(nodo_TextNode, ocultar);
  }
  else
  {
    //alert("ELSE case 4.\nEl nodo con valor:"+pagina.childNodes[i].nodeValue+"\nY su padre es
tag:"+pagina.tagName);
  }
}
break;

case 0: // resultado=true resultado_atributos=false resultado_innerHTML=false
case 1: // resultado=true resultado_atributos=true resultado_innerHTML=false
if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
  resultado = 1;
  WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
}
else
{
  resultado = 1;
  WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(pagina.childNodes[i], ocultar);
}
break;

case 3: //resultado=true resultado_atributos=true resultado_innerHTML=true
if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
  resultado = 1;
  WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
}
else
{
  resultado = 1;
  WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(pagina.childNodes[i], ocultar);
  if(!WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo)
  {
    WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina_recurso, ocultar);
  }
}
break;

case 2: // resultado=true resultado_atributos=false resultado_innerHTML=true
if(WebFiltering.SliceToolbar.webfiltering_inverso == 1) // true
{
  resultado=1;
  var webfiltering_iglobal_antiguo = WebFiltering.SliceToolbar.webfiltering_iglobal;
  var webfiltering_jglobal_antiguo = WebFiltering.SliceToolbar.webfiltering_jglobal;
  var resultado_funcion = WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina_recurso,
ocultar);
  if(resultado_funcion == 0)
  {
    WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
  }
  else
  {
    WebFiltering.SliceToolbar.webfiltering_anyadir_a_visibleNodes(pagina.childNodes[i], ocultar);
    var valor=",";
    for(var k=webfiltering_iglobal_antiguo; k<WebFiltering.SliceToolbar.webfiltering_iglobal; k++)
    {
      valor += k.toString()+",";
    }
    WebFiltering.SliceToolbar.webfiltering_tolerance_level[2][WebFiltering.SliceToolbar.webfiltering_vglobal-
1].hijos_keynode = valor;
  }
}
else

```

```

{
    resultado = 1;
    var webfiltering_iglobal_antiguo = WebFiltering.SliceToolbar.webfiltering_iglobal;
    var webfiltering_iglobal_antiguo = WebFiltering.SliceToolbar.webfiltering_iglobal;
    var resultado_funcion = WebFiltering.SliceToolbar.webfiltering_calcular_slicing_sobre_html(pagina_recurso,
ocultar);

    if(resultado_funcion == 0)
    {
        WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes(pagina.childNodes[i], ocultar);
        //Hay que volver a mostrar a los hijos y quitarlos del arrayHidden
        var hijos_ocultos = WebFiltering.SliceToolbar.webfiltering_iglobal-webfiltering_iglobal_antiguo;

        for(var nodoOculto = 0; nodoOculto<hijos_ocultos;nodoOculto++)
        {
            if(WebFiltering.SliceToolbar.webfiltering_colapsarestructura == 1)
            {

WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[webfiltering_iglobal_antiguo+nodoOculto].style.display = "";
                WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[webfiltering_iglobal_antiguo+nodoOculto] = null;
            }
            else //WebFiltering.SliceToolbar.webfiltering_mantenerestructura==1
            {

WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[webfiltering_iglobal_antiguo+nodoOculto].style.visibility =
"visible";
                WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[webfiltering_iglobal_antiguo+nodoOculto] = null;
            }
        }
        WebFiltering.SliceToolbar.webfiltering_iglobal=webfiltering_iglobal_antiguo;
    }
    else
    {
        if(WebFiltering.SliceToolbar.webfiltering_mantenerestructura==0 &&
WebFiltering.SliceToolbar.webfiltering_colapsarestructura==0)
        {
            WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
        }
        else
        {
            if(ocultar)// && WebFiltering.SliceToolbar.webfiltering_mantenerestructura==1)
            {
                WebFiltering.SliceToolbar.webfiltering_anyadir_a_visibleNodes(pagina.childNodes[i], ocultar);
                var valor ="";
                for(var k=webfiltering_iglobal_antiguo; k<WebFiltering.SliceToolbar.webfiltering_iglobal; k++)
                {
                    valor += k.toString() + ",";
                }
                WebFiltering.SliceToolbar.webfiltering_tolerance_level[2][WebFiltering.SliceToolbar.webfiltering_vglobal-
1].hijos_keynode = valor;
            }
            else
            {
                WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes(pagina.childNodes[i], ocultar);
            }
        }
    }
    break;

default:
    break;
}
}
else //No contiene tagValido
{
    //No hacer nada, podriamos ocultarlo pero de momento simplemente lo ignoramos.
    //alert("no contiene tagValido");
}
}
//alert("return resultado: " + resultado);

```

```

//return resultado;
};

WebFiltering.SliceToolbar.webfiltering_contiene_tagNameValido = function(nodo)
{
  // RESUMEN: Es encarga de comprobar si el tagName del nodo es valido.
  // Por ejemplo, tiene que ignorar el nodo TITLE o SCRIPT.

  switch(nodo.nodeType)
  {
    case 8: // Node.COMMENT_NODE
      return false;
      break;

    case 3:
      if(nodo.nodeValue!=null)
      {
        var texto = nodo.nodeValue;

        while(texto.indexOf(" ") != -1)
          texto = texto.replace(" ", "");

        while(texto.indexOf("\n") != -1)
          texto = texto.replace("\n", "");

        while(texto.indexOf("\t") != -1)
          texto = texto.replace("\t", "");

        if(texto.length>0)
          return true;
        else
          return false;
      }
      else
      {
        return true;
      }
      break;

    case 10: // Node.DOCUMENT_TYPE_NODE
      return false;
      break;

    default:
      if(nodo.style.visibility == "hidden")
      {
        return false;
        break;
      }
      switch(nodo.tagName.toLowerCase())
      {
        case "head":
        case "title":
        case "meta":
        case "script":
        case "style":
        case "br":
        case "webfiltering":
          return false;
          break;
        default:
          return true;
          break;
      }
      break;
  }
  return true;
};

WebFiltering.SliceToolbar.webfiltering_contiene_coincidencia = function (nodo)
{

```



```

// Codigos de Retorno de la funcion
// -2 No hay coincidencia y es de tipo TextNode
// -1 No hay coincidencia.
// 0 Se encuentra en el resultado
// 1 Se encuentra solo en el atributo
// 2 Se encuentra solo en el innerHTML
// 3 Se encuentra en ambos
// 4 Es un nodo de tipo TextNode y tiene coincidencia

var innerHTML = "";
var j = 0;
var resultado = false;

//alert("nodo type = " + nodo.nodeType);

if(nodo.nodeType == 3 || nodo.nodeType == 8) //Nodo tipo #text o #comentario
{
    resultado = true;

    //alert("WebFiltering.SliceToolbar.webfiltering_textos_cuantos: " +
    WebFiltering.SliceToolbar.webfiltering_textos_cuantos);

    for(var i=1; i<WebFiltering.SliceToolbar.webfiltering_textos_cuantos; i++)
    {
        if(WebFiltering.SliceToolbar.webfiltering_textos[i].operator == "AND")
        {
            if(nodo.nodeValue.toLowerCase().indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
            {
                resultado = true;
            }
            else
            {
                resultado = false;
            }
        }
        else //operator=="OR";
        {
            if(resultado == false)
            {
                if(nodo.nodeValue.indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
                {
                    resultado = true;
                }
            }
        }
    }

    //alert("resultado: " + resultado);
    if (resultado == true)
        return 4;
    else
        return -2;
}
else
{
    //alert("tagName del nodo: " + nodo.tagName);

    if(nodo.tagName == "FRAME" || nodo.tagName == "IFRAME")
    {
        //alert("es un (i)frame");
        innerHTML = nodo.contentDocument.body.innerHTML.toLowerCase();
    }
    else if (nodo.tagName == "NOSCRIPT")
    {
        // TODO: test with google and some alerts
        return -1;
    }
    else
    {
        innerHTML = nodo.innerHTML.toLowerCase();
    }
}
}

```

```

//alert("innerHTML del nodo: " + innerHTML);
//alert("WebFiltering.SliceToolbar.webfiltering_textos_cuantos: " +
WebFiltering.SliceToolbar.webfiltering_textos_cuantos);

//Metodo completo (AND y OR)
for(var i=1; i<WebFiltering.SliceToolbar.webfiltering_textos_cuantos; i++)
{
    if(WebFiltering.SliceToolbar.webfiltering_textos[i].operator == "AND")
    {
        //Inicializo.
        WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = false;
        WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = false;
        WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = false;

        //Como resultado=resultado_innerHTML || resultado_atributos;
        if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado == false)
        {
            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = false;
            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = false;
            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = false;
        }
        else
        {
            //Atributos.
            if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_atributos ||
WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado)
            {
                for(j=0; j<nodo.attributes.length; j++)
                {
                    if(nodo.attributes[j].value.toLowerCase().indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
                    {
                        if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_atributos)
                        {
                            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true;
                            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = true;
                        }
                        else
                        {
                            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true;
                        }
                    }
                }
            }
        }

        //InnerHTML
        if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_innerHTML == true ||
WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado == true)
        {
            if(innerHTML.indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
            {
                if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_innerHTML == true)
                {
                    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true
                    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = true;
                }
                else
                {
                    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true
                }
            }
        }
    }
}
else //OR
{
    //Inicializo.
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = false;
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = false;
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = false;
}
}
}

```

```

if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado)
{
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true;
}

//Atributos
if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_atributos)
{
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true;
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = true;
}
else
{
    for(j=0; j<nodo.attributes.length; j++)
    {
        if(nodo.attributes[j].value.toLowerCase().indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
        {
            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true;
            WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_atributos = true;
        }
    }
}

//InnerHTML
if(WebFiltering.SliceToolbar.webfiltering_textos[i-1].resultado_innerHTML == true)
{
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true
    WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = true;
}
else
{
    if(innerHTML.indexOf(WebFiltering.SliceToolbar.webfiltering_textos[i].palabra) != -1)
    {
        WebFiltering.SliceToolbar.webfiltering_textos[i].resultado = true
        WebFiltering.SliceToolbar.webfiltering_textos[i].resultado_innerHTML = true;
    }
}
}

if (WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-1] == null)
{
    return -1;
}

if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-1].resultado ==
false)
{
    return -1;
}

if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-
1].resultado_atributos == true)
{
    if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-
1].resultado_innerHTML == true)
        return 3;
    else
        return 1;
}
else
{
    if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-
1].resultado_innerHTML == true)
        return 2;
    else
        return 0;
}

if(WebFiltering.SliceToolbar.webfiltering_textos[WebFiltering.SliceToolbar.webfiltering_textos_cuantos-

```

```

1].resultado_innerHTML == true)
{
  return 1;
}

return 2;
}
};

```

```

WebFiltering.SliceToolbar.webfiltering_convertir_textNode = function(nodo)
{
  // RESUMEN: Se encarga de convertir un nodo de tipo TextNode
  // a un nodo de tipo (que se puede colorear entre otras cosas).
  // Coge el contenido del nodo y lo introduce en un tipo label. Despues, llama
  // al padre y cambia el parametro nodo por el label creado.
  var elemento_a = document.createElement("label");
  elemento_a.innerHTML = nodo.nodeValue;
  nodo.parentNode.replaceChild(elemento_a, nodo);
  return elemento_a;
};

```

```

WebFiltering.SliceToolbar.webfiltering_anyadir_a_keyNodes = function(elnodo, ocultar)
{
  //alert("se llamo a anyadir a keyNodes: " + elnodo.innerHTML);
  if(ocultar)
  {
    elnodo.style.visibility = "visible";
  }
  WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[WebFiltering.SliceToolbar.webfiltering_iglobal] = elnodo;
  WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][WebFiltering.SliceToolbar.webfiltering_iglobal] = new
  webfiltering_tolerance_level_class();
  WebFiltering.SliceToolbar.webfiltering_tolerance_level[0][WebFiltering.SliceToolbar.webfiltering_iglobal].level = 0;
  if(!WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo)
  {
    elnodo.setAttribute("HHH", "K" + WebFiltering.SliceToolbar.webfiltering_iglobal.toString());
  }
  WebFiltering.SliceToolbar.webfiltering_iglobal++;
};

```

```

WebFiltering.SliceToolbar.webfiltering_anyadir_a_hiddenNodes = function(elnodo, ocultar)
{
  //alert("se llamo a anyadir a hiddenNodes: " + elnodo.innerHTML);
  if(ocultar)
  {
    if(elnodo == null)
    {
      return;
    }

    if(elnodo.tagName == null)
    {
      //n.style.visibility="hidden";
      var nuevo_nodo = webfiltering_convertir_textNode(elnodo);
      nuevo_nodo.style.visibility = "hidden";
      elnodo.parentNode.replaceChild(nuevo_nodo, elnodo);
      elnodo = nuevo_nodo;
    }
    else
    {
      if(WebFiltering.SliceToolbar.webfiltering_colapsarestructura == 1)
      {
        elnodo.style.display = "none";
      }
      else
      {
        elnodo.style.visibility = "hidden";
      }
    }
  }
}

WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[WebFiltering.SliceToolbar.webfiltering_iglobal] = elnodo;
WebFiltering.SliceToolbar.webfiltering_tolerance_level[1][WebFiltering.SliceToolbar.webfiltering_iglobal]=new

```

```

webfiltering_tolerance_level_class();
WebFiltering.SliceToolbar.webfiltering_tolerance_level[1][WebFiltering.SliceToolbar.webfiltering_iglobal].level_inverso=0;
if(!WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo)
{
    elnodo.setAttribute("HHH","H"+WebFiltering.SliceToolbar.webfiltering_iglobal.toString());
}
WebFiltering.SliceToolbar.webfiltering_iglobal++;
};

WebFiltering.SliceToolbar.webfiltering_anyadir_a_visibleNodes = function(elnodo, ocultar)
{
    //alert("se llamo a anyadir a visibleNodes: " + elnodo.innerHTML);
    elnodo.style.visibility="";
    WebFiltering.SliceToolbar.webfiltering_arrayVisibleNodes[WebFiltering.SliceToolbar.webfiltering_vglobal]=elnodo;
    WebFiltering.SliceToolbar.webfiltering_tolerance_level[2][WebFiltering.SliceToolbar.webfiltering_vglobal]=new
webfiltering_tolerance_level_class();

WebFiltering.SliceToolbar.webfiltering_tolerance_level[2][WebFiltering.SliceToolbar.webfiltering_vglobal].hijos_keynode="";
if(!WebFiltering.SliceToolbar.webfiltering_multiple_webpages_activo)
{
    elnodo.setAttribute("HHH","V" + WebFiltering.SliceToolbar.webfiltering_vglobal.toString());
}
WebFiltering.SliceToolbar.webfiltering_vglobal++;
};

WebFiltering.SliceToolbar.webfiltering_nodo_colorear = function (nodo, enfocar)
{
    // RESUMEN: Este metodo se encargar de colorear o bien
    // colorea el fondo o bien colorea la palabra.
    // Si el parametro enfocar es true, entonces lo enfoca [focus() ].

    var resultado_coincidencia = WebFiltering.SliceToolbar.webfiltering_contiene_coincidencia(nodo);

    //alert("resultado coincidencia: " + resultado_coincidencia);

    if(resultado_coincidencia == 0 || resultado_coincidencia == 1)
    {
        //significa que la coincidencia se encuentra en el VALOR de un atributo
        if(enfocar)
        {
            nodo.focus();
        }
        nodo.style.background = "yellow";
        WebFiltering.SliceToolbar.webfiltering_texto_innerHTML = "";
    }
    else
    {
        WebFiltering.SliceToolbar.webfiltering_texto_innerHTML = nodo.innerHTML;
        var texto = nodo.innerHTML;

        //alert("texto: " + texto);

        for(var j=1; j<WebFiltering.SliceToolbar.webfiltering_textos_cuantos; j++)
        {
            var webfiltering_posicion_texto =
texto.toLowerCase().indexOf(WebFiltering.SliceToolbar.webfiltering_textos[j].palabra.toLowerCase());
            if(webfiltering_posicion_texto != -1)
            {
                var texto_aux = texto.substr(0, webfiltering_posicion_texto);
                var texto_aux2 = "<a style='color: black; background:yellow'" +
WebFiltering.SliceToolbar.webfiltering_textos[j].palabra + "</a>";
                var texto_aux3 = texto.substr(webfiltering_posicion_texto +
WebFiltering.SliceToolbar.webfiltering_textos[j].palabra.length);
                texto = texto_aux + texto_aux2 + texto_aux3;
                //alert("texto (intermedio): " + texto);
            }
        }
        //alert("nuevo texto: " + texto);
        nodo.innerHTML = texto;
        if(enfocar)
        {

```

```

    nodo.focus();
  }
}
};

WebFiltering.SliceToolbar.webfiltering_nodo_descolorear = function(nodo)
{
  // RESUMEN:
  // Este metodo se encarga de 'descolorear' un nodo
  // bien quitando el fondo o la palabra coloreada.

  if(WebFiltering.SliceToolbar.webfiltering_texto_innerHTML.length == 0)
  {
    nodo.style.background = "none";
  }
  else
  {
    nodo.innerHTML = WebFiltering.SliceToolbar.webfiltering_texto_innerHTML;
  }
};

WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia = function(opcion, sentido, tolerancia_desde,
tolerancia_hasta)
{
  if(tolerancia_desde == tolerancia_hasta)
  {
    // TODO: tirar de localStorage si es necesario para almacenar la propiedad
    //webfiltering_setIntegerPreference("slice.tolerance", tolerancia_hasta);
    return;
  }

  var elpadre, elpadre_atributo, elpadre_position, elpadre_fila;

  if(opcion=="normal")
  {
    if(sentido=="creciente")
    {
      for(var i=0; i<WebFiltering.SliceToolbar.webfiltering_iglobal; i++)
      {
        var seguir=true;

        elpadre =
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_desde(tolerancia_desde, tolerancia_hasta, WebFiltering.
SliceToolbar.webfiltering_arrayKeyNodes[i], WebFiltering.SliceToolbar);

        if(elpadre.tagName!="BODY" && elpadre.tagName!="HTML")
        {
          //alert("llamando a mostrarcontoloerancia_nodo_hasta");
          elpadre = WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_hasta(tolerancia_desde,
tolerancia_hasta, elpadre, opcion, i, WebFiltering.SliceToolbar);
        }

        if(elpadre.tagName=="BODY" || elpadre.tagName=="HTML")
        {
          // Este if se debe a que si llegamos al nodo mas alto con que se realice el mostrar
          // desde este nodo ya no hace falta continuar con todo el metodo.
          i = WebFiltering.SliceToolbar.webfiltering_iglobal+1;
        }

        WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_desde(elpadre, opcion, tolerancia_desde);
      }
    }
    else //sentido=="decreciente"
    {
      for(var i=0; i<WebFiltering.SliceToolbar.webfiltering_iglobal; i++)
      {
        var seguir=true;

        elpadre =
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_desde(tolerancia_desde, tolerancia_hasta, WebFiltering.
SliceToolbar.webfiltering_arrayKeyNodes[i], WebFiltering.SliceToolbar);

```

```

    if(elpadre.tagName=="BODY" || elpadre.tagName=="HTML")
    {
        // Este if se debe a que si llegamos al nodo mas alto con que se realice el mostrar
        // desde este nodo ya no hace falta continuar con todo el metodo.
        i = WebFiltering.SliceToolbar.webfiltering_iglobal+1;
    }

    WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_desde(elpadre,opcion,tolerancia_hasta);
}
}
}
else //opcion="normal"inverson";
{
    if(sentido=="creciente")
    {
        for(var i=0; i<WebFiltering.SliceToolbar.webfiltering_jglobal; i++)
        {
            var seguir=true;

            elpadre =
            WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_desde(tolerancia_desde,tolerancia_hasta,WebFiltering.
            SliceToolbar.webfiltering_arrayHiddenNodes[i], WebFiltering.SliceToolbar);

            if(elpadre.tagName!="BODY" && elpadre.tagName!="HTML")
            {
                elpadre =
                WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_hasta(tolerancia_desde,tolerancia_hasta,elpadre,opcion
                ,i,WebFiltering.SliceToolbar);
            }

            if(elpadre.tagName=="BODY" || elpadre.tagName=="HTML")
            {
                // Este if se debe a que si llegamos al nodo mas alto con que se realice el mostrar
                // desde este nodo ya no hace falta continuar con todo el metodo.
                i = WebFiltering.SliceToolbar.webfiltering_iglobal + 1;
            }

            WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_desde(elpadre,opcion,tolerancia_desde);
        }
    }
    else //sentido=="decreciente"
    {
        for(var i=0;i<WebFiltering.SliceToolbar.webfiltering_jglobal;i++)
        {
            var seguir=true;

            elpadre =
            WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_desde(tolerancia_desde,tolerancia_hasta,WebFiltering.
            SliceToolbar.webfiltering_arrayHiddenNodes[i], WebFiltering.SliceToolbar);

            if(elpadre.tagName=="BODY" || elpadre.tagName=="HTML")
            {
                // Este if se debe a que si llegamos al nodo mas alto con que se realice el mostrar
                // desde este nodo ya no hace falta continuar con todo el metodo.
                i = WebFiltering.SliceToolbar.webfiltering_iglobal + 1;
            }

            WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_desde(elpadre,opcion,tolerancia_hasta);
        }
    }
}

WebFiltering.SliceToolbar.webfiltering_tolerancia_antigua = WebFiltering.SliceToolbar.webfiltering_tolerancia;

// envio un request a background para guardar la tolerancia
chrome.extension.sendRequest({type: "set-storage-item", item: "tolerance", value:
WebFiltering.SliceToolbar.webfiltering_tolerancia}, function(response) {
    //alert("set-storage-item(tolerance): " + WebFiltering.SliceToolbar.webfiltering_tolerancia);
});
};

```

```

//se activa al pulsar show the full page
WebFiltering.SliceToolbar.webfiltering_verpagentera = function (recargar_pagina)
{
  /*
  * Se encarga de inicializar todas las variables.
  * Ademas, si mantener arbol es falso:
  * 1 - Vuelve todos los nodos ocultos en visibles.
  * 2 - Si hay algun nodo sombreado, lo desombrea.
  * Si mantener arbol es true, entonces recargar contenido.
  */

  WebFiltering.SliceToolbar.webfiltering_microformatosBuscados=new Array();
  WebFiltering.SliceToolbar.webfiltering_match_nodos = new Array ();
  WebFiltering.SliceToolbar.webfiltering_sliceAplicado = 0;
  WebFiltering.SliceToolbar.webfiltering_AplicadoSiguiete = 0;
  WebFiltering.SliceToolbar.webfiltering_AplicadoResaltar = 0;

  //WebFiltering.SliceToolbar.webfiltering_mantenerestructura = 0;
  //WebFiltering.SliceToolbar.webfiltering_mantenerarbol = 0;
  WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes = 0;
  WebFiltering.SliceToolbar.webfiltering_inverso = 0;
  WebFiltering.SliceToolbar.webfiltering_microformato = "";
  WebFiltering.SliceToolbar.webfiltering_texto = "";
  WebFiltering.SliceToolbar.webfiltering_texto_innerHTML_antiguo = "";
  WebFiltering.SliceToolbar.webfiltering_palabra_guardada = "";
  WebFiltering.SliceToolbar.webfiltering_posicion = -1;
  WebFiltering.SliceToolbar.webfiltering_tolerancia_antigua = 0;

  WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes = new Array();
  WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes = new Array();
  WebFiltering.SliceToolbar.webfiltering_arrayVisibleNodes = new Array();
  WebFiltering.SliceToolbar.webfiltering_iglobal = 0;
  WebFiltering.SliceToolbar.webfiltering_jglobal = 0;
  WebFiltering.SliceToolbar.webfiltering_vglobal = 0;

  // TODO: tema de multiples paginas, en vez de usar document.getElement.... habra
  // que usar accesos a localStorage a traves de las opciones
  //webfiltering_deshabilitar_botones(false,true); //Para habilitar todos los botones
  //if(document.getElementById("checkbox_multiple_webpage").checked) //Descheck multiples
  //{}
  // document.getElementById("checkbox_multiple_webpage").checked = false;
  //}

  if(recargar_pagina)
  {
    WebFiltering.SliceToolbar.webfiltering_contentDocument.location.reload();
  }

  WebFiltering.SliceToolbar.webfiltering_contentDocument = document;
};

WebFiltering.SliceToolbar.webfiltering_inicializar_opciones = function()
{
  // keepstructure
  chrome.extension.sendRequest({type: "get-storage-item", item: "keepstructure"}, function(response) {
    if (WebFiltering.SliceToolbar.debug == true) console.log("response: " + response);
    if (response != null)
    {
      if (response == "false") WebFiltering.SliceToolbar.webfiltering_mantenerestructura = 0;
      else if (response == "true") WebFiltering.SliceToolbar.webfiltering_mantenerestructura = 1;
    }
  });

  // formatiframes
  chrome.extension.sendRequest({type: "get-storage-item", item: "formatiframes"}, function(response) {
    if (WebFiltering.SliceToolbar.debug == true) console.log("response: " + response);
    if (response != null)
    {
      if (response == "false") WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes = 0;
      else if (response == "true") WebFiltering.SliceToolbar.webfiltering_formateartamanyoiframes = 1;
    }
  });
}

```



```

}
});

// keptree
chrome.extension.sendRequest({type: "get-storage-item", item: "keptree"}, function(response) {
  if (WebFiltering.SliceToolbar.debug == true) console.log("response: " + response);
  if (response != null)
  {
    if (response == "false") WebFiltering.SliceToolbar.webfiltering_mantenerarbol = 0;
    else if (response == "true") WebFiltering.SliceToolbar.webfiltering_mantenerarbol = 1;
  }
});

// inverse
chrome.extension.sendRequest({type: "get-storage-item", item: "inverse"}, function(response) {
  if (WebFiltering.SliceToolbar.debug == true) console.log("response: " + response);
  if (response != null)
  {
    if (response == "false") WebFiltering.SliceToolbar.webfiltering_inverso = 0;
    else if (response == "true") WebFiltering.SliceToolbar.webfiltering_inverso = 1;
  }
});

// continuous
chrome.extension.sendRequest({type: "get-storage-item", item: "continuous"}, function(response) {
  if (WebFiltering.SliceToolbar.debug == true) console.log("response: " + response);
  if (response != null)
  {
    if (response == "false") WebFiltering.SliceToolbar.webfiltering_slicing_continuo = 0;
    else if (response == "true") WebFiltering.SliceToolbar.webfiltering_slicing_continuo = 1;
  }
});

// followlinks
// TODO: for now the implementation of this feature is not required

// TODO: **no he encontrado lo de _Interfaz por ningun lado en el codigo original**
//WebFiltering.SliceToolbar.webfiltering_colapsarestructura = webfiltering_colapsarestructura_Interfaz;

WebFiltering.SliceToolbar.webfiltering_texto_particionar();

//Si hubiese aplicado algun coloreado entonces hay que quitarlo
if(WebFiltering.SliceToolbar.webfiltering_AplicadoSiguiente==1 && WebFiltering.SliceToolbar.webfiltering_posicion != -1)
{

WebFiltering.SliceToolbar.webfiltering_nodo_descolorear(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[WebFiltering.SliceToolbar.webfiltering_posicion]);
}

WebFiltering.SliceToolbar.webfiltering_AplicadoSiguiente = 0;
WebFiltering.SliceToolbar.webfiltering_posicion = -1;

};

//Tolerancias
WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_desde = function(desde,hasta,keyNode, THIS)
{
  var nodo_padre=keyNode;
  for(var i=1;i<desde+1;i++)
  {
    if(nodo_padre.parentNode.tagName == "BODY" || nodo_padre.parentNode.tagName == "HTML" ||
nodo_padre.getAttribute("HHH")==null)
    {
      if(desde<hasta)
      {
        // TODO: actualizar la tolerancia en el popup (si previamente se guardar en localStorage
        // en la siguiente carga del popup se vera correctamente, sino usar jquery para actualizarla)
        //document.getElementById("cuadrotolerancia").value =i;
        THIS.webfiltering_tolerancia_antigua=i-1;
        THIS.webfiltering_tolerancia=i;
      }
    }
  }
}

```

```

else
{
    THIS.webfiltering_tolerancia_antigua=i;
}
i=desde+1;
}

if(!WebFiltering.SliceToolbar.webfiltering_es_keynode(nodo_padre.parentNode))
    nodo_padre=nodo_padre.parentNode;
else
    return nodo_padre;
}
return nodo_padre;
}

WebFiltering.SliceToolbar.webfiltering_es_keynode = function(nodo)
{
    var elpadre_atributo=nodo.getAttribute("HHH");
    if(elpadre_atributo==null)
        return false;

    if(elpadre_atributo.substr(0,1)=="K")
        return true;
    else
        return false;
}

WebFiltering.SliceToolbar.webfiltering_mostrarcontolerancia_nodo_hasta =
function(desde,hasta,nodo_desde,opcion,posicionKeyNode,THIS)
{
    //elpadre.level_tolerance=0;
    nodo_padre=nodo_desde;
    for(var i=1;i<(hasta-desde)+1;i++)
    {
        if(nodo_padre.parentNode.tagName == "BODY" || nodo_padre.parentNode.tagName == "HTML" ||
nodo_padre.getAttribute("HHH")==null)
        {
            if(desde<hasta)
            {
                // TODO: actualizar la tolerancia en el popup (si previamente se guardar en localStorage
                // en la siguiente carga del popup se vera correctamente, sino usar jquery para actualizarla)
                //document.getElementById("cuadrotolerancia").value =i+desde;
                THIS.webfiltering_tolerancia_antigua=i+desde-1;
                THIS.webfiltering_tolerancia=i+desde;
            }
            else
            {
                THIS.webfiltering_tolerancia_antigua=i;
            }
            nodo_padre=nodo_padre.parentNode;
            break;
        }
        else
        {
            if(WebFiltering.SliceToolbar.webfiltering_es_keynode(nodo_padre.parentNode))
                return nodo_padre;

            nodo_padre=nodo_padre.parentNode;
        }
    }
}

WebFiltering.SliceToolbar.webfiltering_asociar_informacion_tolerancia(nodo_padre,desde+i,opcion,posicionKeyNode,THIS)
;
}
}
return nodo_padre;
}

WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_a_nodo = function(elpadre,opcion,tolerancia_desde,THIS)
{
    if(elpadre.nodeType==3 || elpadre.nodeType==8)
        return

```

```

var parar=false;
if(elpadre.tagName!=null && elpadre.tagName!="BODY" && elpadre.tagName!="HTML")
{
var nodo_atributo=elpadre.getAttribute("HHH");
if(nodo_atributo!=null)
{
var nodo_tipo=nodo_atributo.substr(0,1);
var nodo_posicion=nodo_atributo.substr(1);
var nodo_tipo_valor;

if(nodo_tipo=="K")
nodo_tipo_valor=0;
if(nodo_tipo=="H")
nodo_tipo_valor=1;
if(nodo_tipo=="V")
nodo_tipo_valor=2;

if(opcion=="normal")
{
if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion]==null ||
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level==null)
{
if(nodo_tipo=="H" || nodo_tipo=="V")
{
if(WebFiltering.SliceToolbar.webfiltering_colapsarestructura==1)
elpadre.style.display="";
else //this.webfiltering_mantenerestructura==1
elpadre.style.visibility="visible";
}
}
else
{
if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level>tolerancia_desde)
elpadre.style.visibility="visible";
else //if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level<tolerancia_desde+1)
parar=true;
}
}
}
else
{
if(nodo_tipo=="H" || (THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion]!=null &&
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso!=null &&
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso<tolerancia_desde))
{
parar=true;
}
else
{
if(WebFiltering.SliceToolbar.webfiltering_colapsarestructura==1)
elpadre.style.display="none";
else //this.webfiltering_mantenerestructura==1
elpadre.style.visibility="hidden";
}
}
/*
if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion]==null ||
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso==null ||
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso>tolerancia_desde)
{
if(nodo_tipo=="K" || nodo_tipo=="V")
{
{
if(THIS.webfiltering_colapsarestructura==1 )
elpadre.style.display="none";
}
else //this.webfiltering_mantenerestructura==1
elpadre.style.visibility="hidden";
}
}
}
else

```

```

{
    if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso>tolerancia_desde)
        elpadre.style.visibility="hidden";
    else //if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level<tolerancia_desde+1)
        parar=true;
    }
    */
}
}
}
return parar;
}

WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_a_nodo = function(elpadre,opcion,tolerancia_desde,THIS)
{
if(elpadre.nodeType==3 || elpadre.nodeType==8)
return

var parar=false;
if(elpadre.tagName!=null && elpadre.tagName!="BODY" && elpadre.tagName!="HTML")
{
var nodo_atributo=elpadre.getAttribute("HHH");
if(nodo_atributo!=null)
{
var nodo_tipo=nodo_atributo.substr(0,1);
var nodo_posicion=nodo_atributo.substr(1);
var nodo_tipo_valor;

if(nodo_tipo=="K")
    nodo_tipo_valor=0;
if(nodo_tipo=="H")
    nodo_tipo_valor=1;
if(nodo_tipo=="V")
    nodo_tipo_valor=2;

if(opcion=="normal")
    {
if(nodo_tipo!=null && nodo_tipo=="V")
    {
        //No hacer nada
    }
else
    {
if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion]==null ||
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level==null)
    {
if(nodo_tipo=="H" // || nodo_tipo=="V")
    {
if(THIS.webfiltering_colapsarestructura==1)
        elpadre.style.display="none";
else //this.webfiltering_mantenerestructura==1
        elpadre.style.visibility="hidden";
    }
else
    {
        parar="true";
    }
    }
else
    {
if (THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level>tolerancia_desde)
        elpadre.style.visibility="hidden";
else //if(THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level<tolerancia_desde+1)
        parar=true;
    }
    }
    }
else
    {
if(nodo_tipo=="H" || (THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion]!=null &&

```

```

THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso!=null &&
THIS.webfiltering_tolerance_level[nodo_tipo_valor][nodo_posicion].level_inverso<=tolerancia_desde))
    {
        parar=true;
    }
    else
    {
        if(THIS.webfiltering_colapsarestructura==1)
            elpadre.style.display="";
        else //this.webfiltering_mantenerestructura==1
            elpadre.style.visibility="visible";
    }
}
}
}
return parar;
}

```

WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_desde = **function** (elpadre,opcion,tolerancia_desde)

```

{
    var parar =
WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_a_nodo(elpadre,opcion,tolerancia_desde,WebFiltering.SliceToolbar);

```

```

if(parar)
    return;
//Hacer la llamada recursiva sobre sus hijos.
for (var i=0; i<elpadre.childNodes.length; i++)
{
    WebFiltering.SliceToolbar.webfiltering_aplicar_tolerancia_desde(elpadre.childNodes[i],opcion,tolerancia_desde)
}
};

```

WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_desde = **function** (elpadre,opcion,tolerancia_desde)

```

{
    var parar =
WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_a_nodo(elpadre,opcion,tolerancia_desde,WebFiltering.SliceToolbar);
if(parar)
    return;
//Hacer la llamada recursiva sobre sus hijos.
for (var i=0; i<elpadre.childNodes.length; i++)
{
    WebFiltering.SliceToolbar.webfiltering_quitar_tolerancia_desde(elpadre.childNodes[i],opcion,tolerancia_desde)
}
};

```

WebFiltering.SliceToolbar.webfiltering_formateariframes = **function** (pagina)

```

{
    /* RESUMEN: se encarga de convertir todos los nodos
    * con etiqueta IFRAME a etiquetas de tipo DIV.
    * Se realiza cogiendo el body de los iframe y lo
    * introduce en el innerHTML del DIV.
    * Mas tarde reemplaza el nodo IFRAME pro el nodo DIV. */

    var array = new Array();
    var altura,anchura;
    var padreiframe=0;
    var divNuevo;

    array=pagina.getElementsByTagName("IFRAME");

    //Quiero que el metodo solo se use cuando webfiltering_formateartamanyoiframes==1
    for(var i=0;i<array.length;i++)
    {
        altura = array[i].height; //obtenemos la altura del iframe
        anchura = array[i].width; //obtenemos la anchura la iframe
        padreiframe = array[i].parentNode; //guardamos el padre del iframe

        divNuevo = pagina.createElement('div');//crea un div
    }
}

```

```

divNuevo.id='nuevo';
divNuevo.innerHTML = array[i].getElementsByTagName("body").innerHTML; //le anyade al nuevo div el contenido del
iframe

divNuevo.height = altura;
divNuevo.width = anchura;
divNuevo.overflow="auto";

alert("Formateo de iframe: altura=" + altura + " anchura=" + anchura);

//Asignamos los mismos atributos excepto el src que es la ruta
var atributos=array[i].attributes;
var j;
for(j=0;j<atributos.length;j++)
{
//if(atributos[j].name.toLowerCase()!="src"){
divNuevo.setAttribute(atributos[j].name,atributos[j].value);
}

var posicion=-1;
var posicion2=-1;

var scripts=divNuevo.getElementsByTagName("script");
for(j=0;j<scripts.length;j++)
{
scripts[j].innerHTML="";
}
padreiframe.replaceChild(divNuevo,array[i]);//reemplaza el iframe por el div
}
};

```

WebFiltering.SliceToolbar.webfiltering_nokeeptree = **function** (pagina)

```

{
/* RESUMEN: Se encarga de romper la estructura en arbol de la
* pagina html. Dependiendo de webfiltering_inverso hace:
* 1.- (Inverso desactivo). Cuelgan del nodo raiz body solo
* los nodos que coinciden con la busqueda/s.
* 2.- (Inverso activo). Se descuelgan los nodos que coinciden
* con la busqueda de sus respectivos padres.
* Si no se ha aplicado slicing, la busqueda de nodos es sobre
* todo el codigo html. Mientras que si ya se ha realizado un
* slicing, la busqueda de coincidencias es sobre arraymuestra.
*/
if(WebFiltering.SliceToolbar.webfiltering_sliceAplicado==1)
{
WebFiltering.SliceToolbar.webfiltering_nokeeptree_sobre_arraymuestra(pagina);
}
else
{
WebFiltering.SliceToolbar.webfiltering_nokeeptree_sobre_html(pagina);
}

if(WebFiltering.SliceToolbar.webfiltering_iglobal==0)
{
alert("La palabra buscada no se encuentra");
if(WebFiltering.SliceToolbar.webfiltering_slicing_continuo==false)
{
WebFiltering.SliceToolbar.webfiltering_verpagentera(true);
//content.window.location.reload();
}
}
};

```

WebFiltering.SliceToolbar.webfiltering_nokeeptree_sobre_html = **function** (pagina)

```

{
// RESUMEN:
// Se encarga de realizar el metodo keeptree sobre html, es decir,
// todavia no se ha realizado un slicing sobre esta pagina.
// Este metodo cambia de cualquier otro slicing ya que
// aquellos nodos que 'no' coinciden con los criterios de busqueda
// no son ocultados sino que rompen el enlace en forma de arbol

```

```

// respecto a su nodo padre (dando un efecto de 'amontonamiento')

var vectorDeFrames=pagina.getElementsByTagName("FRAME");

if(vectorDeFrames.length > 0)
{
    for(var w=0; w < vectorDeFrames.length; w++)
    {
        WebFiltering.SliceToolbar.webfiltering_nokeeptree(vectorDeFrames[w].contentDocument);
    }
}
else
{
    WebFiltering.SliceToolbar.webfiltering_ejecucionfunc_sinocultar(pagina);

    if(WebFiltering.SliceToolbar.webfiltering_inverso == 1)
    {
        for(i=0;i<WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes.length;i++)
        {
            WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[i].parentNode.removeChild(WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[i]);
        }
    }
    else
    {
        var i;

        // Sustituir los nodoHuerfanos en nodosCompletos.
        for(i=0;i<WebFiltering.SliceToolbar.webfiltering_iglobal;i++)
        {
            if(WebFiltering.SliceToolbar.webfiltering_es_un_nodo_huerfano(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]))
            {
                WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]=WebFiltering.SliceToolbar.webfiltering_convertir_nodoHuerfano_e_n_nodoPadre(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]);
            }
        }

        //Eliminar todos los hijos del body
        for(i=pagina.body.childNodes.length-1;i>-1;i--)
        {
            pagina.body.removeChild(pagina.body.childNodes[i]);
        }

        //Que los keynode cuelquen todos del body
        for(i=0;i<WebFiltering.SliceToolbar.webfiltering_iglobal;i++)
        {
            pagina.body.appendChild(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]);
        }
    }
};

WebFiltering.SliceToolbar.webfiltering_nokeeptree_sobre_arraymuestra = function (pagina)
{
    // RESUMEN:
    // Se encarga de realizar el metodo keeptree sobre arrayKeynode.
    // Este metodo cambia de cualquier otro slicing ya que
    // aquellos nodos que 'no' coinciden con los criterios de busqueda
    // no son ocultados sino que rompen el enlace en forma de arbol
    // respecto a su nodo padre (dando un efecto de 'amontonamiento')

    var i = 0;
    var ArrayAuxiliar = new Array();
    var ArrayAuxiliarContador = 0;
    var resultado;
    for(i=0; i<WebFiltering.SliceToolbar.webfiltering_iglobal; i++)
    {

```

```

resultado =
WebFiltering.SliceToolbar.webfiltering_contiene_coincidencia(WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]);

if(resultado == -1)
{
    //Los voy ya introduciendo en array_oculto
    WebFiltering.SliceToolbar.webfiltering_arrayHiddenNodes[WebFiltering.SliceToolbar.webfiltering_iglobal] =
WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i];
    WebFiltering.SliceToolbar.webfiltering_iglobal++;
    if(WebFiltering.SliceToolbar.webfiltering_inverso == 0)
    {

WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i].parentNode.removeChild(WebFiltering.SliceToolbar.webfiltering_ar
rayKeyNodes[i]);
    }
}
else
{
    if(WebFiltering.SliceToolbar.webfiltering_inverso == 1)
    {

WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[WebFiltering.SliceToolbar.webfiltering_iglobal]=WebFiltering.SliceTo
olbar.webfiltering_arrayKeyNodes[i];
    WebFiltering.SliceToolbar.webfiltering_iglobal++;

WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i].parentNode.removeChild(WebFiltering.SliceToolbar.webfiltering_ar
rayKeyNodes[i]);
    }
}
else
{
    ArrayAuxiliar[ArrayAuxiliarContador] = WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i];
    ArrayAuxiliarContador++;
}
}
}

if(ArrayAuxiliarContador>0 && WebFiltering.SliceToolbar.webfiltering_inverso != 1)
{
    //alert("Entra teniendo el iglobal_"+this.webfiltering_iglobal+" y el arrayauxiliar:"+ArrayAuxiliarContador);
    WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes = new Array();
    WebFiltering.SliceToolbar.webfiltering_iglobal = 0;
    for(i=0;i<ArrayAuxiliarContador;i++)
    {
        WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i]=ArrayAuxiliar[i];
        WebFiltering.SliceToolbar.webfiltering_iglobal++;
    }
    return true;
}

if(WebFiltering.SliceToolbar.webfiltering_iglobal > 0)
{
    return true;
}
else
{
    return false;
}

if(WebFiltering.SliceToolbar.webfiltering_inverso == 1)
{
    for(i=0; i<WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes.length; i++)
    {

WebFiltering.SliceToolbar.webfiltering_arrayKeyNodes[i].parentNode.removeChild(WebFiltering.SliceToolbar.webfiltering_ar
rayKeyNodes[i]);
    }
}
};

/* AUXILIAR FUNCTIONS */
WebFiltering.SliceToolbar.webfiltering_es_un_nodo_huerfano = function (nodo)

```



```

{
if(nodo==null || nodo.tagName==null)
{
return false;
}

switch(nodo.tagName.toLowerCase())
{
case 'li':
case 'tr':
case 'td':
case 'tbody':
case 'dd':
case 'dt':
return true;
break;
default:
return false;
break;
}

return false;
};

```

WebFiltering.SliceToolbar.webfiltering_asociar_informacion_tolerancia = **function**
(nodo_Padre,nivel,opcion,posicionKeyNode,THIS)

```

{
var elpadre_atributo=nodo_padre.getAttribute("HHH");
var elpadre_position=parseInt(elpadre_atributo.substr(1));
var elpadre_fila;

if(elpadre_atributo.indexOf("K")==0) //Es un keynode
{
elpadre_fila=0;
}
else
{
if(elpadre_atributo.indexOf("H")==0) //Es un hidden
elpadre_fila=1;
else //Es un visible
elpadre_fila=2;
}

//Inicializar si no lo esta;
if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position]==null)
{
THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position]=new webfiltering_tolerance_level_class();
}

if(opcion=="normal")
{
//Actualizar nivel de tolerancia
if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level==null ||
THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level>nivel)
{
THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level=nivel;
}

//Anyadir al historico
if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history==null)
{
THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count=0;
THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history = new Array();
}

//Habria que comprobar que no existe ya.
var anyadir=true;
var valor=posicionKeyNode.toString()+" "+nivel.toString();

for(var i=0;i<THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count;i++)

```

```

    {
        if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history[i]==valor)
        {
            anaydir=false;
        }
    }

    if(anyadir)
    {

THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history[THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count]=valor;
        THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count++;
    }
}
else //opcion="inverso";
{
    //Actualizar nivel de tolerancia
    if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level_inverso==null || THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level_inverso>nivel)
    {
        THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].level_inverso=nivel;
    }

    //Anyadir al historico
    if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history_inverso==null)
    {
        THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count=0;
        THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history_inverso = new Array();
    }

    //Habria que comprobar que no existe ya.
    var anyadir=true;
    var valor=posicionKeyNode.toString()+" "+nivel.toString();

    for(var i=0;i<THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count;i++)
    {
        if(THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history_inverso[i]==valor)
        {
            anaydir=false;
        }
    }

    if(anyadir)
    {

THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].history_inverso[THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count]=valor;
        THIS.webfiltering_tolerance_level[elpadre_fila][elpadre_position].count++;
    }
}
};

WebFiltering.SliceToolbar.webfiltering_quitar_VisibleNode_asociado_a_KeyNode = function(nodo,i,ocultar,THIS)
{
    var arrayVisibleNodes_nuevos=new Array();
    var arrayVisibleNodes_nuevos_cuantos=0;
    var arrayVisibleNodes_nuevos_tolerance_level=new Array();

    for(var j=0; j< THIS.webfiltering_vglobal;j++)
    {
        // hijos del visible node
        var hijos=THIS.webfiltering_tolerance_level[2][j].hijos_keynode;

        if( hijos.indexOf(";"+i.toString()+";")!=-1) //Significa que el visibleNode contiene el KeyNode
        {
            while(hijos.indexOf(";"+i.toString()+";")!=-1) //Quitar la informacion en VisibleNode que apunta a KeyNode
            {
                hijos=hijos.replace(";"+i.toString()+";","");
            }
        }
    }
}

```

```

while(hijos.indexOf(",")!==-1) //Quitamos los ;
{
    hijos=hijos.replace(",","");
}

while(hijos.indexOf(" ")!==-1) //Quitamos los espacios
{
    hijos=hijos.replace(" ","");
}

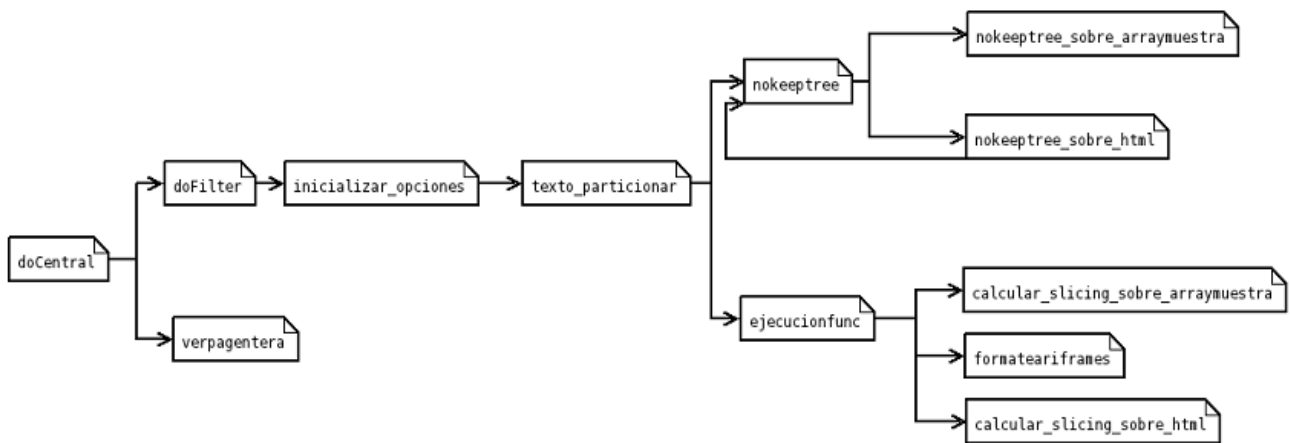
if(hijos.length==0) //Es que no tiene mas keynode
{
    //Habria que ocultarlo (sin agregarlo a hiddenNodes).
    if(ocultar)
    {
        if (THIS.webfiltering_mantenerestructura==1)
            THIS.webfiltering_arrayVisibleNodes[j].style.display="none";
        else
            THIS.webfiltering_arrayVisibleNodes[j].style.visibility="hidden";
    }
}
else //Tiene mas keynode asociados.
{
    var hijos_nuevo=THIS.webfiltering_tolerance_level[2][j].hijos_keynode;
    while(hijos_nuevo.indexOf(";"+i.toString()+";")!==-1) //Quitar la informacion en VisibleNode que apunta a KeyNode
    {
        hijos_nuevo=hijos_nuevo.replace(";"+i.toString()+";",";");
    }
    THIS.webfiltering_tolerance_level[2][j].hijos_keynode=hijos_nuevo;
    arrayVisibleNodes_nuevos[arrayVisibleNodes_nuevos_cuantos]=THIS.webfiltering_arrayVisibleNodes[j];
arrayVisibleNodes_nuevos_tolerance_level[arrayVisibleNodes_nuevos_cuantos]=THIS.webfiltering_tolerance_level[2][j];
    arrayVisibleNodes_nuevos_cuantos++;
}
}
else //El visibleNode no contiene el keyNode, entonces hay simplemente agregarlo.
{
    arrayVisibleNodes_nuevos[arrayVisibleNodes_nuevos_cuantos]=THIS.webfiltering_arrayVisibleNodes[j];
arrayVisibleNodes_nuevos_tolerance_level[arrayVisibleNodes_nuevos_cuantos]=THIS.webfiltering_tolerance_level[2][j];
    arrayVisibleNodes_nuevos_cuantos++;
}
}
THIS.webfiltering_arrayVisibleNodes=arrayVisibleNodes_nuevos;
THIS.webfiltering_vglobal=arrayVisibleNodes_nuevos_cuantos;
THIS.webfiltering_tolerance_level[2]=arrayVisibleNodes_nuevos_tolerance_level;
};

/* CLASSES */

webfiltering_textos_clase = function()
{
    // CONSTRUCTOR de la clase webfiltering_textos_clase
    // que sirve para particionar la frase en el cuadro_blanco
    this.palabra = "";
    this.operator = "AND";
    this.resultado = false;
    this.resultado_atributos = false;
    this.resultado_innerHTML = false;
};

webfiltering_tolerance_level_class = function()
{
    this.level;
    this.level_inverso;
    this.history;
    this.history_inverso;
    this.count;
    this.hijos_keynode;
};

```



Las funciones más importantes son:

doFilter

Se ejecuta tras apretar el botón Filtrar y dependiendo de si la opción 'mantener árbol' está o no seleccionada se ejecutará la función 'nokeepree' (no mantiene el árbol de nodos), o 'ejecucionfunc' (mantiene el árbol de nodos). Posteriormente se guarda la palabra clave utilizada.

inicializar_opciones

Inicializa los parametros (keepree, keepstructure, inverse...) y les asocia el respectivo valor a las variables.

texto_particionar

Método que se encarga de coger la frase que hay en el cuadro de texto y particionarlo siguiente reglas AND, OR y literal.

nokeepree

Se encarga de romper la estructura en árbol de la página html. Si no se ha aplicado slicing, la búsqueda de nodos es sobre todo el código html. Mientras que si ya se ha realizado un slicing, la búsqueda de coincidencias es sobre arraymuestra.

nokeepree_sobre_arraymuestra

Se encarga de realizar el método keepree sobre arrayKeynode. Este método cambia de cualquier otro slicing ya que aquellos nodos que 'no' coinciden con los criterios de búsqueda no son ocultados sino que rompen el enlace en forma de árbol respecto a su nodo padre (dando un efecto de 'amontonamiento')

nokeepree_sobre_html

Se encarga de realizar el método keepree sobre html, es decir, todavía no se ha realizado un slicing sobre esta página. Este método cambia de cualquier otro slicing ya que aquellos nodos que 'no' coinciden con los criterios de búsqueda no son ocultados sino que rompen el enlace en forma de árbol respecto a su nodo padre (dando un efecto de 'amontonamiento')

ejecucionfunc

Realiza un filtro slice dependiendo de si ya se ha ejecutado o no. Si no se ha ejecutado se realizará entre el código html pero si ya se ha ejecutado se realizará solo entre los nodos del vector 'arraymuestra'. A ambos métodos se les pasa un parámetro a 'true' que oculta los nodos no coincidentes.

calcular_slicing_sobre_arraymuestra

Se encarga de buscar entre los nodos del 'arraymuestra' aquellos que coinciden con los criterios de búsqueda. Si coinciden se guardan en el 'arraymuestra' de nuevo o sino se guardan al final de 'arrayocultos'.

calcular slicing sobre html

Este método se encarga en encontrar sobre los hijos del parametro 'pagina' los hijos que coinciden con la búsqueda. El parametro ocultar es para ocultar o no los parametros que coinciden con la búsqueda. Este metodo es utilizado desde el boton filtrar (ocultar=true) pero tambien desde el boton siguiente, anterior y resaltar (ocultar=false). Hay que tener en cuenta que si la opcion slicing inverso se encuentra activa, se guarda en arraymuestra los nodos no coinciden mientras si esta desactivada es al revés.

formateariframes

Convierte todos los nodos con etiqueta IFRAME a etiquetas de tipo DIV. Se realiza cogiendo el body de los iframe y lo introduce en el innerHTML del DIV. Más tarde reemplaza el nodo IFRAME pro el nodo DIV.

Página de Opciones

La página de opciones como no podría ser de otra forma también está dividida en 2 ficheros: por un lado un fichero options.html que contendrá la parte de la interfaz que interactuará con el usuario para que este cambie las propiedades de nuestra extensión, y por otro lado constará de un fichero que llevará todo el código JavaScript necesario para acceder a operaciones de guardado de opciones mediante el mencionado 'localStorage' etc.

El fichero *options.html* queda así:

```
<!DOCTYPE html>
<html id="web-filtering-options">

<head>
<meta charset="utf-8">
<title>Web Filtering Options</title>
<link href="css/options.css" rel="stylesheet">
<script src="js/jquery.js"></script>
<script src="js/options.js"></script>
</head>

<body>

<h1>Web Filtering Toolbar</h1>

<div id="options">
<form>

  <h2>Options</h2>
  <hr />

  <table>
  <tr>

    <td>
    <input type="checkbox" id="keepstructure" checked />
    <label for="keepstructure">Keep Structure</label>
    </td>
    <td>
    <p>It is used to specify if the slice should keep the structure of the original
    web page. If this option is marked, the content of the slice keeps the same
    position as in the original page. If it is not marked, the elements of the
    slice are located at the top of the sliced page.</p>
    </td>

  </tr>
  <tr>

    <td>
    <input type="checkbox" id="formatiframes" checked />
    <label for="formatiframes">Format Size iFrames</label>
    </td>
    <td>
    <p>When this option is marked the iframes of the original web page are formatted
    in the slice. Otherwise, the original size of the iframes is kept in the slice.</p>
    </td>

  </tr>
  <tr>
  <td>
```

```

in the slice. If it is desactivated, the ancestors are deleted
in the final slice.</p>
</td>

</tr>
</table>

<hr />

<div id="notification">
  <span class="icon"></span>
</div>

<br />
<input type="submit" value="Save" id="save">
<input type="submit" value="Cancel" id="cancel">
</form>
</div>

</body>
</html>

```

Y su fichero de código `options.js` queda así:

```

var WebFiltering = WebFiltering || {};

WebFiltering.Options = WebFiltering.Options || {};

$(document).ready(function()
{
  // check first time we try to load variables from localStorage
  if (window.localStorage.getItem("keepstructure") == null)
  {
    WebFiltering.Options.setDefaults();
  }
  WebFiltering.Options.load();
  $("#cancel").click(WebFiltering.Options.cancel);
  $("#save").click(WebFiltering.Options.save);
});

// Handles the options being cancelled
WebFiltering.Options.cancel = function()
{
  window.close();
};

// Initialize with defaults
WebFiltering.Options.setDefaults = function()
{
  window.localStorage.setItem("keepstructure", true);
  window.localStorage.setItem("formatiframes", false);
  window.localStorage.setItem("keeptree", true);
};

// Loads the options
WebFiltering.Options.load = function()
{
  //alert("to restore keepstructure= " + window.localStorage.getItem("keepstructure"));
  if (window.localStorage.getItem("keepstructure") == "false")
  {
    $("#keepstructure").get(0).checked = false;
  }
}

```

```

//alert("to restore formatiframes= " + window.localStorage.getItem("formatiframes"));
if (window.localStorage.getItem("formatiframes") == "false")
{
    $("#formatiframes").get(0).checked = false;
}

//alert("to restore keptree= " + window.localStorage.getItem("keptree"));
if (window.localStorage.getItem("keptree") == "false")
{
    $("#keptree").get(0).checked = false;
}
};

// Handles the options being saved
WebFiltering.Options.save = function()
{
    //alert("to save keepstructure= " + $("#keepstructure").get(0).checked);
    window.localStorage.setItem("keepstructure", $("#keepstructure").get(0).checked);

    //alert("to save formatiframes= " + $("#formatiframes").get(0).checked);
    window.localStorage.setItem("formatiframes", $("#formatiframes").get(0).checked);

    //alert("to save keptree= " + $("#keptree").get(0).checked);
    window.localStorage.setItem("keptree", $("#keptree").get(0).checked);

    // shows a notification message
    alert("Confirm changes to continue");
    WebFiltering.Options.showSimpleNotification("Options saved successfully");
};

// Shows the notification
WebFiltering.Options.showNotification = function()
{
    $("#notification").show(); // .fadeIn(500) would be better but doesn't work
};

// Clears a notification
WebFiltering.Options.clearNotification = function()
{
    var notification = $("#notification");
    notification.hide();
    notification.children().remove();
};

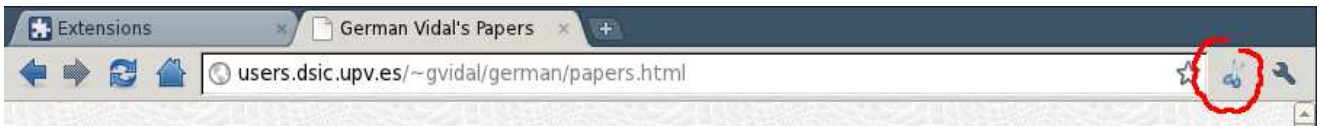
// Shows a simple notification
WebFiltering.Options.showSimpleNotification = function(message)
{
    WebFiltering.Options.clearNotification();
    $("<p></p>").html(message).appendTo($("#notification"));
    WebFiltering.Options.showNotification();
};

```

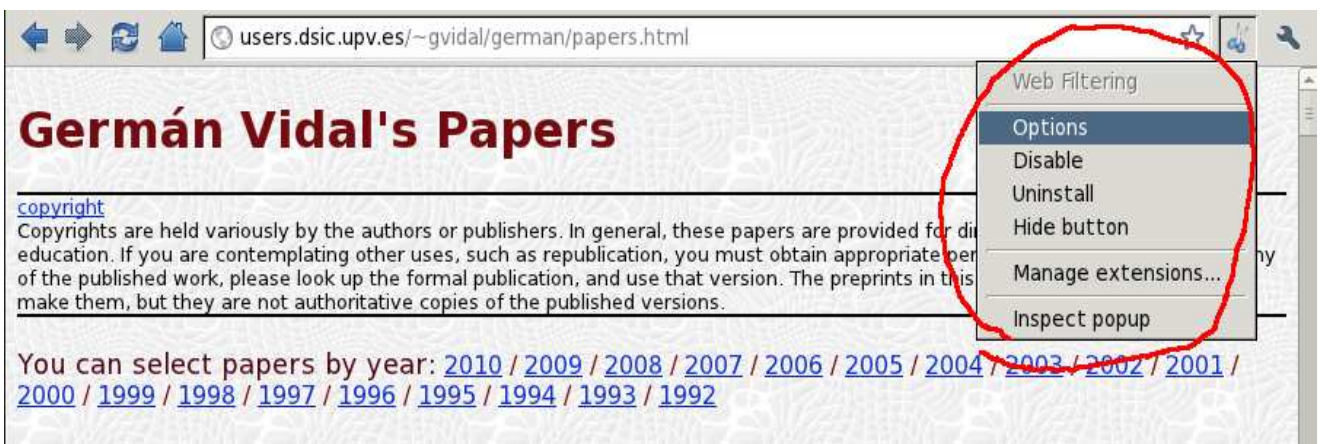
5. GUIA DE USO DE LA EXTENSIÓN

A continuación se detalla brevemente cómo se debe usar la extensión mediante ejemplos.

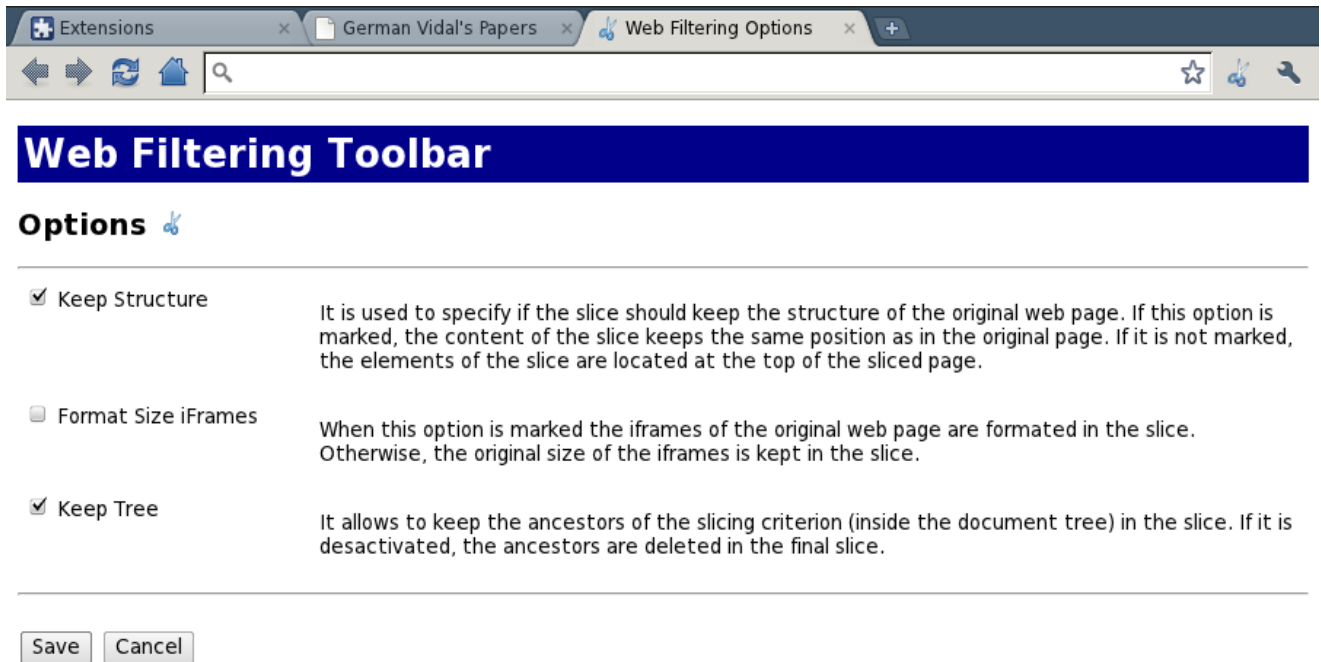
Lo primero y una vez instalada y habilitada la extensión para su uso, será inspeccionar la pantalla de opciones para ver qué propiedades están establecidas por defecto y modificarlas a nuestro gusto si lo viéramos necesario. Para ello deberemos dirigirnos al icono de nuestra extensión, situado en la esquina superior derecha del navegador (junto al icono de propiedades de Google Chrome).



Una vez localizado nuestro icono, con el botón derecho del ratón hacemos clic para mostrar el desplegable:



A continuación seleccionamos 'Options' que nos llevará hasta la siguiente pantalla de opciones, donde seleccionaremos nuestras preferencias de filtrado.

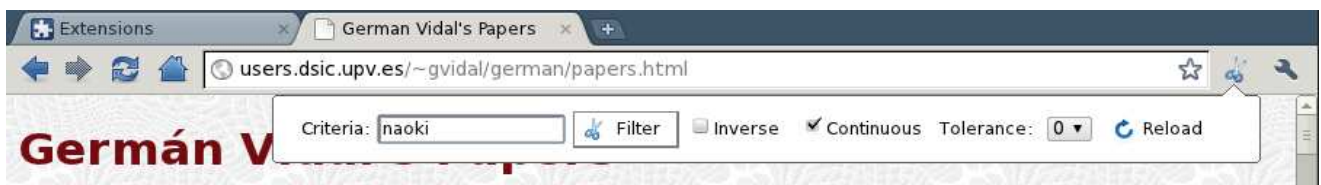


Como se puede ver en la pantalla de opciones, existen 3 propiedades a seleccionar:

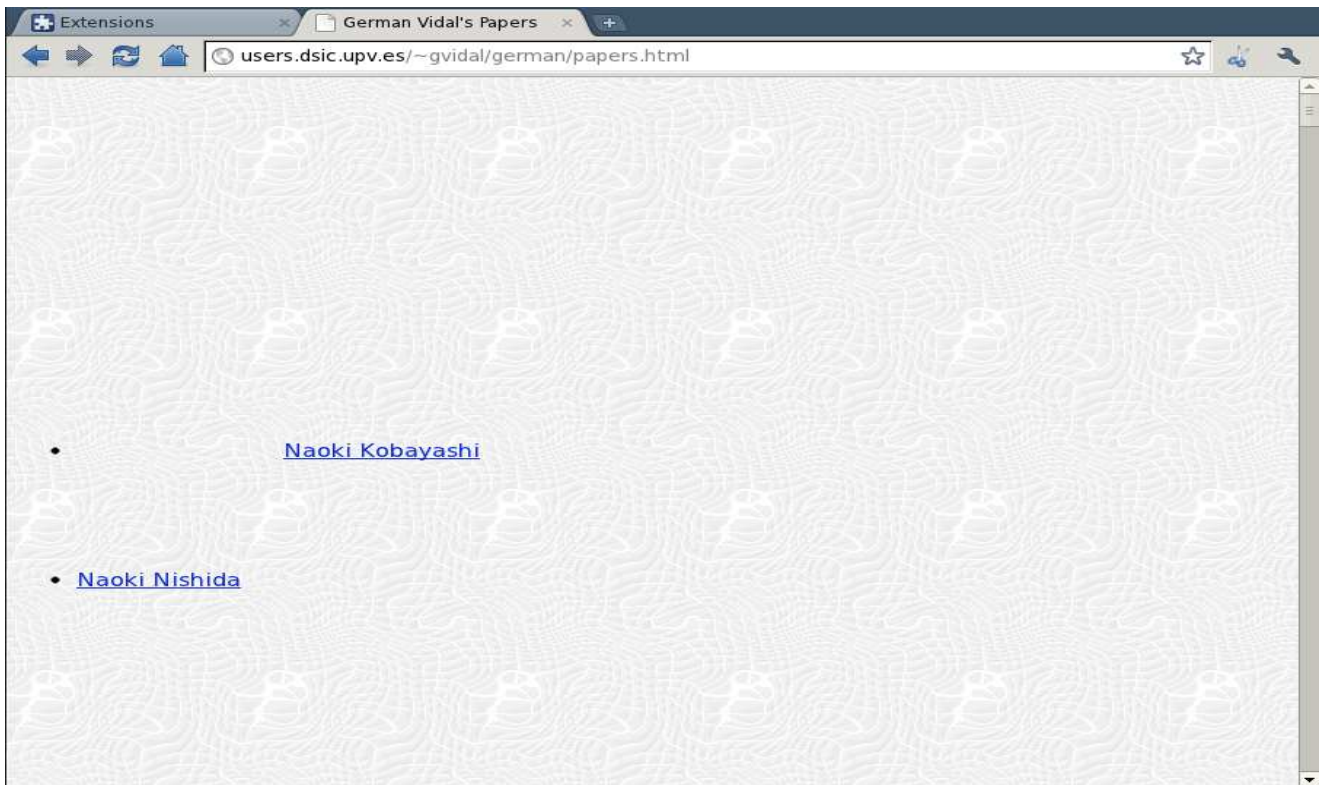
- **Keep Structure** (mantener estructura). Se refiere a si el filtro debe mantener la estructura original de la página web visitada, de manera que cada elemento permanezca en su lugar original tras el filtrado o bien sean visibles al principio de esta.
- **Format Size iFrames** (formatear el tamaño de los iframes). Como su nombre indica formateará el tamaño de los iframes que encuentre en la página web visitada cuando se encuentre activada esta opción.
- **Keep Tree** (mantener árbol). Hace referencia a si se deben mantener o borrar los padres de los nodos (dentro del árbol DOM de la página) que coincidan con el criterio de filtrado.

A continuación, se muestra un ejemplo de uso del filtro, en el que pretendemos ver los cambios que sufre la página web visitada ante un criterio de filtrado y diferentes opciones.

Para ello, visitamos una página de muestra en la que introducimos la palabra 'naoki' como criterio de filtrado. Acto seguido le damos al botón 'Filter' para realizar la acción de filtrado según las opciones por defecto:



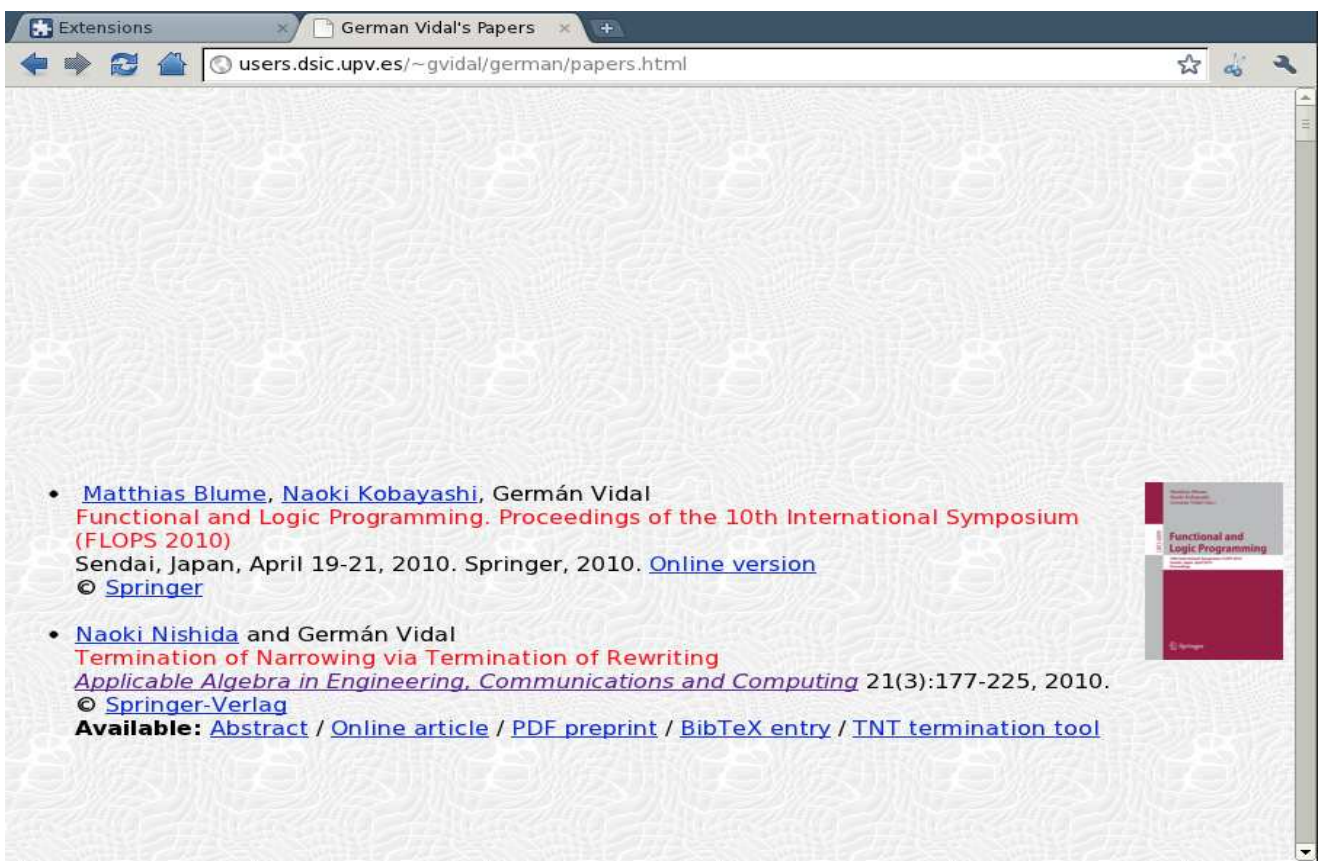
Y el resultado obtenido, con tolerancia igual a 0 es el siguiente:

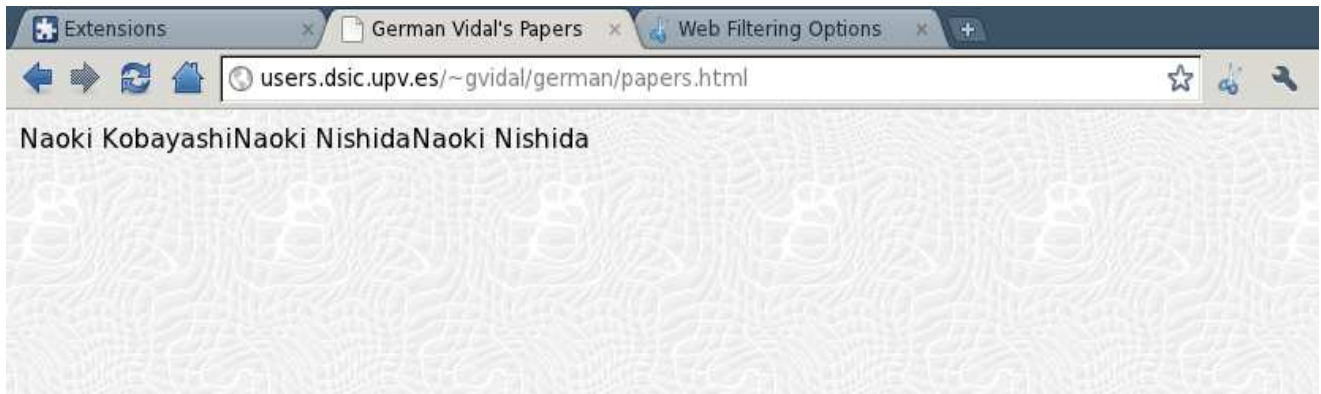


Como se puede apreciar, la herramienta ha buscado cada porción que contenga el criterio de búsqueda 'Naoki' y posteriormente ha filtrado cada porción anterior de acuerdo a las opciones que tenemos configuradas, por lo que ha mantenido la estructura visual de los elementos en la página.

Ahora repetiremos el ejemplo anterior pero esta vez en vez de mantener las opciones por defecto acudiremos a la ventana de opciones y desmarcaremos 'KeepStructure' y 'KeepTree'. Como se ha comentado deberá alterar la estructura visual y mostrarnos todos los elementos al principio de la página resultante. Veamos el efecto.

El resultado obtenido con tolerancia 0, sin mantener la estructura de los elementos ni el árbol DOM de la página es el siguiente.





En efecto, el resultado demuestra el amontonamiento de los elementos resultantes al principio de la página.

Ahora volvamos al primer ejemplo, y repitamos el filtrado pero esta vez con tolerancia 2 a ver cual es el resultado.



Y el resultado ante el mismo criterio de búsqueda, con las opciones por defecto, y con tolerancia igual a 2 queda de la siguiente forma.

6. CONCLUSIONES

6.1. CONCLUSIONES DEL PROYECTO

Durante la realización de este proyecto se intentó cubrir la mayor parte de funcionalidades existentes en el plugin original, pero aun así quedaron algunas facetas por terminar y pulir. La que yo considero más importante es la de combatir el problema de la ausencia de llamadas síncronas.

Este hecho hace que sea inestable el trabajar con opciones de usuario, ya que éstas se deben almacenar y recuperar mediante llamadas de *callback* asíncronas, lo que supone que puede ocurrir que se procese alguna propiedad en una estructura de control de decisión condicional (como por ejemplo un *if-then-else*) sin que se tenga el último valor guardado para esa propiedad, pudiendo elegirse así una sentencia errónea. Todo esto queda reflejado en el siguiente ticket de su Bugtracker^[11] y se espera que pueda cambiar en un futuro no muy lejano.

La solución abordada por el momento se trata de mostrar una ventana emergente bloqueante (un alert) justo antes de iniciar el filtrado de la página en curso. De esta forma se detiene la secuencia de eventos hasta que se confirme la continuación desde la ventana emergente, y de este modo se obtienen todos los valores correctos para las propiedades del objeto WebFiltering, como pueden ser el nivel de tolerancia y el resto de opciones.

Otra característica a reseñar sería que esta extensión se ha desarrollado para un navegador que todavía está en su fase más prematura de desarrollo, por lo que cualquier nueva versión del navegador Chrome podría afectar al funcionamiento de la extensión. Si bien podría cubrir la necesidad de llamadas síncronas como antes se comentaba, podría darse el caso de que alguna faceta de nuestra extensión dejara de funcionar y tuviera que ser reimplementada.

6.2. TRABAJO FUTURO Y AMPLIACIONES

Como posibles futuros trabajos para la continuación de esta extensión para Chrome se podrían proponer varias ampliaciones:

Idiomas

Se podría traducir cada cadena de texto e ir añadiendo estas éstas a modo de idiomas a la extensión. Para ello se debería añadir una línea al archivo *manifest.json* con el siguiente contenido:

```
"default_locale": "en"
```

Esta directiva especificaría cual es el subdirectorio dentro de *_locales* que contendría las cadenas de texto por defecto usadas para la extensión. Más detalles se pueden encontrar en la siguiente url:

<http://code.google.com/chrome/extensions/i18n.html>

Luego faltaría ir cambiando en el código aquellas cadenas de texto candidatas a su traducción, como podrían ser los mensajes de alert, pantallas de opciones, popup, etc.

Actualizaciones

Chrome provee facilidades para tener un sistema de actualizaciones para extensiones. Si lo que se desea es tener esta característica habría que añadir una línea al archivo *manifest.json* con el siguiente contenido:

```
"update_url": "http://path/to/updateInfo.xml"
```

Con la siguiente directiva se pretende indicar cual es la url al fichero xml para las actualizaciones automáticas. Para más información: <http://code.google.com/chrome/extensions/autoupdate.html>

Por otro lado, si se pretende publicar la extensión en el tablón oficial de extensiones para Chrome (Chrome Developer Dashboard) la directiva anterior se debe ignorar, y recurrir en ese caso al *dashboard* antes mencionado: <https://chrome.google.com/webstore/developer/dashboard>

7. BIBLIOGRAFIA

- [1] Mozilla Firefox – Mozilla Europe, Mozilla Foundation. 650 Castro Street Suite 300 CA, 94041-2021 USA (Octubre 2010)
<http://www.mozilla.com/>
http://en.wikipedia.org/wiki/Mozilla_Firefox
- [2] Google Chrome – Google Chrome Web Browser. Google Inc. 1600 Amphitheatre Parkway Mountain View CA 94043 USA (Octubre 2010)
<http://www.google.com/chrome?hl=es>
http://en.wikipedia.org/Google_Chrome
- [3] Subversion – Apache Software Foundation. Apache Team Subversion © “Enterprise-class centralized version control for the masses” (Octubre 2010)
<http://subversion.apache.org>
http://en.wikipedia.org/Apache_Subversion

- [4] Git – Fast Version Control System (Enero 2011)
<http://git-scm.com/>
[http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software))

- [5] DOM – W3C Document Object Model (Octubre 2010)
<http://www.w3.org/DOM/>
http://en.wikipedia.org/wiki/Document_Object_Model

- [6] JavaScript – “THE scripting language of the web” (Noviembre 2010)
http://www.w3schools.com/js/js_intro.asp
<http://en.wikipedia.org/wiki/JavaScript>
<https://developer.mozilla.org/en/javascript>

- [7] JSON – JavaScript Object Notation (Enero 2011)
<http://www.json.org/>

- [8] JQuery – Cross-browser JavaScript Library (Enero 2011)
<http://jquery.com/>
http://docs.jquery.com/Main_Page
<http://en.wikipedia.org/wiki/JQuery>

- [9] Firefox Addons – Mozilla Europe, Mozilla Foundation (Diciembre 2010)
<https://addons.mozilla.org/en-US/firefox/developers>
<https://addons.mozilla.org/en-US/firefox/>

- [10] Chrome Extensions (Diciembre 2010 - Junio 2011)
<http://code.google.com/chrome/extensions/index.html>
<https://chrome.google.com/extensions/>

- [11] Bugtracker for Chromium Project – An open-source web browser project to help move the web forward (Diciembre 2010 - Junio 2011)
<http://code.google.com/p/chromium/issues/detail?id=54257>

- [12] HTML5 – A vocabulary and associated APIs for HTML and XHTML. (Diciembre 2010)
<http://www.w3.org/TR/html5/>
<http://dev.w3.org/html5/spec/Overview.html>
http://en.wikipedia.org/wiki/HTML_5

- [13] Webfiltering Toolbar – The Web Filtering Project. A Toolbar to Filter Webpages (Diciembre 2010)
<http://www.dsic.upv.es/~jsilva/webfiltering>
<https://addons.mozilla.org/es-ES/firefox/addon/web-filtering-toolbar/>

- [14] IEEE 830-1984 – IEEE Guide to Software Requeriments Specifications. IEEE Standards Board. 354 Eas 47th Street. New York, NY 10017, USA. 1998 (Mayo 2011)
<http://standards.ieee.org/findstds/standard/830-1984.html>
http://ieeexplore.ieee.org/search/freeschabstract.jsp?tp=&arnumber=278253&queryText%3Dspecifications+830%26openedRefinements%3D*%26searchField%3DSearch+All