

▪ Proyecto Fin de Grado ▪
Ingeniería de Software

BLOQUEO DE PUBLICIDAD EN LA WEB: Enfoque técnico
en la búsqueda de un compromiso entre anunciantes y
lectores

Sergio Rivas Ramos

Junio 2015

Dirigido por: Oscar Díaz García

Agradecimientos

A Óscar Díaz por haberme dado la oportunidad de trabajar en un proyecto tan interesante y haberme orientado a lo largo de su desarrollo proporcionándome buenos consejos e información.

A mis padres M^a Elena Ramos y Vicente Rivas por todos los ánimos, la confianza y el cariño que me han dado a lo largo de todos estos años. Sin ellos nada de esto habría sido posible.

A Lucía Bastida por todo el apoyo, la paciencia y la comprensión que me ha mostrado estos años. Además, quiero agradecerle el asesoramiento que me ha proporcionado en las partes legales del proyecto.

A mis abuelos Clementina Sanz y Jaime Rivas por haberme cuidado desde niño como si fuese un hijo y haberme proporcionado tantos buenos momentos.

Resumen

Este proyecto trata el problema del bloqueo de anuncios que existe actualmente en la web. Teniendo en cuenta la opinión de diversos autores sobre el tema, se analiza la problemática del uso de bloqueadores de anuncios. Además, se estudia cómo funcionan estos bloqueadores. En concreto, se analiza cómo está construido y cómo funciona, mediante ingeniería inversa, la extensión más popular en este campo, es decir, Adblock Plus.

Aparte de esto, se proponen una serie de soluciones para atacar al problema. Por último, se desarrolla e implementa una de las propuestas. Como resultado, se mejora el funcionamiento de Adblock Plus, en el sentido de que da más libertad al usuario para elegir lo que bloquea dando la oportunidad a anunciantes y proveedores de contenido de mantener su modelo de negocio basado en la publicidad.

Palabras clave: bloqueo de anuncios, bloqueadores, ingeniería inversa, Adblock Plus, publicidad.

Abstract

This project is about the problem of blocking ads that currently exists on the web. Taking into account the views of various authors on the subject, the issue of using ad blockers is analyzed. In addition, is studied how these blockers work. Specifically, is analyzed how is built and how works, by using reverse engineering, the most popular extension in this field: Adblock Plus.

Apart from this, a number of solutions to attack the problem are proposed. Finally, one of the proposals is developed and implemented. As a result, the functioning of Adblock Plus, in the sense that it gives more freedom to the user to choose which blocks giving the opportunity to advertisers and content providers to maintain their business model based on advertising, is improved.

Keywords: blocking ads, blockers, reverse engineering, Adblock Plus, advertising.

Tabla de contenido

Resumen.....	3
Abstract	3
Tabla de contenido	5
Índice de imágenes.....	7
1 Introducción	9
1.1 Objetivos	9
1.2 Antecedentes	9
1.3 Descripción General.....	10
1.4 ¿Qué es la ingeniería inversa?.....	10
1.4.1 ¿Qué es software libre?	14
2 La publicidad.....	17
2.1 Evolución de la publicidad.....	18
2.2 Medios de distribución.....	18
2.3 Legalidad de la publicidad	20
2.4 Publicidad aceptable en la web	21
2.5 Crítica a la publicidad	23
3 El bloqueo de publicidad.....	25
3.1 En que consiste el bloqueo de publicidad	25
3.2 Pros.....	25
3.3 Contras	26
3.4 Las opiniones de los usuarios.....	27
3.4.1 Artículo de la BBC (2013): <i>¿Son los anuncios de publicidad dirigida demasiado intrusivos?</i>	27
3.4.2 Artículo en adblockplus.org : <i>Los devastadores efectos del bloqueo de anuncios</i> (Palant, 2010).....	28
3.4.3 Artículo del Blog de Josemi (2010): <i>Gracias por no usar Adblock Plus</i>	29
3.4.4 Artículo del Blog de Enrique Dans (2014): <i>La hora de la verdad para la publicidad en la web</i>	29
3.5 Los bloqueadores más populares	30
3.6 Conclusiones	32
4 Adblock Plus.....	35
4.1 ¿Qué es Adblock Plus?.....	35
4.2 Una idea general de cómo funciona	40
4.3 Ingeniería inversa sobre Adblock Plus	42

4.3.1	Cómo está construido ADBlock Plus y que módulos usa	42
4.3.2	Clases implementadas	43
4.3.3	¿Cómo actúa ADBlock Plus?	52
5	Soluciones al problema del adblocking	65
5.1	Concienciar a los usuarios	65
5.2	Dejar de ofrecer contenido a los usuarios de bloqueadores	66
5.3	Dejar de ofrecer contenido gratuito.....	67
5.4	Modificar el tipo de publicidad	68
5.5	Modificar el funcionamiento de los bloqueadores de anuncios	68
6	Implementación de una solución	69
6.1	La idea	69
6.2	El proceso de desarrollo	70
6.2.1	Obteniendo el código fuente de ABP	70
6.2.2	Modificando el código	71
6.2.3	Instalando la extensión modificada	77
6.3	El resultado.....	78
7	Gestión del proyecto.....	85
7.1	Planificación.....	85
7.1.1	Identificación de tareas.....	85
7.1.2	Diagrama de Gantt	89
7.1.3	EDT	90
7.1.4	Gestión de Riesgos	91
7.1.5	Gestión de comunicaciones	91
7.1.6	Análisis de viabilidad	91
7.2	Seguimiento y control	92
8	Conclusiones	95
9	Referencias	97
10	Anexo 1: Glosario de términos	101

Índice de imágenes

Ilustración 1: Proceso de ingeniería de software normal	11
Ilustración 2: Proceso Ingeniería inversa de software.....	11
Ilustración 3: Proceso de ingeniería directa, inversa y re-ingeniería (gráfico realizado por Henry Hurtado)	12
Ilustración 4: Imagen extraída de la web antena3.com	27
Ilustración 5: Captura de la pantalla de opciones de AdMuncher.....	31
Ilustración 6: Comparativa de uso de memoria entra ABP y uBlock	31
Ilustración 7: Comparación de web sin usar AdBlock Plus (1) y usando (2).	36
Ilustración 8: Captura de pantalla del menú de AdBlock Plus.....	36
Ilustración 9: Captura de pantalla de la ventana para informar de errores.....	37
Ilustración 10: Captura con los elementos que detecta AdBlock Plus en una web.....	37
Ilustración 11: Montaje de capturas de pantalla con los menús desplegados de la ventana de preferencias de filtros (Subscripciones de filtros).....	38
Ilustración 12: Captura de pantalla de la ventana de preferencias de filtros (Filtros personalizados).....	39
Ilustración 13: Imagen extraída de la web adblockplus.org.....	41
Ilustración 14: Captura de pantalla de la interfaz de AdBlock Plus (recuperación y copias de seguridad).....	44
Ilustración 15: Montaje de capturas de pantalla de la interfaz de AdBlock Plus (acciones de filtros).....	46
Ilustración 16: Captura de pantalla de la interfaz de ABP (búsqueda de filtros)	46
Ilustración 17: Captura de pantalla de la interfaz de ABP (vista de los filtros)	47
Ilustración 18: imagen de adblockplus.org que muestra el comportamiento de las pestañas en objetos.	48
Ilustración 19: Captura de pantalla de la interfaz de ABP (selección y adición de una suscripción).	49
Ilustración 20: Captura de pantalla de la interfaz de ABP (acciones en suscripciones).....	50
Ilustración 21: Árbol realizado con el programa Dia que muestra la estructura de clases de los filtros.	51
Ilustración 22: Árbol realizado con el programa Dia que muestra la estructura de clases de las suscripciones.....	51
Ilustración 23: Diagrama de interacción entre clases de la función inicializadora del listener de filtros	53
Ilustración 24: Diagrama de interacción entre clases de la función inicializadora de la clase Policy	54
Ilustración 25: Diagrama de interacción entre clases de la función inicializadora de la clase PolicyImplementation	55
Ilustración 26: Diagrama de secuencia de la función inicializadora de la clase Synchronizer	56
Ilustración 27: Diagrama de secuencia de la función inicializadora de la clase Notification.....	56
Ilustración 28: Diagrama de secuencia de la función inicializadora de la clase SyncServiceObserver	57
Ilustración 29: Diagrama de secuencia de la función inicializadora de la clase optionsObserver	57

Ilustración 30: Diagrama de secuencia de la función inicializadora de la clase UI	58
Ilustración 31: Diagrama de secuencia del procesamiento de una página nueva	59
Ilustración 32: Diagrama de secuencia de la comprobación de si se debe cargar un elemento	60
Ilustración 33: Diagrama de secuencia del procesamiento de un elemento (parte 1).....	62
Ilustración 34: Diagrama de secuencia del procesamiento de un elemento (parte 2).....	63
Ilustración 35: Imagen extraída de vidaextra.com que sustituye los anuncios cuando se usa ABP	65
Ilustración 36: Mensaje de aviso para desactivar ABP en la web yes.fm (radio de los 40 principales).....	67
Ilustración 37: Imagen de la campaña de recaudación de Wikipedia	68
Ilustración 38: Primer boceto de la idea.....	70
Ilustración 39: Captura de pantalla mostrando cómo descargar el fichero de instalación de ABP	70
Ilustración 40: Código añadido a overlay.xul (1ª vez)	71
Ilustración 41: Código añadido a overlay.dtd	71
Ilustración 42: Código añadido a overlay.xul (2ª vez)	72
Ilustración 43: Código añadido a ui.js.....	72
Ilustración 44: Función openModDialog en ui.js.....	72
Ilustración 45: Contenido del archivo mod.xul (definición de la nueva ventana).....	73
Ilustración 46: Contenido del archivo mod.js (funciones de la nueva ventana).....	74
Ilustración 47: Contenido del archivo mod.dtd de la carpeta correspondiente al idioma castellano.....	75
Ilustración 48: Modificación en el archivo prefs.js	75
Ilustración 49: Modificaciones en contentPolicy.js (1º parte).....	76
Ilustración 50: Condiciones introducidas en la función processNode.....	77
Ilustración 51: Condición introducida en la función shouldLoad	77
Ilustración 52: Mensaje de error a la hora de instalar la extensión modificada.	78
Ilustración 53: Menú antes (izquierda) y después (derecha) de la modificación	79
Ilustración 54: Diálogo de modificación del nivel de seguridad (Tercer nivel activo)	79
Ilustración 55: Diálogo de modificación del nivel de seguridad (Primer nivel activo).....	79
Ilustración 56: Diálogo de modificación del nivel de seguridad (Segundo nivel activo).....	80
Ilustración 57: Diálogo de modificación del nivel de seguridad (Cuarto nivel activo).....	80
Ilustración 58: Efecto de tener ABP desactivado en animelv.net	80
Ilustración 59: Popups (a partir del segundo empezando por la izquierda) que se abren con ABP desactivado en animeflv.net	81
Ilustración 60: Efecto del nivel 1 de seguridad en animeflv.net	81
Ilustración 61: Popups abiertos (a partir del segundo empezando por la izquierda) cuando se aplica el nivel 1 de seguridad en animeflv.net	82
Ilustración 62: Efecto del nivel 2 de seguridad en animeflv.net	82
Ilustración 63: Efecto sobre los popups del nivel 2.....	82
Ilustración 64: Efecto del nivel 3 de seguridad en animeflv.net	83
Ilustración 65: Efecto del nivel 4 de seguridad en animeflv.net	83
Ilustración 66: Diagrama de Gantt	89
Ilustración 67: EDT.....	90
Ilustración 68: Diagrama de Gantt con desviaciones respecto a la planificación	94

1 Introducción

Documento elaborado para comentar el proceso y resultado del proyecto.

1.1 Objetivos

Este proyecto tiene como objetivo analizar el problema actual que hay entorno al bloqueo de anuncios en la web y presentar alguna solución para el mismo. Para ello se busca completar los siguientes objetivos:

- Presentar un informe sobre la situación actual del problema del adblocking en la Web conteniendo una serie de aspectos positivos y negativos del bloqueo de anuncios así como información sobre lo que opinan los usuarios.
- Mostrar cómo funciona AdBlock Plus siguiendo un proceso de ingeniería inversa tomando como punto de partida el código de libre acceso de la aplicación.
- Enumerar las diferentes soluciones que se podrían usar para ayudar a mejorar la situación del problema que se está abordando.
- Presentar la implementación de una solución. Esta solución consistirá en una mejora sobre AdBlock Plus que dará más control al usuario sobre lo que desea bloquear.
- Informar de la gestión del proyecto, explicando las actividades realizadas, cuándo han sido desarrolladas, y el esfuerzo (medido en tiempo) dedicado a las mismas. Se incluirán los aspectos oportunos en relación al seguimiento de comunicaciones y riesgos.

1.2 Antecedentes

El bloqueo de publicidad es un tema que está muy de moda. Cada vez hay más usuarios que utilizan este tipo de herramientas y esto está causando un problema en Internet. Existen multitud de artículos que ponen de manifiesto el impacto del uso de los bloqueadores de anuncios.

Sin embargo, apenas existen sitios en los que se especifique cómo funcionan realmente los bloqueadores. Sí, existen webs en las que se explica muy por encima cómo

funcionan, pero no lo suficiente como para poder usar ese conocimiento para encontrar soluciones. Además, se usa esta explicación como método de marketing para venderlos.

1.3 Descripción General

Este documento está formado por varios apartados principales:

- La publicidad de hoy en día
- El bloqueo de publicidad
- ¿Cómo funciona AdBlock Plus? (Ingeniería Inversa)
- Soluciones al problema del adblocking
- Implementación de una solución
- Gestión del proyecto
- Conclusiones
- Referencias

Además se incluye un anexo con los conceptos que requieren de una explicación.

1.4 ¿Qué es la ingeniería inversa?

En este proyecto se realiza ingeniería inversa para entender mejor la extensión AdBlock Plus pero, ¿qué es ingeniería inversa?

Según la definición ofrecida por la página web alegsa.com la ingeniería inversa “es el proceso de descubrir los principios tecnológicos de un dispositivo, objeto o sistema, a través de razonamiento abductivo de su estructura, función y operación. La ingeniería inversa trata de tomar algo (un dispositivo mecánico o electrónico, un software de computadora, etc.) para analizar su funcionamiento en detalle, generalmente para intentar crear un dispositivo o programa que haga la misma o similar tarea sin copiar la original”.

Existen diferentes usos de la ingeniería inversa:

- Analizar si un producto de la competencia infringe patentes de sus productos.
- En el área militar es utilizada para investigar (y copiar) las tecnologías de otras naciones, sin obtener planos ni detalles de su construcción o desarrollo.

- En el software y en el hardware, la ingeniería inversa muchas veces es empleada para desarrollar productos que sean compatibles con otros productos, sin conocer detalles de desarrollo de estos últimos.
- La ingeniería inversa también es empleada para comprobar la seguridad de un producto, generar keygens de aplicaciones, reparación de productos, etc.

Vamos a centrarnos específicamente en ingeniería inversa sobre software. Durante este proceso se va a contracorriente. Normalmente, la salida de la fase de análisis es la entrada de la fase de implementación, en cambio al realizar ingeniería inversa, la salida de la fase de implementación es la entrada de la fase de análisis. Las ilustraciones de a continuación ayudarán a entender el concepto:

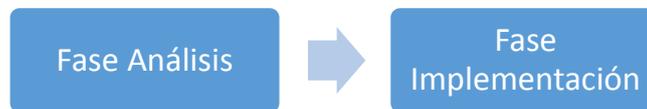


Ilustración 1: Proceso de ingeniería de software normal

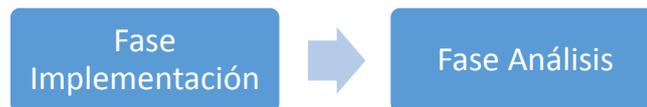


Ilustración 2: Proceso Ingeniería inversa de software

Durante el proceso de ingeniería inversa el producto desde el que se parte no se ve modificado, simplemente es un análisis del producto. Si se modificase dicho producto estaríamos hablando ya de re-ingeniería inversa.

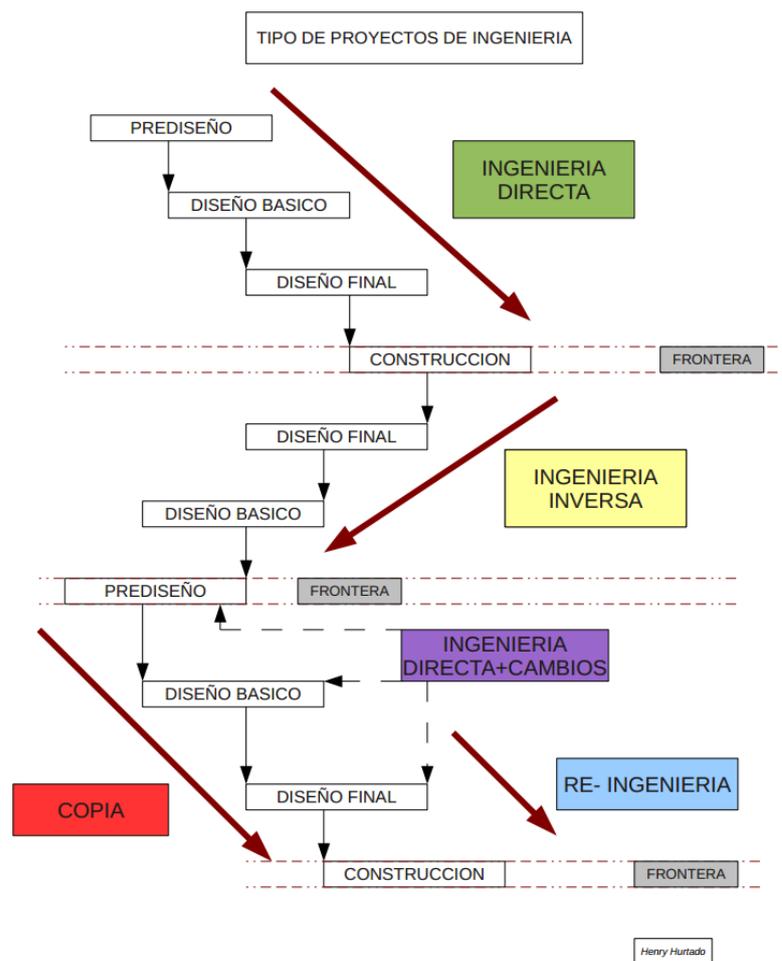


Ilustración 3: Proceso de ingeniería directa, inversa y re-ingeniería (gráfico realizado por Henry Hurtado)

A la hora de realizar la ingeniería inversa podemos tener dos posibilidades. El primer escenario se da al disponer del código fuente y el segundo al no disponer de él. En caso de encontrarnos en el segundo escenario el proceso sería más complicado. Por suerte Adblock Plus es de código libre.

Un último tema que hay que comentar sobre ingeniería inversa son las cuestiones legales ya que habrá quien, después de leer lo anterior, se pregunte si la ingeniería inversa es legal. La respuesta es sí, pero en determinados casos.

Rafael Soria Ruíz (2013), jurista especializado en Derecho de las Nuevas Tecnologías, declara que “la ingeniería inversa, referida a software, puede definirse como el procedimiento que tiene como objetivo recuperar el código fuente de un determinado programa de ordenador para su posterior utilización con diversos fines. Para que la ingeniería inversa sea legal, la finalidad con la que se realice es fundamental”.

Hay que distinguir entre simplemente observar o entre analizar para posteriormente utilizar el código fuente.

- Sólo observar: hay que tener en cuenta el artículo 100.3 de la Ley de Propiedad Intelectual que dice:

“3. El usuario legítimo de la copia de un programa estará facultado para observar, estudiar o verificar su funcionamiento, sin autorización previa del titular, con el fin de determinar las ideas y principios implícitos en cualquier elemento del programa, siempre que lo haga durante cualquiera de las operaciones de carga, visualización, ejecución, transmisión o almacenamiento del programa que tiene derecho a hacer.”

- Utilizar el código: las empresas interesadas en la realización concreta de ingeniería inversa podrán hacerlo siempre que cumplan con las condiciones establecidas en los artículos 100.5, 100.6 y 100.7 de esa misma ley:

“5. No será necesaria la autorización del titular del derecho cuando la reproducción del código y la traducción de su forma en el sentido de los párrafos a) y b) del artículo 99 de la presente Ley, sea indispensable para obtener la información necesaria para la interoperabilidad de un programa creado de forma independiente con otros programas, siempre que se cumplan los siguientes requisitos:

- a) Que tales actos sean realizados por el usuario legítimo o por cualquier otra persona facultada para utilizar una copia del programa, o, en su nombre, por parte de una persona debidamente autorizada.*
- b) Que la información necesaria para conseguir la interoperabilidad no haya sido puesta previamente y de manera fácil y rápida, a disposición de las personas a que se refiere el párrafo anterior.*
- c) Que dichos actos se limiten a aquellas partes del programa original que resulten necesarias para conseguir la interoperabilidad.*

6. La excepción contemplada en el apartado 5 de este artículo será aplicable siempre que la información así obtenida:

- a) Se utilice únicamente para conseguir la interoperabilidad del programa creado de forma independiente.*

- b) *Sólo se comunique a terceros cuando sea necesario para la interoperabilidad del programa creado de forma independiente.*
- c) *No se utilice para el desarrollo, producción o comercialización de un programa sustancialmente similar en su expresión, o para cualquier otro acto que infrinja los derechos de autor.*

7. Las disposiciones contenidas en los apartados 5 y 6 del presente artículo no podrán interpretarse de manera que permitan que su aplicación perjudique de forma injustificada los legítimos intereses del titular de los derechos o sea contraria a una explotación normal del programa informático.”

El no cumplimiento de los preceptos anteriores supone una vulneración de los derechos del autor y puede incurrir en la comisión de varios delitos como los recogidos en el artículo 197.3 o 270.3 del Código Penal.

En este proyecto se realiza ingeniería inversa para estudiar el código y, además, la copia a partir de la que se hace es legítima ya que es un software libre. Por lo tanto, no se incumple la normativa española.

1.4.1 ¿Qué es software libre?

Como se acaba de mencionar, Adblock Plus es software libre y según la web hispalinux.com “el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades centrales de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.”

Jurídicamente depende de varias leyes que son:

- Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual (en particular, arts. 95 a 104 relativos a los programas de ordenador).
- Ley 7/1998, de 13 de abril, sobre Condiciones Generales de Contratación.
- Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios y otras leyes complementarias.
- Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio electrónico (en particular, arts. 23 a 29 sobre contratación electrónica).

Por lo tanto, se debería tener la libertad de distribuir copias, sea con o sin modificaciones, gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. Al ser libre no hace falta pedir o pagar permisos. También se debería tener la libertad de hacer modificaciones y utilizarlas de manera privada para lo que quieras, sin tener que anunciar que dichas modificaciones existen. Y, si se publican los cambios, no hace falta avisar a nadie.

Sin embargo, se permiten ciertos tipos de reglas sobre la manera de distribuir software libre pero no deben entrar en conflicto con las libertades centrales. Por ejemplo, Copyleft es un tipo de regla que implica que se encarga de proteger las libertades centrales cuando se redistribuye el software.

Por lo tanto “software libre” no es lo mismo que “no comercial”. El desarrollo comercial del software libre ha dejado de ser inusual. De hecho, el software comercial libre es muy importante.

2 La publicidad

La publicidad no es considerada por todos de la misma forma, unos la ven como prensa libre o como la expansora de mercados, mientras que otros la ven como una forma de capitalismo o como una creadora de conformistas. Lo que sí está claro es que la publicidad está en nuestras vidas y es casi imposible huir de ella.

La RAE la define como:

- 1. f. Cualidad o estado de público.*
- 2. f. Conjunto de medios que se emplean para divulgar o extender la noticia de las cosas o de los hechos.*
- 3. f. Divulgación de noticias o anuncios de carácter comercial para atraer a posibles compradores, espectadores, usuarios, etc.*

Esta primera aproximación a la publicidad parece sencilla y no muestra la verdadera realidad de su naturaleza. Para Bernal y Márquez (1997) “la publicidad es el arte de persuadir a las personas para que realicen con frecuencia y en gran número algo que se quiere que hagan”. Esta se encuentra constituida por personas que se comunican con otras en relación con productos o servicios que un grupo suministra con el fin de satisfacer las necesidades o deseos de un grupo mayor.

Con esta nueva definición podemos observar una característica fundamental: su vocación por la persuasión. La publicidad quiere convencernos de que hagamos, pensemos o compremos algo.

El propósito más común de la publicidad es el aumento de las ventas de un producto. No se trata solo de influir en el individuo que compra el producto sino en el que influye en el comprador para que esa venta llegue a término. Bernal y Márquez (1997) reconocen cinco metas que justifican el uso de la publicidad:

1. Familiarizar al consumidor con el producto.
2. Recordar al consumidor que el producto existe.
3. Diseminar la información.
4. Vencer la inercia.
5. Agregar un valor más al producto.

Para De la Encarnación (2005) la publicidad tiene tres objetivos básicos:

- Informar: dar a conocer un nuevo producto y realzar la marca.
- Persuadir: influir en la conducta del consumidor de forma racional, emotiva o inconsciente para que se decante por un producto.
- Recordar: para conseguir compras repetitivas.

La publicidad influye en nuestras vidas, gracias a ella tomamos sopa de una marca y no de otra, compramos ordenadores de una marca conocida o nos peleamos por tener el último modelo de un smartphone de esa marca que nos lo vende como si fuera un producto maravilloso e imprescindible.

Existen diferentes aspectos que condicionan al consumidor: la familia, los grupos de convivencia y referencia, la clase social, la cultura y condicionamientos internos como la personalidad, la percepción, el aprendizaje, la motivación y la actitud.

2.1 Evolución de la publicidad

Como indica Vilches (2012) “la publicidad moderna surgió con el nacimiento de las agencias de publicidad en el siglo XIX, aunque los orígenes de la publicidad son casi tan antiguos como la historia de la humanidad”. Existen dos grandes acontecimientos históricos que asentaron las bases para el desarrollo de la publicidad tal y como la conocemos ahora: la Revolución Francesa (1789) que sentó las ideológicas, sociales y políticas, y La Revolución Industrial (en especial la segunda entre 1880 y 1914) que sentó las económicas.

La manera de construir la publicidad ha ido evolucionando con los cambios sociales, políticos, económicos, tecnológico-productivos y tecnológico-comunicativos. El incremento cuantitativo y cualitativo de la producción y la creciente similitud en la oferta han provocado que se busque que el producto destaque entre la competencia y para ello se utilizan diferentes medios.

2.2 Medios de distribución

Existen diferentes medios en los que se puede encontrar publicidad aunque no todos son igual de efectivos. Con cada medio se puede influir en un determinado grupo de personas, por ejemplo, si ponemos publicidad en un periódico esta solo será leída por los lectores de este, no llegará a toda la sociedad.

Para determinar que medio es más efectivo hay que ver a este como parte de un mensaje. Está cumpliendo una misión específica: aportar las cualidades estimulantes diferenciales de su lenguaje al poder persuasivo de la publicidad. Por este motivo no hay medios correctos o incorrectos sino que depende del mensaje que quieran transmitir, la audiencia a la que pretendan llegar, sus objetivos y del presupuesto con el que cuentan.

Algunos de los medios más utilizados son: periódicos, revistas, radio, televisión, publicidad en tránsito, publicidad en el exterior, mercado directo e Internet. Este último medio es el que nos interesa en este proyecto.

Internet cuenta con ventajas e inconvenientes y Guzmán (2003) explica algunos de ellos:

- Ventajas:
 - Coste eficiente. Los costes son independientes del tamaño de la audiencia.
 - Los anunciantes pueden dirigirse a sus audiencias específicas al colocar sus banners en páginas de temas relacionados.
 - Los mensajes pueden actualizarse fácil y rápidamente.
 - Los anuncios en Internet pueden ser interactivos.
 - Los banners aparecen con la frecuencia que se seleccione.
 - Los anunciantes en Internet pueden alcanzar una audiencia global.
- Desventajas
 - La publicidad en Internet no debe verse en el vacío. Debe ser un componente más de su estrategia de mercado en Internet.
 - Aunque la popularidad de Internet va en aumento, es difícil constatar los resultados de la publicidad a través de este medio.
 - La gama de los costes de publicidad en Internet varían considerablemente.

Como nos muestra Encarnación (2005) en el año 2004 la inversión publicitaria en Internet representaba un 2,6% del total y esta inversión ha ido aumentando con el paso de los años. La web Puromarketing (2015) asegura que el presupuesto para publicidad en medios tradicionales caerá frente a la digital que crecerá un 14,7% el año que viene.

Existen diferentes estudios como el realizado por la Asociación Europea de publicidad Interactiva que estudió 200 campañas realizadas en Internet y concluyó que la

publicidad online mejora el conocimiento de la marca, el recuerdo del anuncio y predispone favorablemente hacia el producto (Encarnación, 2005).

2.3 Legalidad de la publicidad

A continuación, veremos la legalidad de la publicidad, un tema muy relacionado con este trabajo ya que en Internet se genera cierto anonimato que lleva a que muchos se sientan invulnerables y atenten contra los derechos de los usuarios, creando y publicando publicidad que puede llegar a ser ilícita.

La publicidad es una materia importante que se encuentra regulada jurídicamente. En España las primeras normas sobre publicidad aparecieron con la Ley sobre Propiedad Industrial de 1902 pero fue en 1964 cuando se crea el Estatuto de la Publicidad donde se regularon normas referidas exclusivamente sobre esta.

Actualmente en nuestro código penal podemos ver artículos como el 282 en el que se penalizan acciones relacionadas con la publicidad:

“Serán castigados con la pena de prisión de seis meses a un año o multa de 12 a 24 meses los fabricantes o comerciantes que, en sus ofertas o publicidad de productos o servicios, hagan alegaciones falsas o manifiesten características inciertas sobre los mismos, de modo que puedan causar un perjuicio grave y manifiesto a los consumidores, sin perjuicio de la pena que corresponda aplicar por la comisión de otros delitos.”

No solo el código penal se encarga de disuadir a la sociedad del uso de estas malas prácticas en publicidad, sino que existen organismos como el Instituto Nacional de Consumo o el Consejo Superior de Medios que realizan un control publicitario.

Actualmente existe una ley que regula específicamente la publicidad, la Ley General de Publicidad 34/1988 del 11 de noviembre. En ella se define lo que es “publicidad ilícita” en su artículo 3:

- a) *“La publicidad que atente contra la dignidad de la persona o vulnere los valores y derechos reconocidos en la Constitución, especialmente a los que se refieren sus artículos 14, 18 y 20, apartado 4.
Se entenderán incluidos en la previsión anterior los anuncios que presenten a las mujeres de forma vejatoria o discriminatoria, bien utilizando particular y directamente su cuerpo o partes del mismo como mero objeto desvinculado del*

producto que se pretende promocionar, bien su imagen asociada a comportamientos estereotipados que vulneren los fundamentos de nuestro ordenamiento coadyuvando a generar la violencia a que se refiere la Ley Orgánica 1/2004, de 28 de diciembre, de Medidas de Protección Integral contra la Violencia de Género.”

- b) “La publicidad dirigida a menores que les incite a la compra de un bien o de un servicio, explotando su inexperiencia o credulidad, o en la que aparezcan persuadiendo de la compra a padres o tutores. No se podrá, sin un motivo justificado, presentar a los niños en situaciones peligrosas. No se deberá inducir a error sobre las características de los productos, ni sobre su seguridad, ni tampoco sobre la capacidad y aptitudes necesarias en el niño para utilizarlos sin producir daño para sí o a terceros.”*
- c) “La publicidad subliminal.”*
- d) “La que infrinja lo dispuesto en la normativa que regule la publicidad de determinados productos, bienes, actividades o servicios.”*
- e) “La publicidad engañosa, la publicidad desleal y la publicidad agresiva, que tendrán el carácter de actos de competencia desleal en los términos contemplados en la Ley de Competencia Desleal.”*

En el artículo 4 se define lo que es la “publicidad subliminal”:

“A los efectos de esta ley, será publicidad subliminal la que mediante técnicas de producción de estímulos de intensidades fronterizas con los umbrales de los sentidos o análogas, pueda actuar sobre el público destinatario sin ser conscientemente percibida.”

En Internet la publicidad nos persigue. Es muy difícil poder navegar sin encontrar nada de publicidad y, a veces, hace que te sientas espiado. Por ejemplo, ¿a quién no le ha pasado que ha buscado un producto y luego navegando en otras páginas aparece publicidad sobre este tipo de productos por todos lados?

2.4 Publicidad aceptable en la web

Este tipo de publicidad es la que debería existir únicamente en la web ya que no molesta a los usuarios. Para definir la publicidad aceptable se utilizan los siguientes criterios:

- Sólo publicidad estática (nada de animaciones, sonidos o similar).
- Preferiblemente sólo texto, no imágenes que llamen la atención.

- Ubicación del anuncio:
 - Los anuncios nunca deben ocultar el contenido de la página (por ejemplo, obligando al usuario a pulsar en un botón para cerrar el anuncio y poder ver la página).
 - En las páginas que tengan un texto a leer por el usuario, los anuncios no deben estar situados en medio del texto, interrumpiendo el flujo de lectura. En vez de eso, pueden estar situados encima, debajo o a los lados del texto. Lo mismo se aplica a páginas de resultados de búsqueda, no se pueden mezclar los resultados de pago¹ con los resultados orgánicos².
 - Cuando los anuncios estén situados encima del contenido de la página no pueden obligar al usuario a mover el texto hacia abajo. Dado que el espacio vertical disponible suele ser de al menos 700 píxeles, la publicidad no debe ocupar más de 1/3 de esa altura. Los resultados de pago en las páginas de búsquedas pueden ocupar más espacio, pero nunca podrá haber más resultados de pago que resultados reales.
 - Cuando la publicidad esté en un lateral, debe dejar espacio suficiente para el contenido. Dado que el espacio horizontal disponible suele ser de al menos 1000 píxeles, la publicidad no debe ocupar más de 1/3 de ese espacio.
- La publicidad debe estar marcada claramente como tal con la palabra "Publicidad" o su equivalente y debe ser distinguible del contenido de la página (mediante un borde y/o un color de fondo diferente).
- Los requisitos de señalización y ubicación no se aplican para hipervínculos con ID referentes a programas de afiliación³ insertados en el contenido de páginas web. En estos casos existen unos criterios adicionales:
 - Las redirecciones originadas en el hipervínculo no deben redirigir a ninguna otra página web diferente a la página de destino.
 - En los textos, no más del 2% de las palabras pueden ser enlazadas con propósitos monetarios.

¹ Enlaces patrocinados que generan beneficio por clic al buscador.

² Resultados relacionados con la búsqueda del usuario.

³ Sistemas que permiten que los asociados puedan obtener un beneficio promocionando de alguna manera los productos y/o servicios de una o más compañías.

- Los hipervínculos no deben tener diferente formato o comportamiento que cualquier otro enlace.
- Los hipervínculos no debe inducir a error o malentendido, tanto en el contenido como en la colocación.

2.5 Crítica a la publicidad

La publicidad se utiliza para vender algo, cuanto más se venda, mejor y esto lleva a que se vendan cosas que la gente ni quiere ni necesita. Cada vez más se intenta destacar un producto sobre sus competidores y se utilizan campañas más agresivas e intrusivas que llegan a molestar. Centrándonos en el contexto que nos incumbe, Internet, los usuarios son obligados a ver anuncios que no les interesan e incluso, en algunos casos, estos anuncios son fraudulentos, es decir, algunas páginas poseen publicidad del tipo de “ha ganado 1000€” que solo quieren aprovecharse del internauta. También existen páginas con publicidad de contenido inapropiado como los casos de publicidad pornográfica. Por este tipo de publicidad muchos usuarios han comenzado a utilizar extensiones como Adblock Plus.

Sin embargo, todo tiene su lado bueno. Gracias a la publicidad podemos disfrutar de muchas cosas de manera gratuita en diferentes plataformas (televisión, Internet, radio...), ya que no solo sirve de herramienta de publicidad sino también de herramienta de financiación para los espacios que la alojan. Internet es prácticamente la única plataforma en la que se ha encontrado una manera bastante eficaz de evitar que el usuario tenga que ver la publicidad (los adblockers o bloqueadores de publicidad). Por esto es realmente importante entender y encontrar una solución para el problema del que se hablará en el siguiente apartado: el problema del bloqueo de publicidad.

3 El bloqueo de publicidad

En este apartado se analizará el problema del bloqueo de publicidad. Primero se verá qué es el bloqueo de publicidad y qué aplicaciones existen para hacerlo. Después, se analizarán los factores positivos y negativos unidos al bloqueo de cierto contenido de la web.

Tras ver estos factores, conoceremos que opinan realmente los usuarios de la web, tanto los que acceden a la información como los que la publican. La opinión de los usuarios es muy importante, ya que aunque haya innumerables motivos para bloquear o no los anuncios, lo que opinen la mayoría de usuarios dictará el camino a seguir. Estas opiniones se han extraído de diferentes foros y noticias.

Por último se analizarán las aplicaciones de bloqueo de anuncios más populares en estos momentos.

3.1 En que consiste el bloqueo de publicidad

El bloqueo de publicidad o adblocking consiste en una serie de técnicas que evitan que se muestre publicidad en las páginas web. La mayoría de los bloqueadores de anuncios son extensiones de navegador que suelen ser gratis y de código abierto.

3.2 Pros

El adblocking se presenta hacia los usuarios como un elemento favorable para la navegación. Vamos a ver en qué ayuda realmente a los usuarios.

- La principal característica favorable de los sistemas de bloqueo de anuncios es que proporcionan una experiencia más agradable al usuario de la web ya que este deja de ser continuamente bombardeado con publicidad que, posiblemente, no le interese.
- También se bloquea contenido inadecuado de la web, aunque no sea un anuncio. Existen muchas páginas en las que la publicidad contenga fraudes del tipo “has ganado un viaje al Caribe” o contenido pornográfico, su bloqueo ayuda a ingenuos y menores de edad.

- Favorece que las páginas se carguen con mayor fluidez, cosa que también repercute en la experiencia del usuario. Si no hay que cargar los anuncios, antes se cargará la web.
- En móviles ahorrará recursos como consumo de batería y datos.
- Otra cosa, no tan evidente para el usuario, es que proporciona algo más de privacidad. Por ejemplo, AdBlock Plus incluye un sistema de anti-rastreo. Este sistema consiste en una lista de filtros llamada EasyPrivacy que bloquea elementos de rastreo como cookies o scripts.

En sí los pros que hemos visto, mejoran la experiencia del usuario y su privacidad en la web.

3.3 Contras

Vistos los pros, los cuales parecen convincentes, vamos a ver los contras, es decir, por qué bloquear la publicidad puede llegar a ser perjudicial.

- Perjudica al sitio web ya que deja de tener su única fuente de ingresos, la publicidad.
- Perjudica al usuario porque su uso puede llevar a que mucha información que actualmente es gratuita pase a ser de pago.
- Perjudica a las compañías que han creado la publicidad para este medio porque finalmente no se está viendo.

La publicidad es el origen de los ingresos de la mayoría de sitios web. Sin esta muchos de ellos no existirían debido a la imposibilidad de afrontar los costes de mantenerlas. Por ello, el primer contra es el daño que produce a las webs.

Parece que este primer inconveniente es solo problema de los administradores o propietarios de webs, pero pensar de esta manera es erróneo. Internet es una fuente de información, por regla general, gratuita y de fácil acceso. Pero, ¿por qué es gratuita? Sencillo, porque las webs tienen la posibilidad de ofrecer contenido gratuito, gracias a que obtienen sus ingresos de otra manera: la publicidad. Por esto, si la publicidad desaparece, también lo hará mucho contenido gratuito y esto repercute directamente al usuario. Podrían desaparecer, incluso, numerosas redes sociales, o pasar a ser de pago.

En muchas páginas ya se muestran mensajes de este tipo:

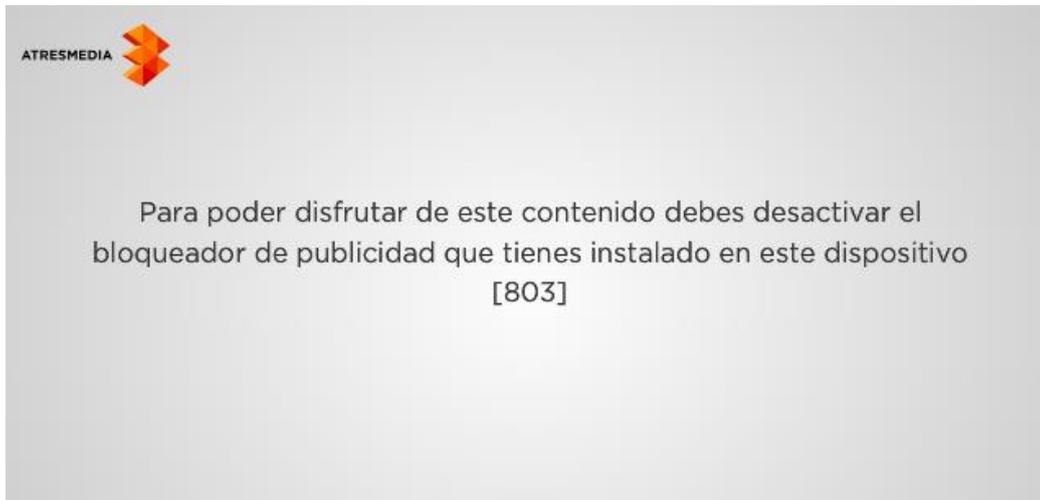


Ilustración 4: Imagen extraída de la web antena3.com

3.4 Las opiniones de los usuarios

A continuación vamos a ver una serie de artículos interesantes que muestran la opinión de diversos usuarios. Cada artículo ofrece un punto de vista diferente al tema que abordamos.

3.4.1 **Artículo de la BBC (2013): ¿Son los anuncios de publicidad dirigida demasiado intrusivos?**

En este artículo se muestra la opinión de seis estudiantes del Museo Británico. El resumen de sus opiniones es simple: ellos creen que este tipo de publicidad es demasiado intrusiva. Tienen la sensación de que Internet sabe demasiado sobre ellos debido a la manera en la que se les proponen diferentes anuncios y esto les resulta inquietante, es como si les estuviesen espiando. Además, por si fuera poco esto, muchas veces la publicidad (que se supone que es dirigida) no da ni de lejos en lo que realmente necesita el usuario, por lo que se convierte en algo aún más molesto. Alguno de los interesados ya reconoce que está dejando de utilizar ciertas aplicaciones por este tipo de publicidad. También se menciona que tener que ver o clicar en publicidad para acceder a contenido no es nada apetecible. Posiblemente, esto también desemboque en dejar de utilizar plataformas que utilizan este tipo de publicidad o, en su defecto, a utilizar un bloqueador de anuncios.

En conclusión, este tipo de publicidad no es para nada algo aceptable para los usuarios. A pesar de que no se trata el tema del adblocking en las entrevistas, se deja entrever que estos usuarios agradecerían tener bloqueada esa publicidad cuando navegasen.

3.4.2 Artículo en adblockplus.org: *Los devastadores efectos del bloqueo de anuncios* (Palant, 2010)

En este artículo se recolectan diferentes opiniones sobre el bloqueo de anuncios en las que, grosso modo, se considera que este no está perjudicando a las páginas web. Al consistir en diferentes opiniones, vamos a resumir algunas de las ideas claves que aparecen:

- Las empresas insisten en utilizar trucos para tratar de obligar a mirar los anuncios que no deseas.
- Si no consigues hacer ventas no culpes al consumidor, has contratado al personal de ventas equivocado, o no fuiste lo suficientemente convincente, o el producto no fue lo suficientemente bueno.
- Si los usuarios del bloqueador de anuncios tienen una disposición previa a ignorar los anuncios de todos modos, convencerlos de desactivar sus bloqueadores simplemente servirá para aumentar el número de vistas inservibles (dud-views).
- Los usuarios no son culpables de la disminución de ingresos.
- La financiación de contenido a través de la publicidad es altamente ineficiente. De las personas que visitan un sitio web, por lo general, sólo una pequeña proporción hacen clic en el anuncio y de esos solo unos pocos compran el producto.
- La publicidad intrusiva está destruyendo la tolerancia de los usuarios por el modelo de negocio de las webs, es decir, la financiación por publicidad.
- Aunque parezca que el uso de los bloqueadores de anuncios conlleve que no se generen ingresos, a menudo los usuarios que acceden a las webs con estas extensiones activas contribuyen al sitio de otras maneras como, por ejemplo, publicando enlaces en redes sociales que atraen a otros usuarios que no utilizan bloqueadores.
- Los editores de medios en línea deberían cambiar la manera de gestionar su publicidad y sus fuentes de ingresos. Pedir a los consumidores que cambien no es un camino al éxito.

- Lo que se necesita es un sistema diferente, que ofrezca valor a las tres entidades (usuario, sitio web y marca publicitada).
- El problema que tenemos es que los anunciantes y sus agentes, en su búsqueda de clics, están constantemente presionando a los editores hacia conductas y tamaños de anuncios cada vez más intrusivos. Si se niegan a publicar un anuncio intrusivo parece que siempre hay alguien que esté dispuesto a publicarlo si el precio es correcto.
- Un medio electrónico trae de 15 a 20 veces menos ingresos publicitarios que un medio impreso.

3.4.3 Artículo del Blog de Josemi (2010): Gracias por no usar AdBlock Plus

En este artículo se habla del daño que AdBlock Plus hace a ciertas páginas que reciben ingresos gracias a la publicidad. Se dice que en unos años este va a ser un problema de gran relevancia que pondrá contra las cuerdas a muchas webs. Se recalca la idea de que hay sitios que subsisten gracias a esa publicidad y que verla no conlleva esfuerzo al usuario, además no hay que olvidar que se está accediendo a contenido gratuito gracias a esa página. Finalmente, se recuerda que los editores de las páginas tienen derecho a no dejar ver el contenido a los usuarios que bloqueen los anuncios o a convertir sus webs en lugares de pago (ej. New York Times).

Se ofrecen dos soluciones. La primera radica en concienciar a los usuarios de no utilizar la extensión. La segunda, por el contrario, pide el cambio en la otra parte, es decir, en la publicidad. Se pide utilizar publicidad poco intrusiva.

Para uno de los lectores de ese blog la segunda solución es la que le convencería de desinstalar el AdBlock Plus. Dicho lector dice lo siguiente en su comentario: “si pusieran publicidad que se queda en un solo lugar desinstalaría AdBlock Plus”.

3.4.4 Artículo del Blog de Enrique Dans (2014): La hora de la verdad para la publicidad en la web

En este artículo, al contrario que en los dos anteriores en los que se podía ver una opinión posicionada en uno de los dos extremos (en contra o a favor de bloquear la publicidad), se aborda el tema del adblocking desde una posición neutra. Se analizan unos datos que hablan de las crecientes cifras de usuarios que usan esta herramienta. Para Enrique Dans la publicidad está en un momento en el que tiene que evolucionar y considera que un amplio porcentaje de la que se utiliza regularmente en la web es muy molesta ya sea por su

formato o por su abundancia. El modelo de negocio basado en la publicidad se encuentra condicionado por las creencias de los usuarios que consideran que este tipo de herramientas son “una forma de evitar un abuso y de promover la autorregulación del mercado”.

Uno de los problemas que ve es que estas herramientas bloquean toda la publicidad por defecto por lo que tampoco se ve la publicidad que realmente no molesta. Existe una iniciativa por parte de Adblock Plus en la que se compromete a realizar una lista blanca con formatos de publicidad no considerados intrusivos para que estos no sean bloqueados.

En cuanto al uso de bloqueadores en dispositivos móviles, dice que este es todavía escaso pero que hay empresas que están trabajando en ello porque supondría un ahorro de tiempo de acceso, ancho de banda y de recursos como la batería.

La única opción viable que él ve es que las webs se comprometan a utilizar publicidad poco intrusiva que convenza a sus usuarios de no usar bloqueadores. Las soluciones agresivas como prohibir el uso de bloqueadores no llevarían a ningún lado.

3.5 Los bloqueadores más populares

A continuación vamos a ver una serie de ejemplos de extensiones que hacen la labor de bloquear los anuncios de las webs:

- Adblock Plus (<https://adblockplus.org/>): se trata de la extensión de este tipo más popular. Veremos sus características más adelante.
- Adblock (<https://getadblock.com/>): no hay que confundirlo con el extinto Adblock que dio origen a Adblock Plus. Esta extensión está creada para Chrome y Safari, en los que suma 40 millones de descargas.

Es un proyecto financiado mediante donaciones. A diferencia de otros bloqueadores no cobran grandes empresas para no filtrar sus anuncios.

Una característica destacable es que lo que se considera propaganda no llega a descargarse, por lo que se reduce el tiempo de carga y el uso del ancho de banda. También tiene algunas opciones para que el usuario decida si permite algún tipo de publicidad. Por ejemplo, da la opción de permitir publicidad de texto. Esto hace que, por ejemplo, al hacer búsquedas en Google se vean los enlaces patrocinados.

- AdMuncher (<https://www.admuncher.com/>): la principal diferencia de esta herramienta con respecto a los demás es que es una aplicación de escritorio. Esto

es que, en vez de instalarse como extensión para el navegador, esta se instala en el ordenador y se adapta a los diferentes navegadores que se utilicen.

Contiene bastantes opciones para personalizar. No solo elimina la publicidad, sino que es capaz de suprimir elementos multimedia como la imagen de fondo de la web, música, vídeos, e incluso permite la sustitución de texto. Aquí vemos una imagen en la que se muestran las opciones con respecto al filtrado de las web:

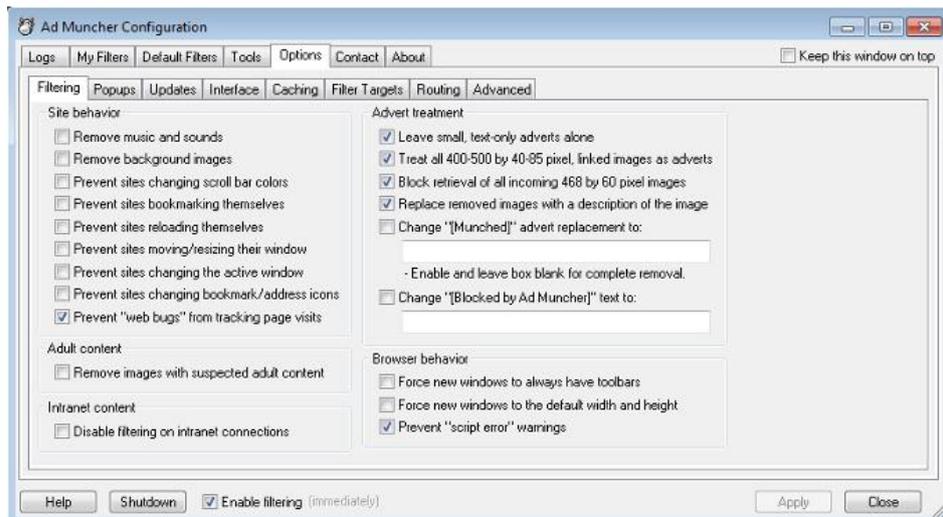


Ilustración 5: Captura de la pantalla de opciones de AdMuncher

- **uBlock** (<https://github.com/gorhill/uBlock>. No tienen web, está todo en GitHub): la principal característica diferencial de esta extensión es su bajo uso de RAM. Esto convierte a esta extensión en una gran alternativa frente a Adblock Plus (puede llegar a consumir 2 gigas de memoria RAM). Aquí vemos una comparativa de uso de memoria entre ambas extensiones:

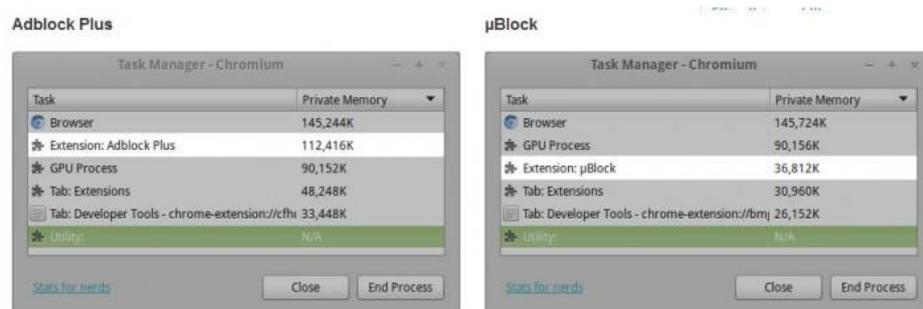


Ilustración 6: Comparativa de uso de memoria entre ABP y uBlock

Su funcionamiento es similar a Adblock Plus, aunque no llega a ser tan completo. Sin embargo, cumple con el objetivo básico de los bloqueadores de anuncios, por lo que es bastante interesante.

Lo que hemos visto hasta ahora son las características diferenciales de cada una de las extensiones. En cuanto a sus similitudes, podemos decir que su funcionamiento es bastante parecido, ya que trabajan con filtros, es decir, se comparan estos filtros con los elementos a cargar en la web, para ver si deben ser bloqueados o no. Además, todos permiten modificar manualmente los filtros y añadir diferentes subscripciones a filtros.

Estas extensiones son ahora mismo las más usadas, siendo de entre todas la más popular Adblock Plus. Por esto, vamos a tomarla como ejemplo en este proyecto. Sin embargo, no hay que olvidarse de las demás, ya que son claras alternativas al uso de ABP (Adblock Plus).

3.6 Conclusiones

La publicidad intrusiva está arrasando con la confianza de muchos usuarios, haciendo que estos sientan la necesidad de protegerse de este tipo de publicidad. Aquí entran los bloqueadores de anuncios. Posiblemente, el número de usuarios de bloqueadores no sería tan alto si no existiese publicidad de este tipo o fuese más moderada. Es molesto estar accediendo a contenido y cada vez que se haga un clic se nos abra una ventana nueva con publicidad para nada apropiada.

Obviamente, también habrá usuarios de adblockers que seguirían usándolos sin la existencia de esta publicidad. Esto también debe ser un punto a cambiar, ya que no podemos pretender acceder a contenido gratis sin un pequeño esfuerzo. Este esfuerzo no sería una tarea titánica, sino que simplemente consistiría en tener a los lados de la web publicidad que no moleste, que no se mueva, que sea conveniente con la temática de la web y que no habrá nuevas ventanas sin previo aviso.

Si no obtenemos una solución pronto, podría ser el fin del contenido gratuito. Hay que pensar que esto es algo muy grave. Simplemente imaginar que para obtener cualquier información habría que abonar una cierta cantidad de dinero hace pensar que la gente podría incluso dejar de usar la web como fuente de información.

Por parte de Adblock Plus se está proponiendo una iniciativa para permitir publicidad aceptable desde la versión 2.0. Esta opción viene activada por defecto porque, según afirman en la web de Adblock Plus, sino no se conseguirían obtener las meta que buscan, que es que los publicistas creen publicidad aceptable y mejor. Sin embargo, se puede desactivar esta funcionalidad, aunque no debería poder desactivarse para realmente conseguir resultados.

Podemos encontrar más información sobre esta iniciativa en esta dirección web: <https://adblockplus.org/es/acceptable-ads>. Los criterios para identificar anuncios aceptables se pueden encontrar en el apartado 2.4 de este mismo documento.

Otro ejemplo de aplicación que está trabajando en el tema de la publicidad es Snapchat. Según Snapchat (2014) su publicidad es totalmente voluntaria. Si no se quiere hacer clic no pasa nada. No te espían, solo te ofrecen lo que creen que te puede gustar. También comenta que la publicidad debe pasar unos filtros y requisitos.

En conclusión, la que parece la mejor manera para solucionar este problema sería que las webs y los usuarios pusieran de su parte. Por parte de las webs y los publicistas debería hacerse el esfuerzo de crear buena publicidad, es decir, aceptable y efectiva. Por su parte, los usuarios deberían aceptar esta publicidad y no bloquearla con extensiones. De esta manera, obtendríamos un equilibrio que permitiría a las webs financiadas únicamente por publicidad sobrevivir y a los usuarios navegar sin molestos anuncios.

4 AdBlock Plus

Para poder comprender mejor cómo funciona el bloqueo de anuncios vamos a analizar la que posiblemente es la extensión más utilizada con este fin. Efectivamente, nos referimos a AdBlock Plus. Con más de 300 millones de descargas esta extensión se ha convertido en el referente en su campo.

AdBlock Plus tiene diferentes versiones y cada una de ellas es para un navegador. Aunque estas versiones tienen un funcionamiento parecido, la manera en la que están compuestas difiere en muchos aspectos (lenguaje, por ejemplo). Por esto, vamos a tomar como referencia una de ellas. La lógica dice que se tome la versión de Firefox ya que fue la inicial y la que más recorrido lleva.

Antes de comenzar a leer este apartado, es conveniente echar un vistazo al [Anexo 1](#), que contiene un glosario con términos que se utilizarán a lo largo de este capítulo. La primera aparición en el texto de estos términos definidos en el glosario será resaltada con negrita.

4.1 ¿Qué es AdBlock Plus?

AdBlock Plus es una extensión (o plugin) disponible para diferentes navegadores (Firefox, Internet Explorer, Google Chrome,...) que filtra los contenidos de una web. Esta extensión evita que los usuarios tengan que ver publicidad o contenido inadecuado mientras navegan por Internet. Un efecto colateral de filtrar los contenidos es que se acelera la velocidad de carga de la página.

Gracias a esta extensión podemos ver como una página repleta de anuncios y mensajes molestos se convierte en algo que aporta únicamente contenido. Aquí podemos ver un ejemplo:



Ilustración 7: Comparación de web sin usar Adblock Plus (1) y usando (2).

Esta herramienta tiene una **interfaz de usuario** bastante intuitiva que nos permite configurar diferentes aspectos. Si clicamos en el botón con el símbolo de ABP () se nos desplegará un menú con diferentes opciones. Antes de analizar dichas opciones, veamos el menú mencionado:

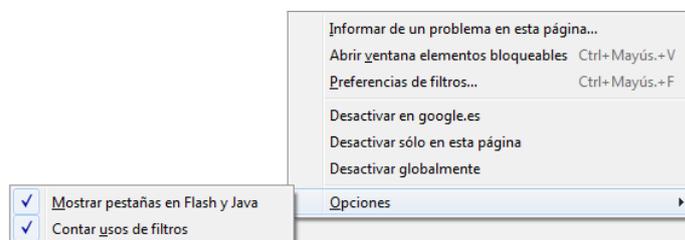


Ilustración 8: Captura de pantalla del menú de Adblock Plus

Ahora veamos que implica cada una de las opciones.

- Informar de un problema en esta página: Esta opción nos permite informar al equipo de Adblock Plus sobre errores que hayamos encontrado en la herramienta mientras se navegaba por la página actual. Cuando clicamos encima se nos abre la siguiente ventana:

- Preferencias de filtros...: Esta opción nos muestra una ventana en la que podremos realizar numerosas acciones de edición de los filtros que tiene Adblock Plus. Podremos añadir nuevos filtros y suscripciones, eliminarlos, reordenarlos, editarlos, activarlos o desactivarlos. Incluso se pueden crear copias de seguridad de los filtros para su restauración en un futuro en caso de fallo o simplemente porque nos gustaba más como estaba configurado antes. Además de tener la función de configuración, también se puede ver que filtros forman parte de las suscripciones y el número de veces que se ha utilizado un filtro.

Vamos a ver ahora en profundidad, con imágenes, las opciones de las que se dispone. Primero veremos el menú de las suscripciones a filtros, es decir, de las listas de filtros ya creadas:

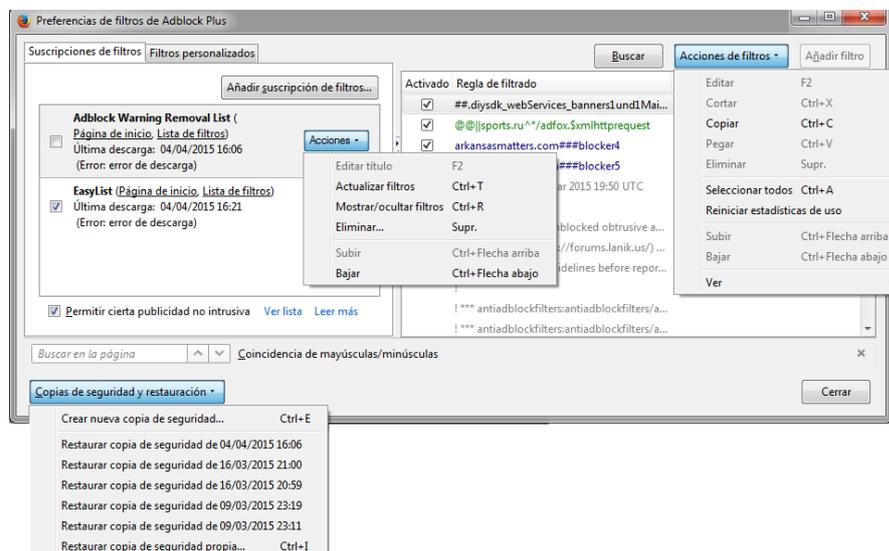


Ilustración 11: Montaje de capturas de pantalla con los menús desplegados de la ventana de preferencias de filtros (Suscripciones de filtros)

El siguiente pantallazo corresponde al menú de los filtros creados por el propio usuario. Como los menús desplegables que hay son iguales a los de la imagen anterior, no se mostrarán desplegados para así tener una visión más clara de la ventana.

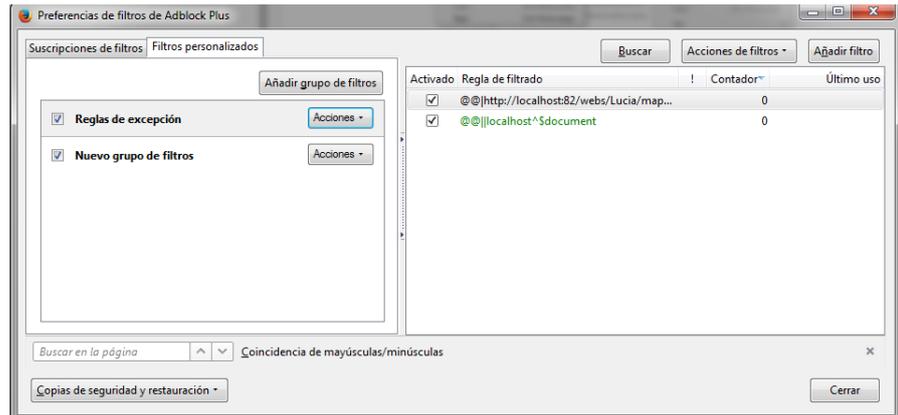


Ilustración 12: Captura de pantalla de la ventana de preferencias de filtros (Filtros personalizados)

- Desactivar en...: Esta opción nos permite desactivar AdBlock Plus para todo el dominio en el que nos encontramos navegando. Por ejemplo, si nos encontramos en un foro, ABP no bloqueará nada en ninguna de sus páginas.
- Desactivar sólo en esta página: Seleccionando esta opción se desactivará AdBlock Plus únicamente en la página actual, es decir, si vamos a otra página dentro de un mismo dominio, ABP sí que efectuará su función.
- Desactivar globalmente: Desactiva completamente AdBlock Plus.
- Opciones:
 - Mostrar pestañas en Flash y Java: Al tener esta opción seleccionada, en los cuadros de contenido que tienen Flash y Java se mostrará una pequeña pestaña que permitirá bloquear el elemento.
 - Contar usos de filtros: Al tener esta opción activada se lleva la cuenta de cuantas veces se han utilizado cada uno de los filtros. Esta información está accesible en la ventana de preferencias de filtros.

AdBlock Plus es, digamos, la actualización de una extensión ya existente llamada AdBlock, que mejora en la facilidad de uso y añade nuevas características. Además de bloquear la publicidad, se puede usar la ocultación de elementos, pudiendo ser el propio usuario quién decida qué elementos ocultar. Otras nuevas ventajas con respecto a AdBlock son el contador de elementos bloqueados y la posibilidad de desactivar los filtros sin, necesariamente, eliminarlos.

4.2 Una idea general de cómo funciona

AdBlock Plus analiza las direcciones de origen de las **peticiones HTTP** y el tipo de elemento a cargar, y en función del resultado las bloquea. También puede bloquear **IFrames**, **scripts** o elementos Flash. De esta manera se evita acceder a contenido no adecuado.

Por otro lado, tenemos la ocultación de elementos. Para realizar esta tarea, la extensión utiliza **hojas de estilo de usuario** generadas automáticamente. Así en vez de tener que bloquear los elementos, simplemente los oculta. Esto se aplica a cosas como la publicidad en forma de texto o imagen que aparecen en una página. Para generar estas hojas de estilo, se traducen las reglas de ocultación a **reglas CSS**. Estas reglas de ocultación son las que dictan si un elemento se debe ocultar, son como filtros. Un ejemplo de regla traducida a CSS que podríamos encontrar en estas hojas de estilo de usuario es este:

example.com#div(evil_ad)



```
@-moz-document domain(example.com)
{
  div#evil_ad, div.evil_ad
  {
    display: none !important;
  }
}
```

Lo que quiere decir este fragmento es que todos aquellos contenedores de la web “example.com” con identificador “evil_ad” no serán mostrados (display: none).

Muchas veces hay que aplicar tanto la ocultación como el bloqueo de elementos. Por ejemplo, puede que se bloquee una petición pero que quede una imagen residual en la web. Aquí entraría la ocultación de elementos.

Para decidir si bloquear u ocultar un elemento, AdBlock Plus tiene una serie de listas de filtros. La lista más popular es EasyList. Estas listas se actualizan automáticamente añadiéndose nuevos filtros para adaptarse a las características de los nuevos elementos a bloquear u ocultar. Los usuarios también pueden incluir sus propias listas de filtros. Estos filtros pueden estar compuestos por **expresiones regulares**. Además AdBlock Plus soporta una sintaxis mejorada para poder controlar de mejor manera los filtros.

Aquí vemos un esquema muy general de cómo funciona AdBlock Plus:

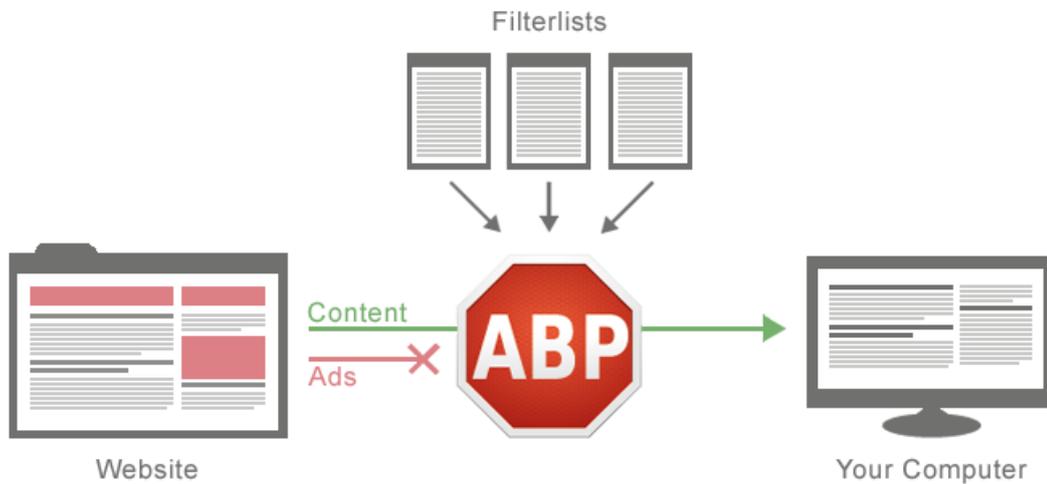


Ilustración 13: Imagen extraída de la web adblockplus.org

Ahora, vamos a ver los pasos que se dan en la ejecución de ABP:

1. Cuando se inicia, AdBlock Plus establece una serie de objetos (**Observers** y **Listeners**) para detectar eventos en el navegador (carga de páginas, clicks...). En este proceso de inicialización también se cargan en el navegador las hojas de estilo de usuario generadas por la extensión, de tal manera que cuando una página sea cargada, se le aplicarán las reglas de ocultación de elementos automáticamente.
2. Cuando se realiza la petición para abrir una nueva página, ABP analiza si la página que se solicita se debe cargar, porque podría ser un molesto pop-up que debe cerrarse. Si ABP considera adecuada la página solicitada, permite que esta se cargue. En caso contrario, se cancela la carga.
3. Durante el proceso de carga de la página, AdBlock Plus decide si cargar o no sus elementos. Una cosa es que se permita cargar la página y otra muy diferente es que sus elementos sean adecuados. Por esto, por cada petición de carga de los elementos de la web se decide si el elemento debe cargarse o no. Los elementos a analizar son scripts, imágenes, links...
4. Una vez que termine este proceso, tendremos una página web libre de anuncios, salvo los permitidos por ABP.

4.3 Ingeniería inversa sobre AdBlock Plus

Vamos a ver cómo está construido AdBlock Plus. Para ello veremos lo siguiente:

- Cómo está construido AdBlock Plus
- **Clases** implementadas
- **Algoritmo** que gobierna AdBlock Plus (cómo actúa)

4.3.1 Cómo está construido AdBlock Plus y que módulos usa

Antes de ver en profundidad como funciona AdBlock Plus es necesario comentar cómo está construido.

Ya se ha mencionado anteriormente que AdBlock Plus es una extensión (en este caso para Firefox), pero no todas se implementan de la misma manera, hay diferentes métodos.

Hay que decir que las extensiones en Firefox se pueden instalar en dicho navegador por medio de un archivo. Este archivo puede ser .jar o .xpi. En caso de AdBlock Plus es un archivo .xpi. Esta extensión de archivo es muy similar a .zip, por lo que podemos descomprimir el contenido.

La mayoría de archivos incluidos en la extensión están implementados usando los lenguajes XUL y JavaScript. XUL es el lenguaje **XML** para interfaces de usuario de Mozilla. También se pueden encontrar hojas de estilo, páginas **HTML** y **DTDs**. Finalmente, se encuentran archivos necesarios para la instalación de la extensión como “install.rdf” que indica al navegador como debe instalar este complemento.

Dejando a un lado el tipo de ficheros que encontramos en AdBlock Plus, se debe prestar atención a qué **módulos** usa. Veamos cuales son dichos módulos:

- Services.jsm: ofrece una amplia variedad de **métodos get**.
- XPCOMUtils.jsm: es el módulo más importante en ABP. Y, más importante que los métodos que proporciona, es que da acceso a diferentes **interfaces** que pueden ser implementadas, dando el comportamiento que deseemos a la extensión. Un ejemplo claro de esto lo encontramos en ABP, concretamente en el archivo contentPolicy.js. Aquí se implementan las funciones de diferentes interfaces. Sin embargo, a la que más conviene echar un vistazo es a la implementación de la interfaz nsIContentPolicy. Esto es de vital importancia en el funcionamiento de AdBlock Plus, porque el método

shouldLoad de esta interfaz es llamado cada vez que hay una petición y es en este método donde se decide si cargar o no el elemento solicitado por dicha petición. Se puede encontrar un completo listado con las interfaces de XPCOM en la web para desarrolladores de Mozilla (<https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XPCOM/Reference/Interface>).

- FileUtils.jsm: ofrece rutinas para tratar con ficheros.
- AddonManager.jsm: sirve para instalar, gestionar y desinstalar los complementos instalados.
- LoadContextInfo.jsm: provee una serie de métodos get para obtener información sobre el contexto de carga.
- osfile.jsm: proporciona rutinas para acceder a ficheros. Estas rutinas ayudarían a leer, escribir, renombrar, crear carpetas...
- Task.jsm: implementa una serie de métodos de task.js (<http://taskjs.org/>) que ayudan a simplificar el proceso de lanzar secuencial o **asíncronamente** las operaciones.
- CustomizableUI.jsm: permite interactuar con los elementos personalizables de la interfaz de usuario principal de Firefox (botones, barras de herramientas...)
- ctypes.jsm: permite al código de JavaScript acceder a **librerías** escritas en C.
- Public.jsm: módulo que podemos encontrar en la carpeta "lib" del paquete de ABP. Define la **API** pública de Adblock Plus.

Como se puede observar, los módulos usados coinciden en su extensión de archivo (.jsm). Esta extensión indica que son módulos programados en el lenguaje JavaScript. Todos estos módulos, excepto Public.jsm (incluido en ABP) y LoadContextInfo.jsm (<https://dxr.mozilla.org/mozilla-central/source/toolkit/modules/LoadContextInfo.jsm>), se encuentran listados en la página de Mozilla sobre módulos JavaScript (https://developer.mozilla.org/en-US/docs/Mozilla/JavaScript_code_modules).

4.3.2 Clases implementadas

En este apartado veremos las clases necesarias para el funcionamiento de la extensión. Para identificar cada una de estas clases se usará el nombre que se utiliza en ABP. Hay clases que están relacionadas directamente con la interfaz de usuario y, en ese caso, se indicará en la descripción de la clase. La lista de clases es la siguiente (por orden alfabético):

- AboutHandler:
Es la implementación de la interfaz de XPCOM nsIAboutModule. Este módulo permite añadir una página ‘about:’. Estas páginas muestran información. Posiblemente la más conocida es la página ‘about: blank’ que no muestra nada. Sin embargo, este tipo de páginas se pueden usar para ver información como la versión del navegador (‘about:’) o las extensiones o complementos que tenemos instalados (‘about: plugins’). En este caso se utilizará para mostrar el número de coincidencias con los filtros de ocultación de elementos.
- ActiveFilter:
Clase abstracta que extiende la clase base Filter y que será usada para representar filtros activos.
- AdblockPlus:
Clase que implementa la API pública de AdBlock Plus.
- Backup (interfaz de usuario):
Esta clase contiene la implementación de la funcionalidad para hacer copias de seguridad y recuperación de filtros. Esta funcionalidad es usada en la siguiente interfaz:

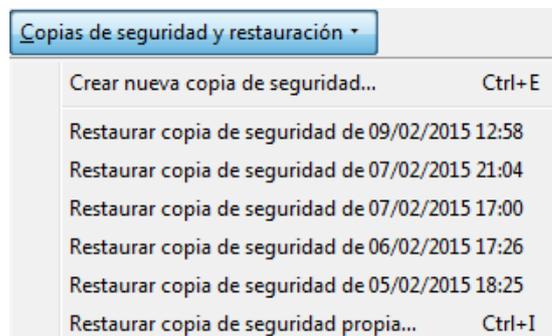


Ilustración 14: Captura de pantalla de la interfaz de AdBlock Plus (recuperación y copias de seguridad)

- BlockingFilter:
Clase que extiende RegExpFilter y que representa los filtros de bloqueo.
- Cache:
Clase que implementa una memoria caché en la que los datos nuevos reemplazan a los que han sido guardados en primer lugar.
- CombinedMatcher:
En esta clase se combinan el buscador de coincidencias de filtros de bloqueo y el buscador de coincidencias de filtros de excepción.

- CommentFilter:
Clase que extiende la clase base Filter y que representará los comentarios.
- Downloadable:
Esta clase implementa un objeto que puede ser descargado.
- DownloadableSubscription:
Clase que extiende la clase abstracta RegularSubscription y que representa las suscripciones de filtros actualizadas por AdBlock Plus.
- Downloader:
Se encarga de todas las tareas relacionadas con la descarga de objetos desde cierta URL.
- ElemHide:
Clase que implementa la funcionalidad de ocultación de elementos.
- ElemHideBase:
Clase que extiende la clase ActiveFilter y que representará la clase base para los filtros de ocultación de elementos. Será extendida para representar los filtros de excepción de ocultación y los de ocultación.
- ElemHideException:
Clase que extiende la clase ElemHideBase y que representará la clase para los filtros de excepción de ocultación de elementos.
- ElemHideFilter:
Clase que extiende la clase ElemHideBase y que representará la clase para los filtros de ocultación de elementos.
- ExternalSubscription:
Clase que extiende la clase abstracta RegularSubscription y que representará las suscripciones de filtros actualizadas por medio de otras extensiones o plugins.
- HitRegistrationChannel:
Contiene toda la información sobre el canal por el que se recibirán los datos de las coincidencias con los filtros, así como las funciones necesarias para manejar dicho canal.
- Filter:
Clase abstracta base que representará a un filtro. Será extendida por otras clases dependiendo del filtro a representar.

- FilterActions (interfaz de usuario):

Es una clase que implementa diversas acciones que se realizan a los filtros en la interfaz gráfica. Estas acciones son las que se permiten realizar en el menú de preferencias de los filtros:

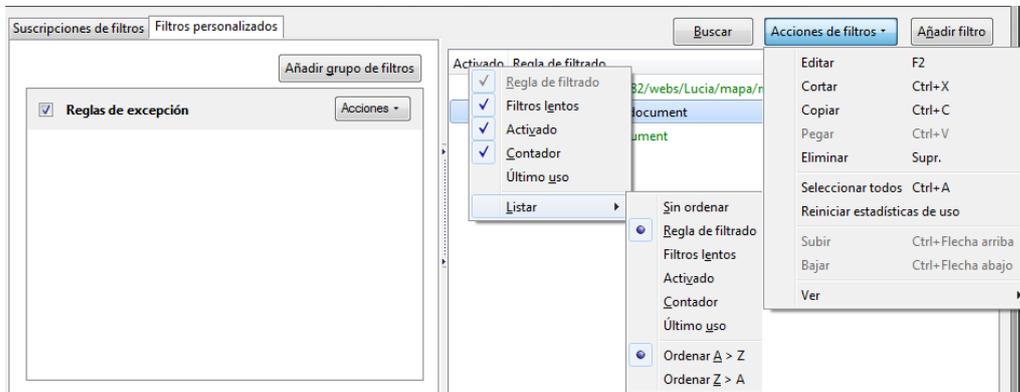


Ilustración 15: Montaje de capturas de pantalla de la interfaz de Adblock Plus (acciones de filtros)

- FilterListener:

Contiene las funciones necesarias para cambiar las propiedades de los listener que controlan los cambios en los filtros.

- FilterNotifier:

Esta clase se encarga de registrar y eliminar listener, así como de notificarles si ha ocurrido un evento.

- FilterSearch (interfaz de usuario):

Contiene la implementación de la funcionalidad de búsqueda de filtros de la ventana de preferencias de filtros. En la siguiente ilustración se muestran resaltados los elementos en los que se usa dicha funcionalidad:

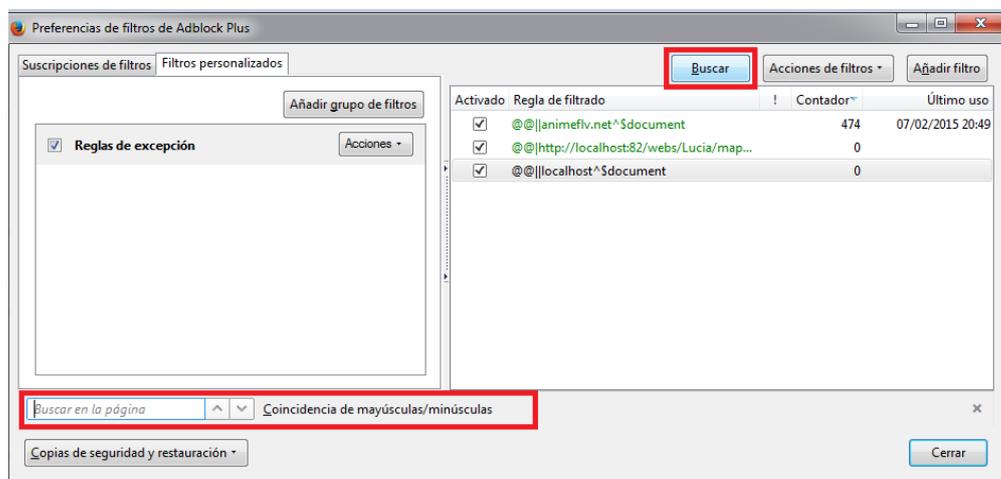
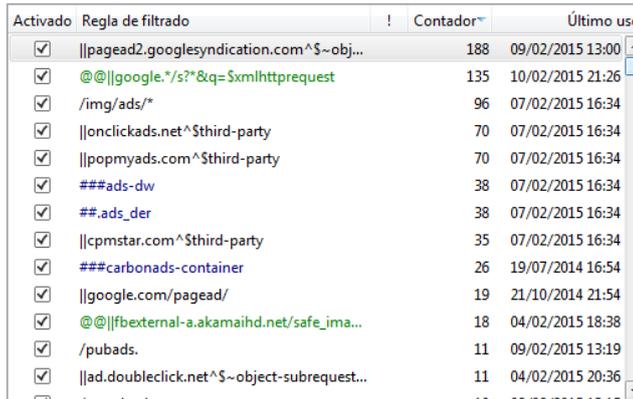


Ilustración 16: Captura de pantalla de la interfaz de ABP (búsqueda de filtros)

- FilterStorage:
Esta clase tiene las funciones necesarias para leer los filtros del usuario, gestionarlos y recuperarlos. Básicamente actúa como almacén de filtros.
- FilterView (interfaz de usuario):
Contiene la implementación de la interfaz nsITreeView (sirve para obtener y mostrar información de un árbol), necesaria para visualizar los filtros de una suscripción. Esta clase está relacionada con la ventana de preferencias de filtros de ABP.



Activado	Regla de filtrado	Contador	Último uso
<input checked="" type="checkbox"/>	pagead2.googlesyndication.com^\$~obj...	188	09/02/2015 13:00
<input checked="" type="checkbox"/>	@ google.*s?*&q=\$xmlhttprequest	135	10/02/2015 21:26
<input checked="" type="checkbox"/>	/img/ads/*	96	07/02/2015 16:34
<input checked="" type="checkbox"/>	onclickads.net^\$third-party	70	07/02/2015 16:34
<input checked="" type="checkbox"/>	popmyads.com^\$third-party	70	07/02/2015 16:34
<input checked="" type="checkbox"/>	##ads-dw	38	07/02/2015 16:34
<input checked="" type="checkbox"/>	##ads_der	38	07/02/2015 16:34
<input checked="" type="checkbox"/>	cpmstar.com^\$third-party	35	07/02/2015 16:34
<input checked="" type="checkbox"/>	##carbonads-container	26	19/07/2014 16:54
<input checked="" type="checkbox"/>	google.com/pagead/	19	21/10/2014 21:54
<input checked="" type="checkbox"/>	@ fbexternal-a.akamaihd.net/safe_ima...	18	04/02/2015 18:38
<input checked="" type="checkbox"/>	/pubads.	11	09/02/2015 13:19
<input checked="" type="checkbox"/>	ad.doubleclick.net^\$~object-subrequest...	11	04/02/2015 20:36

Ilustración 17: Captura de pantalla de la interfaz de ABP (vista de los filtros)

- HistoryPurgeObserver:
Contiene la implementación de un observer que controla si se realizan acciones de borrado de historial.
- INIParser:
Esta clase contiene una función que es llamada cada vez que se lee un fichero para analizar los datos de los filtros contenidos en dicho fichero.
- InvalidFilter:
Clase que extiende la clase base Filter y que representa a los filtros que no se consideran válidos.
- KeySelector:
Esta clase provee una serie de funciones para poder buscar y usar los atajos de teclado disponibles.
- ListManager (interfaz de usuario):
Es la clase que se encarga de gestionar los grupos de filtros y de mantenerlos actualizados en la interfaz gráfica. Esta clase es muy similar a la de FilterView, de hecho, un nombre más adecuado habría sido SubscriptionView.

- Matcher:
 Contiene las diferentes funciones para la búsqueda de coincidencias con los filtros. Se creará una instancia de esta clase para filtros de bloqueo y otra para filtros de excepciones.
- Notification:
 Se encargará de leer las notificaciones y decidir cuáles deben mostrarse.
- objTabs:
 Es la clase encargada de mostrar y ocultar las pestañas en los objetos cargados por plugins externos como Java y Flash. Estas pestañas tienen el texto “Block” y permiten bloquear estos objetos. Podemos ver un ejemplo aquí:



Ilustración 18: imagen de adblockplus.org que muestra el comportamiento de las pestañas en objetos.

- optionsObserver:
 Contiene las funciones necesarias para inicializar las opciones de la extensión y controlar cuales se han abierto.
- Policy:
 Contiene una serie de funciones de comprobación de la **política de contenidos** pública y un conjunto de objetos auxiliares.
- PolicyImplementation:
 Implementación de las interfaces de XPCOM nsIContentPolicy y nsIChannelEventSink. La primera de estas interfaces es la que controla los mecanismos de la política de contenido, es decir, se encarga de decidir si cargar o no algo. En cambio, la segunda interfaz sirve para obtener control sobre los eventos en canales, como por ejemplo la redirección.
- PrivateBrowsing:
 Tiene una serie de funciones para saber si está activado el modo de privacidad en el navegador.
- RegExpFilter:
 Clase abstracta que extiende la clase ActiveFilter y que representará los filtros basados en expresiones regulares.

- **RegularSubscription:**
Clase abstracta que extiende la clase Subscription y que representará las suscripciones normales. Será extendida para representar las suscripciones que se actualizan por medio de otras extensiones y las que mantiene actualizadas ABP.
- **SelectSubscription (interfaz de usuario):**
Contiene los métodos llamados cuando se elige y añade una nueva suscripción de filtros. Implementa la funcionalidad de la siguiente ventana:

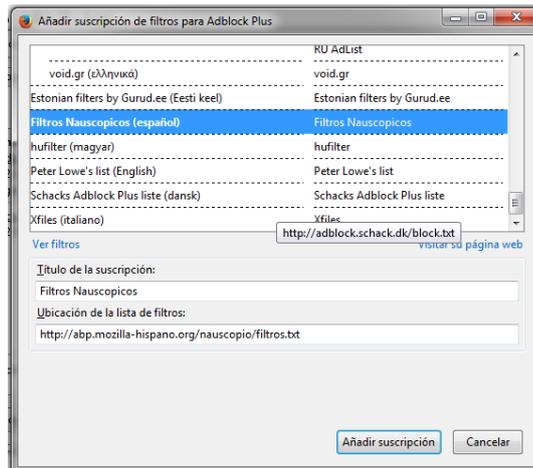


Ilustración 19: Captura de pantalla de la interfaz de ABP (selección y adición de una suscripción).

- **SpecialSubscription:**
Clase que extiende la clase abstracta Subscription y que representará las suscripciones especiales, es decir, las que contienen los filtros definidos por el usuario.
- **Subscription:**
Clase abstracta básica para representar las suscripciones de filtros. Será extendida por otras clases dependiendo de la suscripción a representar.
- **SubscriptionActions (interfaz de usuario):**
Contiene la implementación de las diferentes acciones que se pueden aplicar a las suscripciones de los filtros en la interfaz gráfica. Va muy en la línea de lo que hacía la clase FilterActions. Esta implementación tiene que ver con el menú de la siguiente ilustración.

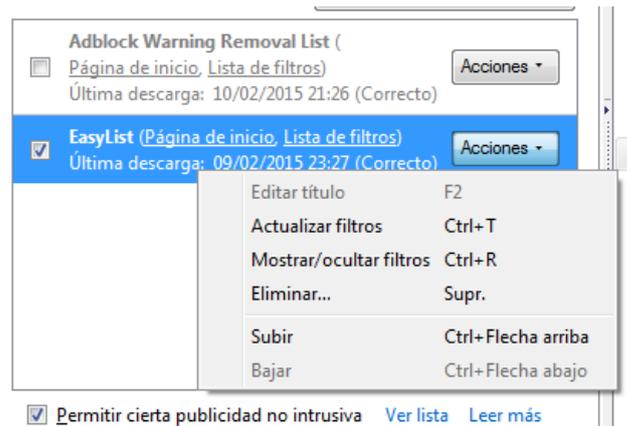


Ilustración 20: Captura de pantalla de la interfaz de ABP (acciones en suscripciones)

- Synchronizer:
Este objeto es el responsable de descargar las suscripciones a filtros cada vez que es necesario.
- SyncServiceObserver:
Contiene funciones para detectar las notificaciones del servicio de **Firefox Sync** y mantener actualizados los datos sobre el servicio.
- Templater (interfaz de usuario):
Contiene funciones para el procesamiento de plantillas. Sirve para poder usar nodos plantilla para introducir nuevos datos.
- TitleEditor (interfaz de usuario):
Contiene la implementación de la funcionalidad de edición de títulos de suscripciones.
- Utils:
Contiene una gran variedad de funciones que son útiles para el correcto funcionamiento de Adblock Plus.
- WindowObserver:
Esta clase implementa un observer que reacciona cada vez que se abre una nueva ventana o se desactiva ABP. El objetivo es mantener actualizada la información de la ventana.
- WhitelistFilter:
Clase que extiende la clase `RegExpFilter` que representa los filtros de lista blanca, es decir, de no bloqueo.

Vista la lista de clases que nos encontramos en AdBlock Plus, vamos a ver una serie de ilustraciones que ayudarán a entender como están construidas algunas de estas clases. En concreto, vamos a ver un par de árboles que muestran cómo están pensadas las representaciones de los filtros y las subscripciones. La razón de hacer esto es que, dado que la lista está en orden alfabético, es posible que quede algo confuso el tema de cómo se van extendiendo estas clases, por eso es necesario ver de una manera más gráfica esto.

Primero vamos a centrarnos en las clases que se encargan de representar los diferentes filtros. Como se puede leer en la lista de filtros, hay una clase base que se va extendiendo para satisfacer las necesidades de cada tipo de filtro. El árbol quedaría de la siguiente manera:

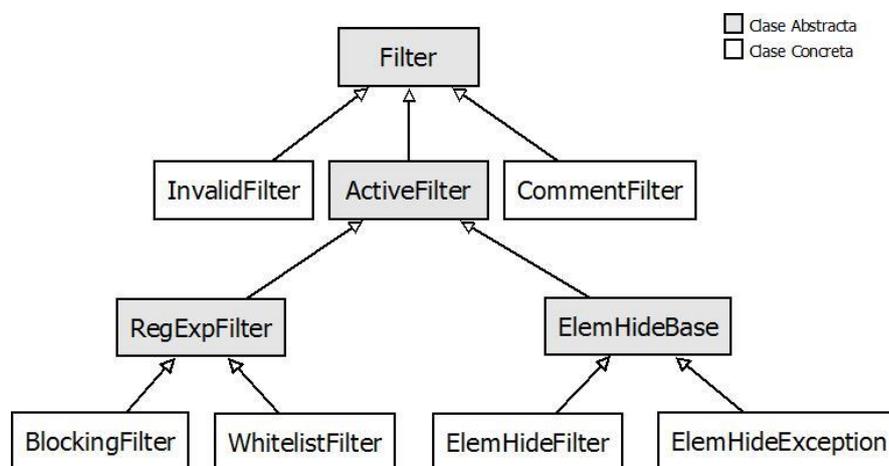


Ilustración 21: Árbol realizado con el programa Dia que muestra la estructura de clases de los filtros.

En segundo lugar, vamos a prestar atención a las clases que representan las subscripciones. Igual que en el caso anterior, existe una clase base que se va extendiendo para representar los diferentes tipos de subscripciones. Aquí vemos como queda el árbol:

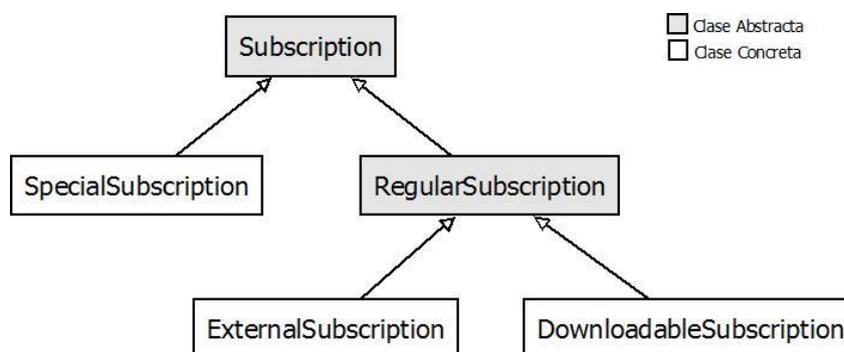


Ilustración 22: Árbol realizado con el programa Dia que muestra la estructura de clases de las subscripciones.

4.3.3 ¿Cómo actúa Adblock Plus?

Para ver los diferentes pasos que sigue Adblock Plus en ejecución normal empezaremos por los más generales y poco a poco iremos desmenuzando cada uno de los pasos hasta llegar a algo suficientemente “profundo”. Nos centraremos en los más interesantes para el funcionamiento de Adblock Plus, es decir, los que se esbozan al final del apartado 4.2. Junto con la explicación irá un diagrama para entender más visualmente la interacción entre las distintas clases. En los diagramas se verán dos clases que no se han mencionado en el apartado anterior. Estas clases son “Services” y “Components”, que proporcionan una serie de utilidades a las herramientas diseñadas para Firefox y que no son de uso único por Adblock Plus.

Empezaremos por las actividades relacionadas con el inicio de ABP.

- Iniciar Adblock Plus:

Esto es lo que sucede cuando se inicia la extensión. El objetivo de esta acción es dejar listo ABP para empezar a funcionar. Los pasos subrayados serán desglosados más adelante. Las acciones realizadas son las siguientes:

1. Carga las funciones para registrar las actividades realizadas para **debug**. Se usa para medir los tiempos de carga.
2. Registra la API pública. Esto se debe hacer para que otras extensiones puedan hacer uso de las funciones de Adblock Plus.
3. Inicializa el listener de los filtros. Sirve para sincronizar el almacenamiento de los filtros con el comparador mediante una serie de funciones que se ejecutan al suceder algunos eventos.
4. Inicializa la política de contenido. Será la piedra angular de la extensión, es decir, es la que se encargará de decidir si bloquear o no los elementos.
5. Inicializa el sincronizador. Hará algo similar que el listener de filtros, pero lo que mantendrá sincronizado es la suscripción a los filtros.
6. Inicializa el notificador. Se encargará de gestionar las notificaciones.
7. Inicializa la integración con Firefox Sync.
8. Inicializa la interfaz de usuario. Esto quiere decir que se preparará todo lo que deba aparecer en la ventana del navegador que tenga referencia con Adblock Plus para que el usuario pueda verlo e interactuar.

- Inicializar el listener de los filtros:
 1. Se cargan todos los módulos que se vayan a usar.
 2. Se declaran e inicializan algunas clases y **variables**.
 3. Se ejecuta una función inicializadora que realiza lo siguiente:
 - I. Añade un listener que se encargará de ejecutar las acciones necesarias cuando haya cambios en los filtros o subscripciones.
 - II. Si se permite a extensiones añadir hojas de estilo, entonces se ejecuta la función inicializadora de la clase “ElemHide”, que programa un evento para que cuando cambien las preferencias de ABP se aplique la hoja de estilo de usuario de Adblock Plus de ocultación de elementos. También se programa otro evento para desaplicar la hoja de estilo en caso de que la extensión se cierre.
 - III. Se cargan las subscripciones del disco y se guardan en el almacenador de filtros.
 - IV. Se añade un observador (para cuando se borra el historial) y se programa un evento para que se elimine el observador cuando se cierre la aplicación.

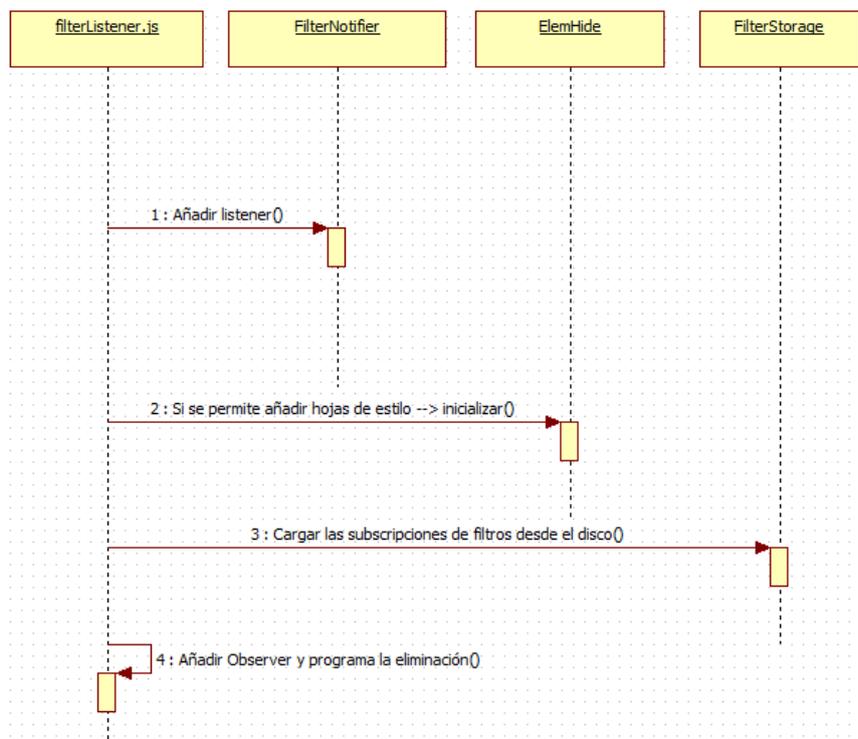


Ilustración 23: Diagrama de interacción entre clases de la función inicializadora del listener de filtros

- Inicializar la política de contenidos:
 1. Se cargan todos los módulos que se van a usar.
 2. Se declaran e inicializan algunas clases y variables.
 3. Se ejecuta la función inicializadora de la clase “Policy” que inicializa algunos tipos de objeto y **propiedades exportadas** y se aplica la hoja de estilo de usuario para ocultar elementos.

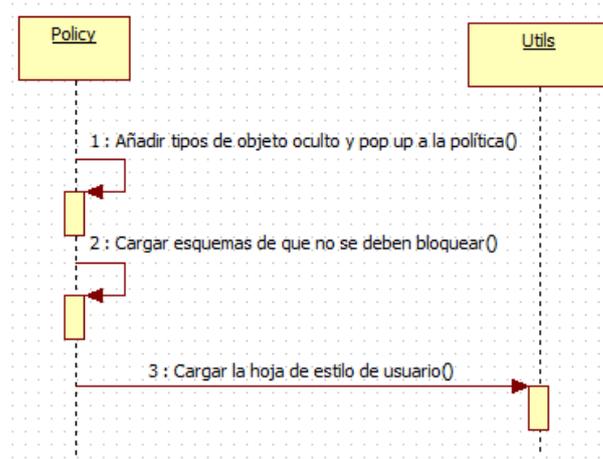


Ilustración 24: Diagrama de interacción entre clases de la función inicializadora de la clase Policy

4. Se ejecuta la función inicializadora de la clase “PolicyImplementation” que registra la política. Realiza las siguientes acciones:
 - I. Se intenta registrar la política. Si no se vuelve a lanzar asincrónicamente la función.
 - II. Se crea una instancia del gestor de **categorías**.
 - III. Se guardan las categorías en el gestor de categorías.
 - IV. Se comprueba la versión del navegador para adaptarse.
 - V. Se añaden cuatro observadores para que controlen las siguientes situaciones:
 - i. Cuando se hace una petición http.
 - ii. Cuando se crea una ventana de contenido web.
 - iii. Cuando se elimina una entrada de una categoría en el administrador de categorías.
 - iv. Cuando se elimina una categoría en el administrador de categorías.
 - VI. Se define el evento para eliminar los observers, borrar las categorías del gestor y eliminar el registro de la política asincrónicamente.

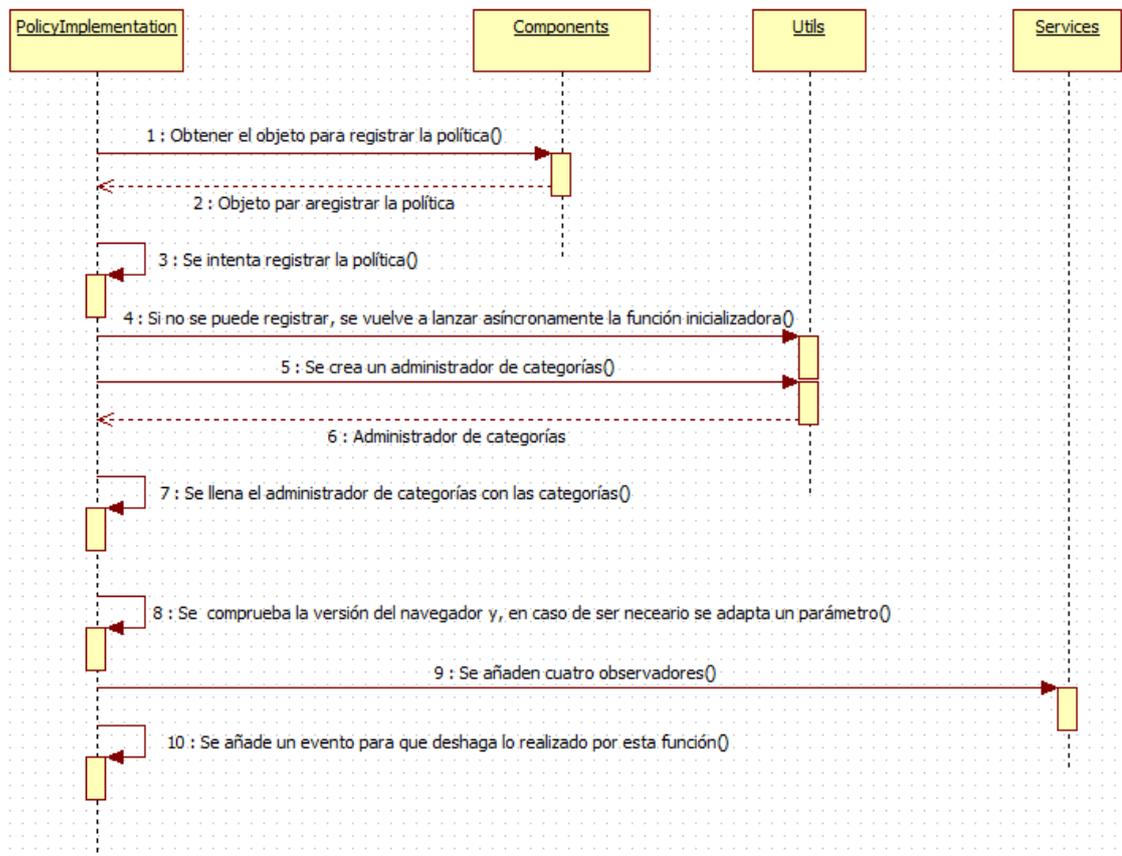


Ilustración 25: Diagrama de interacción entre clases de la función inicializadora de la clase *PolicyImplementation*

- **Inicializa el sincronizador:**
 1. Se cargan todos los módulos que se van a usar.
 2. Se declaran e inicializan algunas clases y variables.
 3. Se ejecuta la función inicializadora de la clase “Synchronizer” que hace lo siguiente:
 - I. Se crea una instancia de la clase “Downloader”.
 - II. Se crea un evento para cuando se cierra la aplicación que cancela el downloader.
 - III. Se asignan eventos para:
 - i. Cuando empieza una descarga.
 - ii. Cuando acaba una descarga con éxito.
 - iii. Cuando una descarga da error.
 - iv. Cuando se cambian los intervalos de excepción.

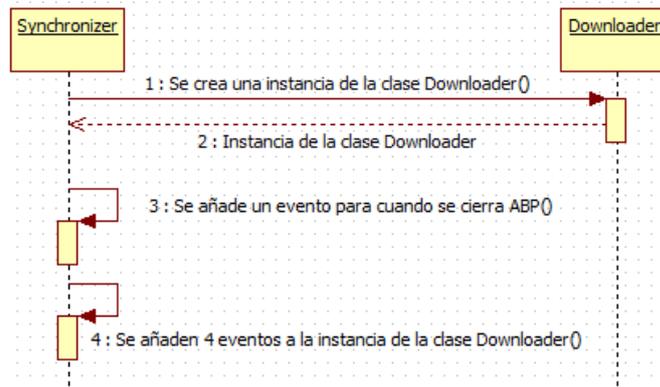


Ilustración 26: Diagrama de secuencia de la función inicializadora de la clase Synchronizer

- Inicializa el notificador:
 1. Se cargan todos los módulos que se van a utilizar.
 2. Se declaran e inicializan algunas clases y variables.
 3. Se ejecuta la función inicializadora de la clase “Notification” que hace lo siguiente:
 - I. Se crea una instancia de la clase Downloader.
 - II. Se crea un evento para cuando se cierra la aplicación que cancela el downloader.
 - III. Se asignan eventos al objeto Downloader para:
 - i. Cuando acaba una descarga con éxito.
 - ii. Cuando una descarga da error.
 - iii. Cuando se cambian los intervalos de excepción.

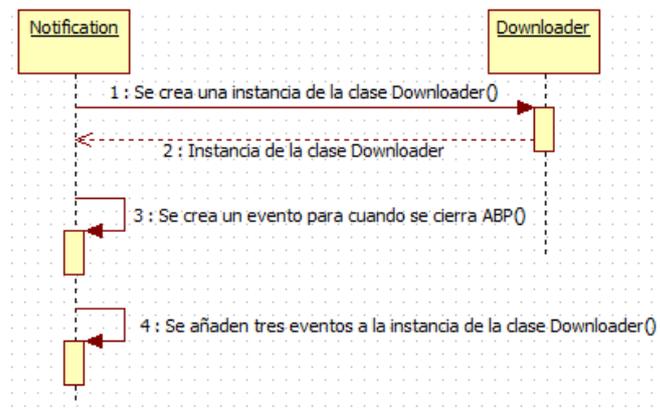


Ilustración 27: Diagrama de secuencia de la función inicializadora de la clase Notification

- Inicializa la integración con Firefox Sync:
 1. Se cargan todos los módulos que se van a usar.
 2. Se declaran e inicializan algunas clases y variables.
 3. Se ejecuta la función inicializadora de la clase “SyncServiceObserver”.

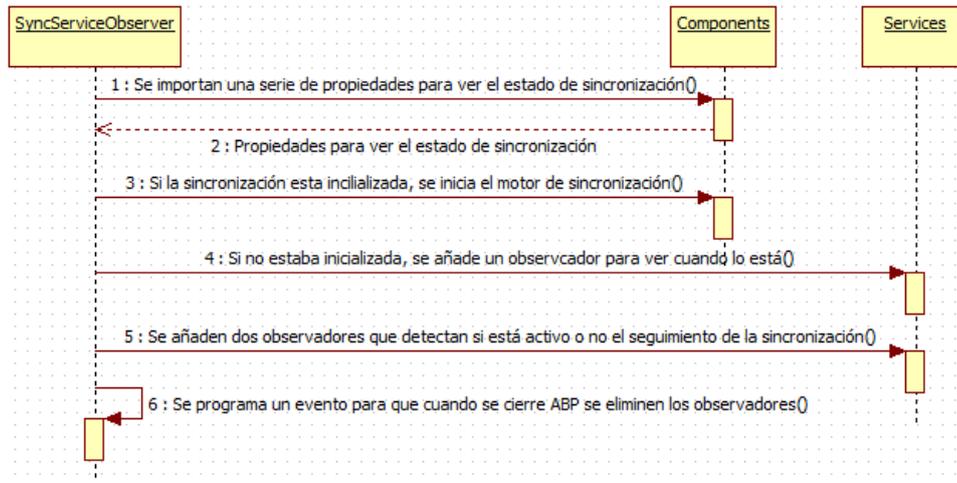


Ilustración 28: Diagrama de secuencia de la función inicializadora de la clase SyncServiceObserver

- Inicializa la interfaz de usuario:
 1. Se cargan todos los módulos que se van a usar.
 2. Se declaran e inicializan algunas clases y variables.
 3. Se ejecuta la función inicializadora de la clase “optionsObserver” que hace lo siguiente:
 - I. Se añade un observer para cada vez que se carga la interfaz de usuario.
 - II. Se programa un evento para que cuando se cierre la aplicación se elimine el observer.

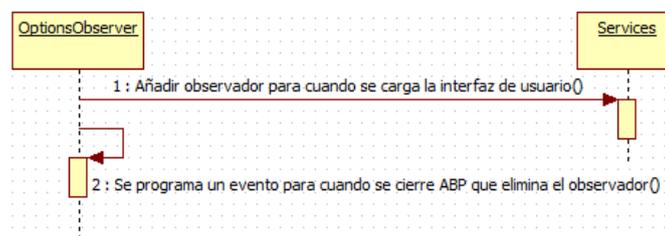


Ilustración 29: Diagrama de secuencia de la función inicializadora de la clase optionsObserver

4. Se declara e inicializa otro conjunto de clases y variables.
5. Se ejecuta la función inicializadora de la clase “UP” que hace lo siguiente:
 - I. Se empieza a cargar la superposición. La superposición está compuesta por archivos escritos en lenguaje XUL que describen los elementos extra que aparecerán en la interfaz de usuario. Estos elementos serán los menús e iconos de ABP.
 - II. Se añade un evento para cuando se carga que, cuando se active, hará lo siguiente:
 - i. Cerrar si se ha cerrado la aplicación.

- ii. Procesar la superposición.
 - iii. Añadir el botón a la barra de herramientas.
 - iv. Si se han cargado la superposición y los filtros. Se cargan todos los elementos de Adblock Plus a la interfaz.
- III. Se espera a que los filtros se carguen.
- IV. Inicializa la interfaz. Además se añade un observador para cuando se recupera la ventana
- V. Si se han cargado la superposición y los filtros. Se cargan todos los elementos de Adblock Plus a la interfaz.

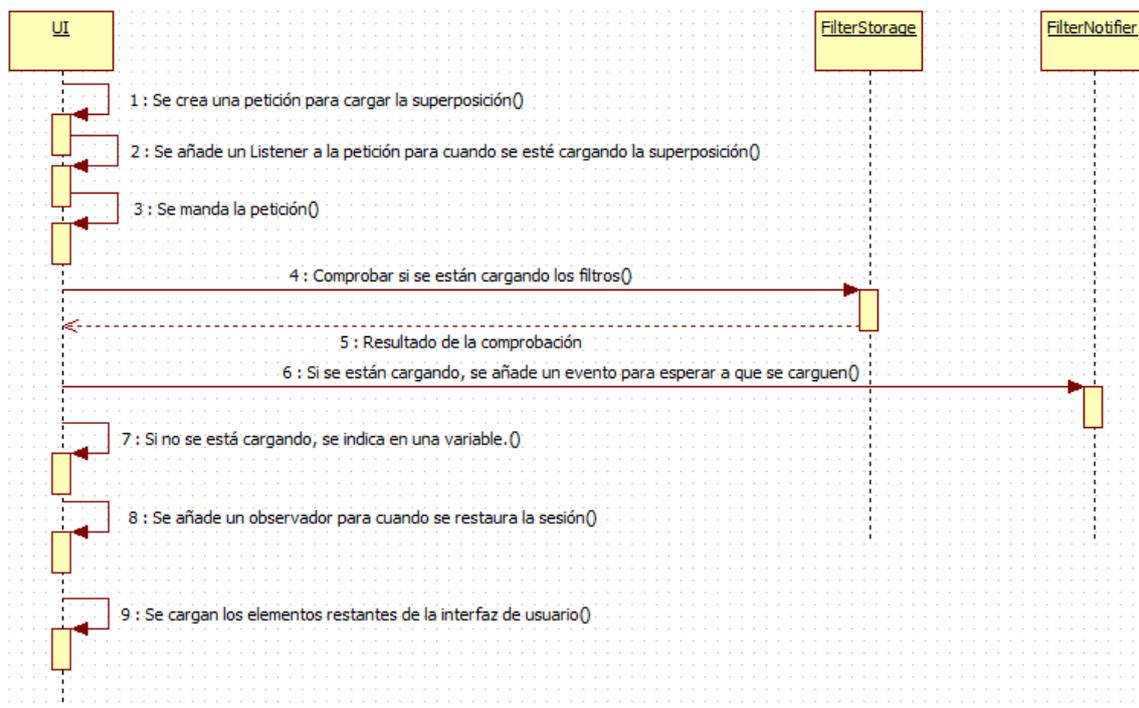


Ilustración 30: Diagrama de secuencia de la función inicializadora de la clase UI

Hasta aquí llegarían los pasos referentes a la inicialización de la extensión. Los siguientes puntos harán referencia al funcionamiento de la extensión cuando se carga una página y sus elementos, es decir, cuando bloquea las cosas Adblock Plus. La manera en la que se han obtenido estos pasos ha sido poniendo numerosos **puntos de ruptura** en el depurador de extensiones de Firefox.

- Tratamiento de una nueva página abierta:

Cuando se abre una nueva pestaña o se abre una nueva ventana, el observador definido en la inicialización reacciona produciendo que se realicen las siguientes acciones:

1. Se comprueba si la página nueva tiene contenido. En caso de no tenerlo, se salta el procesamiento de la página. (FIN)
2. Se obtiene la **URI** de la página.
3. Se procesa la página para decidir si hay que bloquearla o no. Si al procesar la página se detecta que debe ser bloqueada, inmediatamente es cerrada. Este paso se explicará más en profundidad en el punto procesamiento de un nodo (las páginas también son nodos).
4. Si la página abierta es del tipo “about: blank” se comprueba si se espera algún Pop-Up. En caso afirmativo, se señala que así es para procesarlo cuando se haga la petición http.
5. Finalmente se analiza la petición HTTP y se corrigen posibles errores. Además en caso de esperarse un Pop-Up se añade un observador para cuando este se habrá. Cuando esto ocurra se empezará de nuevo el proceso desde el punto uno de este caso de uso.

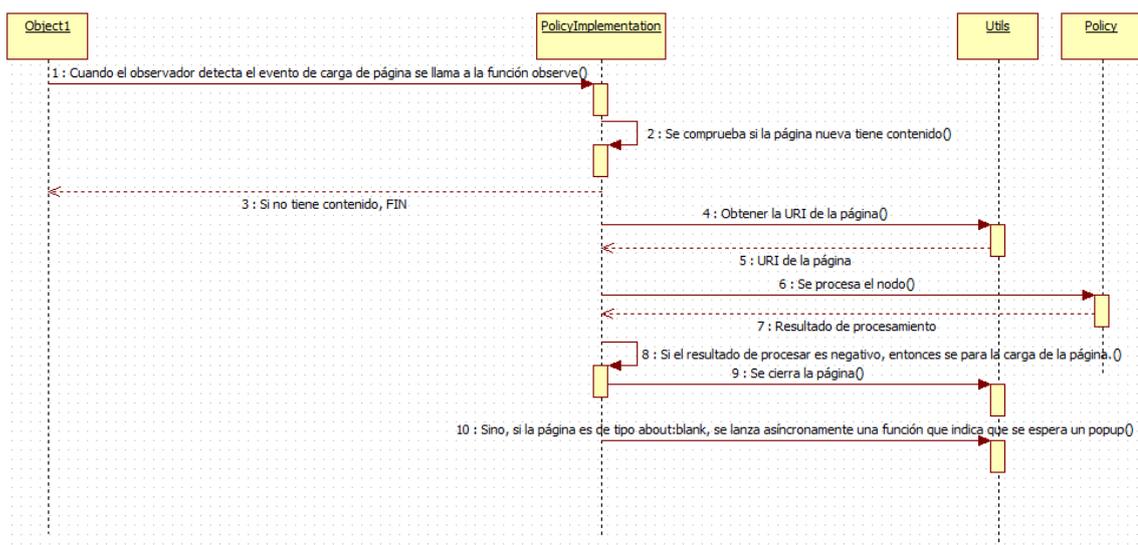


Ilustración 31: Diagrama de secuencia del procesamiento de una página nueva

- Comprobación de si un elemento o página debe cargarse:

Esta comprobación se realiza cada vez que se intenta cargar un elemento en una zona de contenido. Que se produzca esta reacción es gracias a haber implementado en una clase (PolicyImplementation) la interfaz nsIContentPolicy. Esta interfaz contiene una función llamada shouldLoad que está configurada para ser llamada justo antes de cargar un elemento en la página. En esta función, se realiza lo siguiente:

1. Si es una petición sin contexto o lo que se pide es un documento de alto nivel (una página), se acepta la petición. De esta manera las páginas se procesarán en

el último caso descrito (tratamiento de una nueva página abierta) y serán cerradas en caso de ser necesario. (FIN)

2. Si es un elemento independiente, se acepta la petición. (FIN)
3. Si el elemento solicitado no tiene ventana, se acepta la petición. (FIN)
4. Si la petición se encuentra en la lista de excepciones, se acepta. (FIN)
5. Si el tipo de la petición es desconocida se interpreta como de tipo “otro”.
6. Se procesa el elemento a cargar. Esto se explicará con más detenimiento en el caso de uso procesamiento de un nodo.
7. Si el procesamiento del nodo es positivo (elemento permitido) se guarda la petición por si hay redirección.
8. Dependiendo del resultado del procesamiento del nodo, se bloquea o se acepta la petición.

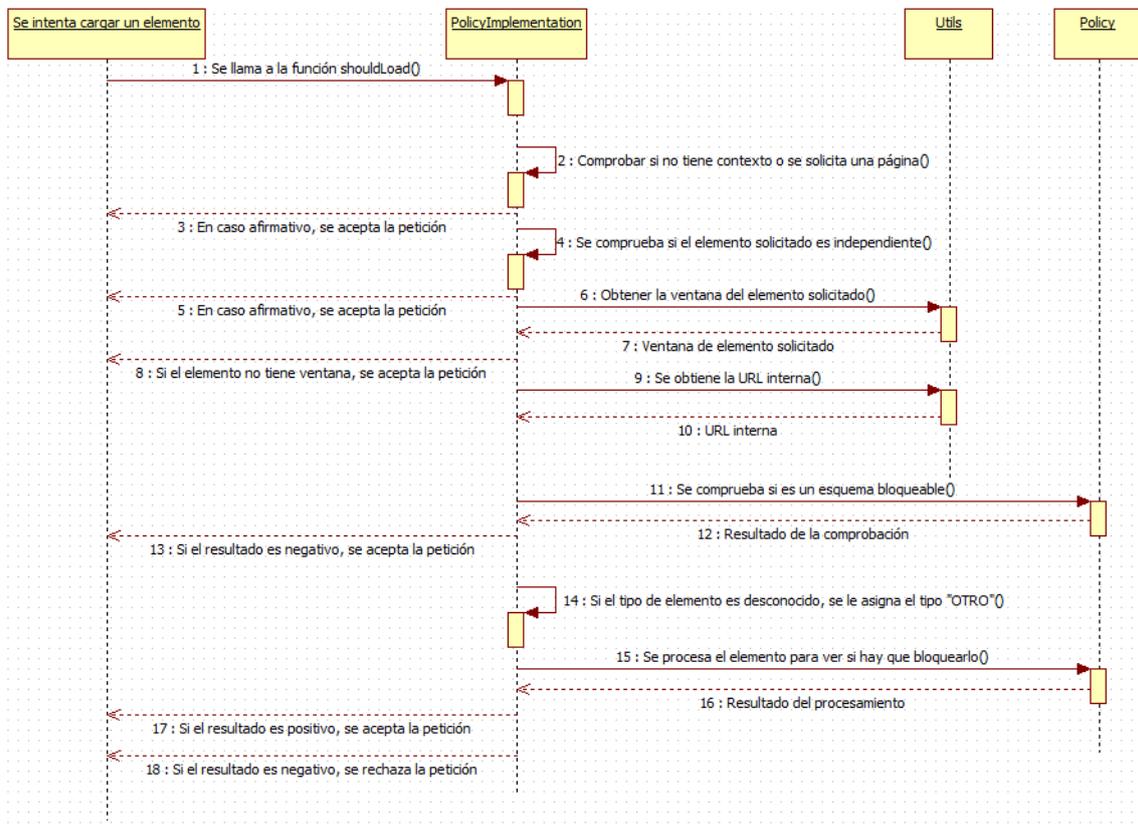


Ilustración 32: Diagrama de secuencia de la comprobación de si se debe cargar un elemento

- Procesamiento de un nodo:

Esta operación se realiza siempre que haya que comprobar si un elemento o página es adecuado con los términos de Adblock Plus. Digamos que aquí es donde se decide si bloquear o no, por lo que se puede decir que en esta operación se encuentra la funcionalidad de ABP más importante. Los pasos que se siguen son los siguientes:

1. Si es un nodo que no pertenece a ninguna ventana, o no procede de ninguna localización (no tiene un enlace asociado), entonces se permite el elemento. (FIN)
2. Se obtiene la ventana del nodo, la localización y donde está alojado.
3. Se comprueba el nodo y sus padres para ver si están en la lista de excepciones, en caso afirmativo, se indica que el nodo puede ser cargado. (FIN)
4. Si en el paso anterior, no se han encontrado coincidencias con los filtros de excepciones, entonces se continúa con el procesamiento del nodo. Se hacen una serie de comprobaciones y ajustes, para poder seguir con el procesamiento. Estas comprobaciones están relacionadas con el tipo del nodo.
5. Si es un elemento oculto, se realizan estos pasos:
 - I. Se repite el proceso del paso 3. (FIN)
 - II. Se comprueba si los filtros están activados en el dominio. En caso negativo, se permite el elemento. (FIN)
 - III. Se comprueba que no haya una excepción para el elemento. Si hay una excepción, se permite. (FIN)
6. Se comprueba si es un elemento de terceros, es decir que no es propio de la web que se está cargando.
7. Se comprueba si está en la lista de filtros de bloqueo y, si es necesario, se programa el nodo para ser ocultado.
8. Se guarda la información del nodo.
9. Se devuelve el resultado del procesamiento. Si es positivo, el elemento podrá cargarse, si no lo es, deberá ser bloqueado.

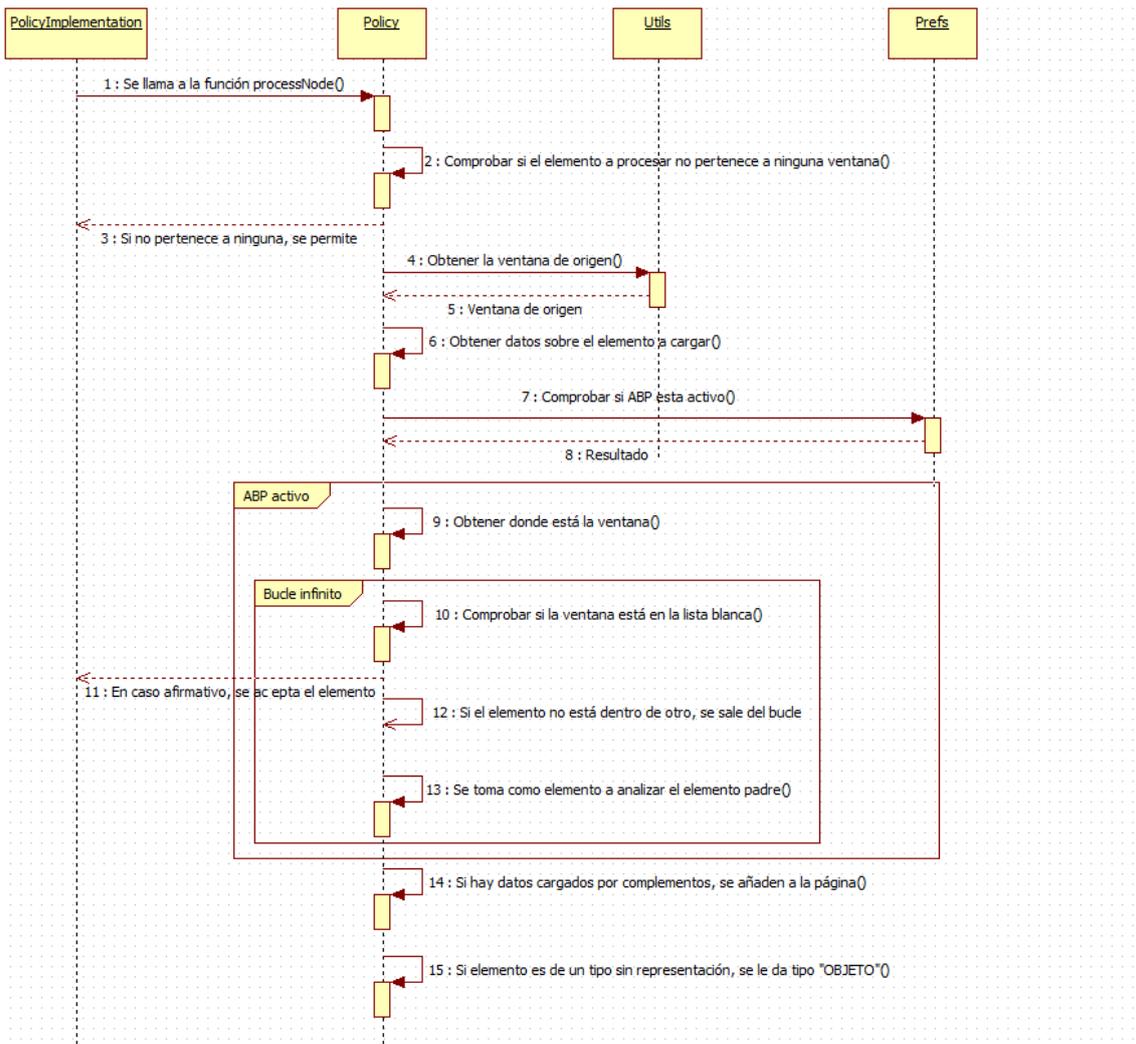


Ilustración 33: Diagrama de secuencia del procesamiento de un elemento (parte 1)

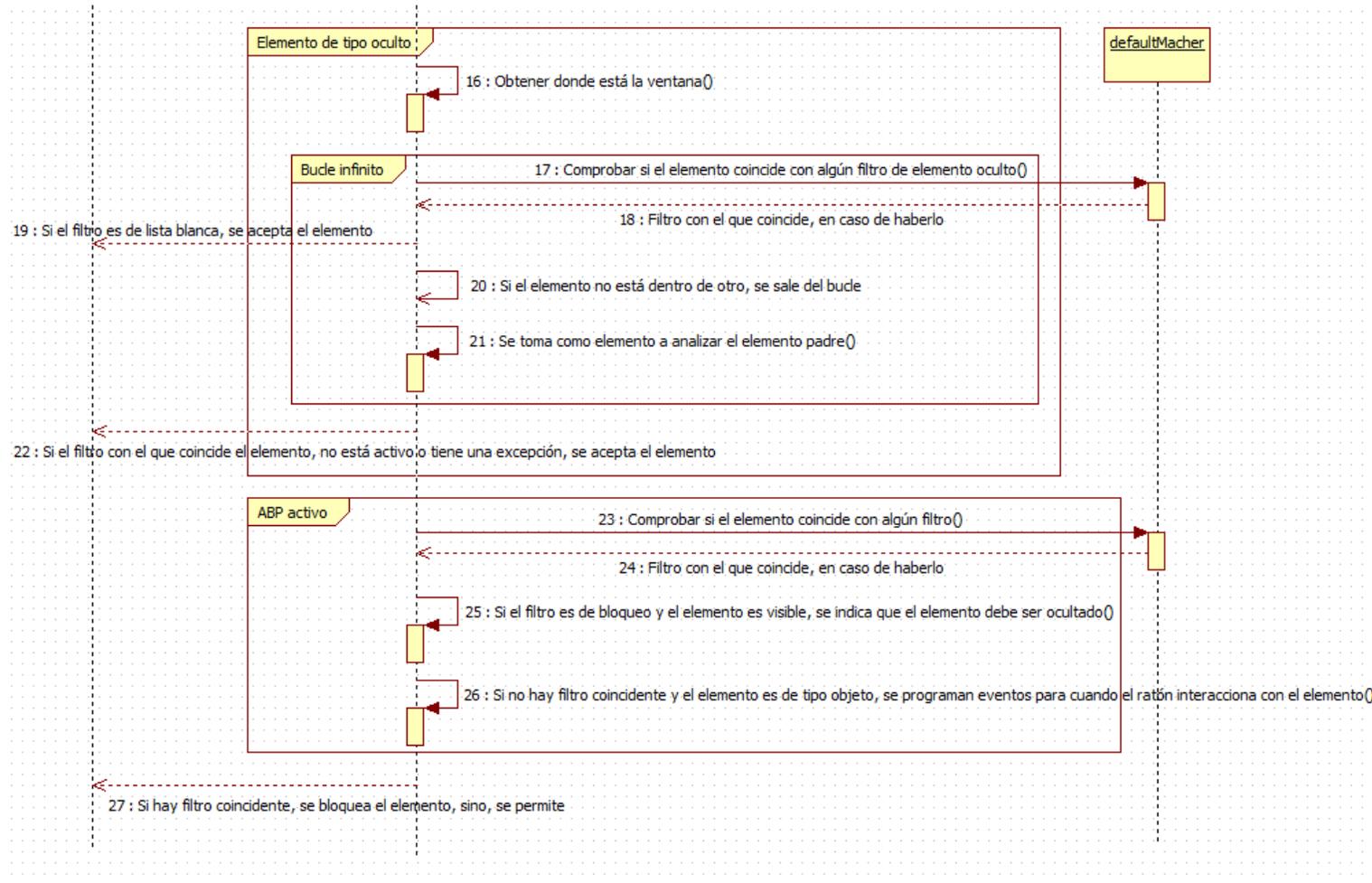


Ilustración 34: Diagrama de secuencia del procesamiento de un elemento (parte 2)

5 Soluciones al problema del adblocking

En este apartado veremos una recopilación de las posibles soluciones para este problema. Algunas de ellas ya han sido esbozadas en apartados anteriores, pero las veremos con más profundidad en esta sección.

5.1 Concienciar a los usuarios

Hoy en día los usuarios están cansados de la publicidad online, producto lógico de las agresivas campañas publicitarias que hay. Por esto, será una tarea ardua el convencer a los usuarios de lo necesario que es que permitan tener publicidad acompañándoles mientras navegan.

Para ello hay que ganarse la confianza de los usuarios actuando de manera correcta con respecto al uso de publicidad. No se puede pretender convencer al usuario de ver publicidad no adecuada para mantener la web a flote. Por esto, lo primero es cerciorarse de que la publicidad de la web es adecuada.

Habrán muchos usuarios que no utilicen herramientas de bloqueo de anuncios y que se sentirán bastante satisfechos con solo este cambio. Sin embargo, lo difícil es convencer a los que las utilizan de añadir la web a la lista de excepciones o de dejar de usarlas. Aquí es donde entran estrategias como sustituir la publicidad por imágenes en las que se muestra un mensaje para intentar llegar a estos usuarios y convencerles. Existen dos opciones, sustituir la publicidad en cualquier caso por estas imágenes o mensajes o, detectar si se está usando una herramienta de bloqueo de publicidad y, en caso afirmativo, sustituir la publicidad por imágenes que permitan dichas herramientas. Hay numerosos ejemplos de webs que ya aplican estas estrategias. Este es uno:



Ilustración 35: Imagen extraída de vidaextra.com que sustituye los anuncios cuando se usa ABP

Una vez conseguido que los usuarios vuelvan a confiar en la web, no hay que traicionar esta confianza, porque cualquier intento por volver a colocar publicidad molesta o no adecuada se traducirá en perder de nuevo a estos usuarios y, como consecuencia final, dejar de percibir ingresos mediante la publicidad.

5.2 Dejar de ofrecer contenido a los usuarios de bloqueadores

Esta es una opción más arriesgada y no se ve como una solución acertada por diversas comunidades. Sin embargo, al existir como opción vamos a ver en qué consistiría.

Esta solución es bastante drástica ya que conllevaría la muy posible pérdida de usuarios en la web, porque algunos no estarían dispuestos a dejar de usar los bloqueadores.

Para poder aplicarla hay que detectar si está en uso alguna de estas herramientas y en dicho caso, hacer que salte una ventana que bloquee el uso de la web y que no pueda ser cerrada. Para detectar si está en uso ABP o alguna de estas herramientas se puede añadir un contenido a la web en el que aparezca la palabra anuncio y comprobar después si está activo dicho contenido. Por ejemplo:

Contenido añadido:

```
<div id="tester" style="display:none">an advertisement</div>
```

Detección de si está activo:

```
<script type="text/javascript">
    if (document.getElementById("tester") == undefined)
    {
        document.write('Adblock activated');
    }
</script>
```

Un ejemplo de aplicación la encontramos al intentar escuchar la radio de los 40 principales con ABP activo. Nos sale el siguiente mensaje:



Ilustración 36: Mensaje de aviso para desactivar ABP en la web yes.fm (radio de los 40 principales)

Para poder escuchar la radio, tendremos que desactivar Adblock Plus o añadir la web a la lista de excepciones y, además, actualizar la página.

Al igual que en la opción anterior, hay que conseguir mantener a los usuarios contentos y no avasallarles con publicidad incómoda una vez han decidido confiar.

5.3 Dejar de ofrecer contenido gratuito

Esta solución, si podemos llamarlo así, es la más radical por parte del dueño de la web. Al final, si el problema del bloqueo de anuncios continúa por este camino y se vuelve más grave, será la opción que tomen la mayoría de las webs. En conclusión, se acabará el contenido gratuito.

Apenas hay webs que hayan tomado esta elección por ahora, pero sí que podemos ver ya indicios en webs que piden donativos. Quizá el ejemplo más sonado es la Wikipedia y, si accedemos a ella, veremos lo siguiente:

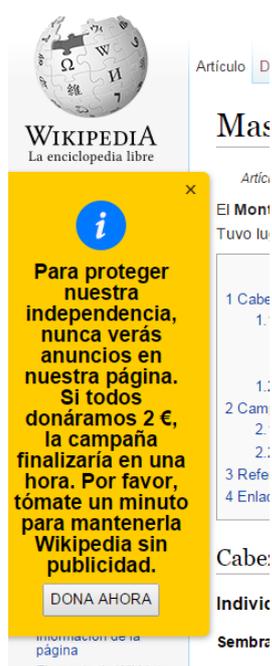


Ilustración 37: Imagen de la campaña de recaudación de Wikipedia

Ver algo así en una fuente de información tan importante hoy en día como es la Wikipedia da que pensar.

5.4 Modificar el tipo de publicidad

Ya se ha hablado de esto en algunas de las otras soluciones y es que, así como de las anteriores opciones se podía elegir una, esta es una opción de prácticamente obligado cumplimiento.

El motivo por el cual estamos hablando de que hay un problema es que la publicidad en la web hace mucho que se pasó de la raya y es algo que hay que solucionar urgentemente.

5.5 Modificar el funcionamiento de los bloqueadores de anuncios

Hasta ahora hemos visto las soluciones que pueden aplicar los propietarios y administradores de las webs, pero hay otros que también pueden ayudar a solucionar el problema. Por ejemplo, los desarrolladores de bloqueadores podrían adaptar los mismos para que no bloqueen nada más que lo necesario para mejorar la experiencia del usuario. También, podrían darle un mayor control al usuario, porque si bien la mayoría permiten modificar los filtros que se usan, esto no es una tarea trivial.

6 Implementación de una solución

Después de analizar las diferentes soluciones propuestas en el apartado anterior, me he decantado por la que consiste en modificar ABP para darle mayor control al usuario (apartado 5.5). Mediante el proceso de ingeniería inversa explicado anteriormente, he dotado a ABP de nuevas funcionalidades que harán más fácil al usuario controlar lo que quiere bloquear y que beneficiara a las tres partes (usuario, web y anunciante).

La motivación para hacer esto es que el usuario necesita tener un mayor control para poder adaptarlo a sus necesidades. Además, la modificación de ABP es muy compleja por lo que un usuario medio sería incapaz de realizarla por su cuenta.

6.1 La idea

Introducir en el menú de Adblock Plus un slider⁴ que permita regular el nivel de seguridad que se aplicará a la navegación. En un principio existirán 4 niveles:

- Nivel 1: Ninguna seguridad, no se bloquea nada.
- Nivel 2: Seguridad baja, se bloquean los pop-ups.
- Nivel 3: Seguridad intermedia, se bloquean todos los anuncios menos los que son imágenes.
- Nivel 4: Máxima seguridad, se bloquean todos los anuncios.

El objetivo aparte de dar control al usuario es incentivar el uso del nivel 3 de seguridad, de tal manera que se permitan anuncios en formato de imagen que no molesten.

Un primer boceto de la idea sería el siguiente:

⁴ Elemento de interfaz gráfica que permite seleccionar un valor moviendo un indicador.

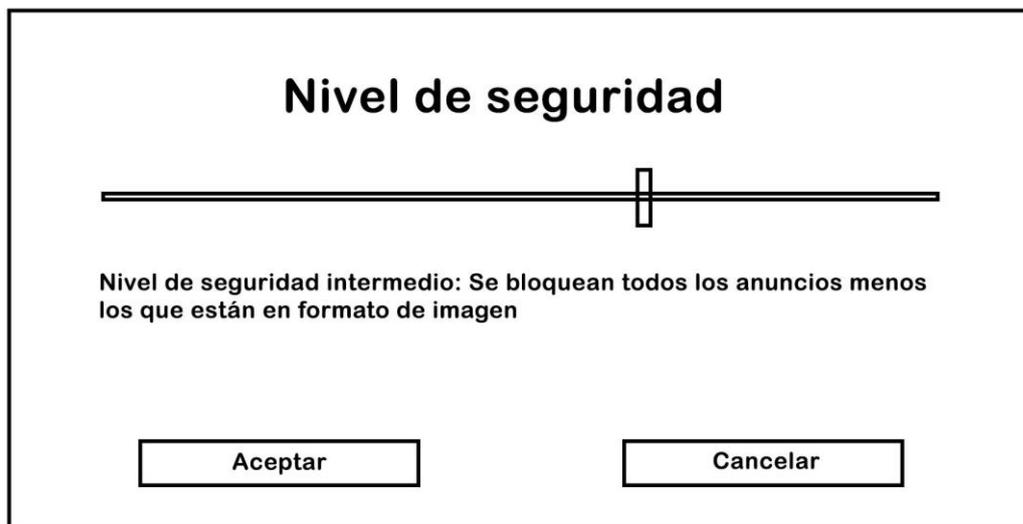


Ilustración 38: Primer boceto de la idea

6.2 El proceso de desarrollo

6.2.1 Obteniendo el código fuente de ABP

El primer paso para poder llevar a cabo esta idea es poder modificar AdBlock Plus. El formato de las extensiones de Firefox suele ser .xpi, que es algo similar al famoso .zip. Normalmente, cuando se va a instalar AdBlock Plus, se hace desde la página oficial o desde la página de descarga de extensiones de Firefox. Este proceso instala directamente el complemento sin que veamos el archivo de instalación. Para poder manejar dicho archivo, se optó por ir a la página oficial y en vez de hacer clic izquierdo en instalar, hacer clic derecho y seleccionar guardar enlace como.



Ilustración 39: Captura de pantalla mostrando cómo descargar el fichero de instalación de ABP

De esta manera se pudo obtener el fichero .xpi.

El siguiente paso es poder acceder al interior de este fichero. Para ello hay que cambiar el programa predeterminado que se ejecuta con la extensión .xpi por un compresor de archivos. Una vez hecho esto, permitirá abrir el archivo como si se tratase de un .zip y, por tanto, también descomprimirlo.

Para poder manejar los archivos dentro del .xpi, descomprimos el fichero. Y, con esto ya se está preparado para empezar a modificar el código.

6.2.2 Modificando el código

El primer paso a la hora de modificar el código fue introducir una nueva opción en el menú popup de AdBlock Plus que diese acceso a la ventana en la que se podría regular la seguridad. Para esto hay que ir al archivo en el que se especifica dicho menú. Se trata de “overlay.xul”. En este archivo se añade lo siguiente (seleccionado en la imagen):

```
<menuitem id="abp-status-whitelistsite" labeltempl="%whitelist.site.label;" type="checkbox" command="abp-command-toggle-site">
</menuitem>
<menuitem id="abp-status-whitelistpage" label="%whitelist.page.label;" type="checkbox" command="abp-command-toggle-page-white">
</menuitem>
<menuitem id="abp-status-disabled" label="%disable.label;" type="checkbox" key="abp-key-enable" command="abp-command-enable">
</menuitem>
<menuseparator/>
<menuitem id="abp-status-mod" label="%mod.label;" command="abp-command-mod"/>
</menuitem>
<menuseparator/>
<menu id="abp-status-options" label="%options.label;">
<menupopup id="abp-status-options-popup">
<menuitem id="abp-status-frameobjects" label="%objecttabs.label;" type="checkbox" command="abp-command-toggle-objecttabs"/>
<menuitem id="abp-status-savestats" label="%count-hits.label;" type="checkbox" command="abp-command-toggle-savestats"/>
<menuitem id="abp-status-sync" label="%sync.label;" type="checkbox" command="abp-command-toggle-sync"/>
</menupopup>
</menu>
```

Ilustración 40: Código añadido a overlay.xul (1ª vez)

Con esto ya tenemos un nuevo elemento en el menú y añadimos un separador para que quede bien diferenciado de los originales. Ahora lo que hay que hacer es asignar el texto de esta opción. Dado que en ABP está todo organizado para adaptarse a diferentes idiomas, habrá que dar este valor en los archivos correspondientes a cada idioma. Por ejemplo para dar el valor en castellano, habrá que ir a la carpeta “locale/es-ES”. Dentro de esta carpeta hay que abrir el archivo “overlay.dtd” y añadir lo siguiente (seleccionado en la imagen):

```
<!ENTITY whitelist.site.label "Desactivar en ?1?">
<!ENTITY filters.label "%&#x26;Preferencias de filtros">
<!ENTITY disable.label "Desactivar globalmente">
<!ENTITY mod.label "Modificar nivel de seguridad">
<!ENTITY objecttab.title "Bloquear">
<!ENTITY objecttab.tooltip "Pulse aquí para bloquear este objeto con">
<!ENTITY menuitem.label "%&#x26;AdBlock Plus">
<!ENTITY objecttabs.label "%&#x26;Mostrar pestañas en Flash y Java">
```

Ilustración 41: Código añadido a overlay.dtd

Como podemos observar, el nombre de la entidad que se ha añadido en el archivo .dtd coincide con el valor del atributo label del elemento del menú que se ha añadido en el archivo .xul.

El último paso de la introducción del elemento en el menú es darle funcionalidad, es decir, que abra la ventana que queremos. Para esto hay que modificar dos archivos. El primero es, de nuevo, el fichero “overlay.xul”. El segundo es “ui.js”, que se encuentra en la carpeta “lib”. Lo que hay que introducir es lo siguiente (seleccionado en las imágenes):

```
<command id="abp-command-settings" oncommand="//"/>
<command id="abp-command-sidebar" oncommand="//"/>
<command id="abp-command-mod" oncommand="//"/>
<command id="abp-command-togglestitewhitelist"/>
<command id="abp-command-togglepagewhitelist"/>
<command id="abp-command-toggleabitabs"/>
```

Ilustración 42: Código añadido a overlay.xul (2ª vez)

```
["abp-command-filters", "command", UI.openFiltersDialog.bind(UI)],
["abp-command-sidebar", "command", UI.toggleBottombar.bind(UI)],
["abp-command-mod", "command", UI.openModDialog.bind(UI)],
["abp-command-togglestitewhitelist", "command", function() { UI.toggleFilter
["abp-command-togglepagewhitelist", "command", function() { UI.toggleFilter
```

Ilustración 43: Código añadido a ui.js

Con estas líneas se indica que al seleccionar el elemento del menú se debe ejecutar la función openModDialog. Por supuesto, hay que crear esta función. Y se hace en el mismo “ui.js”.

```
/**
 * Opens security level dialog.
 * @param {Filter} [filter] filter to be selected
 */
openModDialog: function(filter)
{
  Services.wv.openWindow(null, "chrome://adblockplus/content/ui/mod.xul", "_blank", "chrome,centerscreen,resizeable,dialog=no", {wrappedJSObject: filter});
},
```

Ilustración 44: Función openModDialog en ui.js

Una vez hecho esto ya está listo el nuevo elemento del menú. Solo falta definir la nueva ventana y su funcionalidad.

Para crear la nueva ventana, hay que crear varios archivos nuevos. Uno para la representación de la ventana, otro para sus funciones y varios para el texto que aparece en la ventana (uno por idioma). Por otro lado, habrá que hacer alguna modificación en los ficheros originales de ABP para poder almacenar el nivel de seguridad, que persista tras cerrar la sesión y para que tenga efecto en la navegación el nivel de seguridad.

Vamos a ver primero cómo se definiría la ventana nueva. El archivo será escrito en XUL, por lo que su extensión será .xul. En este caso, se ha definido un diálogo que se abrirá al clicar en la opción del menú anteriormente introducida. Lo que se podrá ver en esta ventana es un título, un slider para seleccionar el nivel de seguridad, una etiqueta que dará una descripción del nivel y dos botones (aceptar y cancelar). Para conseguir todo esto, el código introducido en el archivo “mod.xul” es el siguiente:

```

<>xml version="1.0" >>
<?xml-stylesheet href="chrome://global/skin/global.css" type="text/css" ?>

<!DOCTYPE dialog SYSTEM "chrome://adblockplus/locale/mod.dtd">

<dialog id="modDiag" title="{diag.title}"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
  width="550"
  height="200"
  buttons="accept, cancel"
  buttonlabelcancel="{close.label}"
  buttonlabelaccept="{accept.label}"
  onload="onLoad();"
  ondialogaccept="return doOK();"
  ondialogcancel="return doCancel();">
  <script type="application/x-javascript;version=1.7" src="mod.js"/>
  <script type="application/x-javascript;version=1.7" src="utils.js"/>
  <dialogheader title="{seg.title}" description="{seg.desc}"/>
  <groupbox>
    <scale min="1" max="4" id="slider" pageincrement="1"/>
    <label id="slidertextbox"/>
  </groupbox>
  <script>
  var slider = document.getElementById("slider");
  slider.addEventListener('change', valueChanged, true);
  </script>
</dialog>

```

Ilustración 45: Contenido del archivo mod.xul (definición de la nueva ventana)

Ahora vamos a ver cómo sería el archivo que contiene las funciones de la ventana. Estas funciones serán básicamente 4: actualizar el valor de la etiqueta en función del valor del slider, guardar el resultado al clicar en aceptar, descartar los cambios al clicar en cancelar y cargar el valor del slider y de la etiqueta cuando se habrá la ventana. El archivo será un fichero javascript con el mismo nombre que el archivo .xul.

```

function valueChanged(e){
  var val = e.target.getAttribute('value');
  var text = E("slidertextbox");
  switch(val) {
    case "1":
      text.setAttribute('value', "Ninguna seguridad: No bloquea ningún elemento.");
      break;
    case "2":
      text.setAttribute('value', "Nivel bajo de seguridad: Solo bloquea pop-ups.");
      break;
    case "3":
      text.setAttribute('value', "Nivel intermedio de seguridad: Bloquea todo menos anuncios en formato de imagen.");
      break;
    case "4":
      text.setAttribute('value', "Nivel máxima: Bloquea todo tipo de publicidad.");
      break;
    default:
      break;
  }
}

function doOK(){
  var val = E("slider").getAttribute('value');
  Policy.changeLevel(parseInt(val));
  return true;
}

function doCancel(){
  return true;
}

function onLoad(){
  E("slider").setAttribute('value', Policy.secLevel.toString());
  var val = E("slider").getAttribute('value');
  var text = E("slidertextbox");
  switch(val) {
    case "1":
      text.setAttribute('value', "Ninguna seguridad: No bloquea ningún elemento.");
      break;
    case "2":
      text.setAttribute('value', "Nivel bajo de seguridad: Solo bloquea pop-ups y scripts.");
      break;
    case "3":
      text.setAttribute('value', "Nivel intermedio de seguridad: Bloquea todo menos anuncios en formato de imagen.");
      break;
    case "4":
      text.setAttribute('value', "Nivel máxima: Bloquea todo tipo de publicidad.");
      break;
    default:
      break;
  }
}

```

Ilustración 46: Contenido del archivo mod.js (funciones de la nueva ventana)

En el archivo, se pueden ver una especie de atajos, como “E”. Para poder usarlos, hay que importar la librería “utils.js”. Este paso se puede observar en la imagen del archivo .xul junto a la línea en la que se importa el javascript que contiene las funciones de la ventana.

El siguiente paso fue crear un archivo .dtd en cada carpeta de idioma (dichas carpetas están en el directorio “locale”). En el caso del castellano, el archivo quedaría de la siguiente manera:

```

<!ENTITY diag.title "Modificar nivel de seguridad">
<!ENTITY accep.label "Aceptar">
<!ENTITY close.label "Cancelar">
<!ENTITY seg.title "Seguridad:">
<!ENTITY seg.desc "Nivel de seguridad">

```

Ilustración 47: Contenido del archivo mod.dtd de la carpeta correspondiente al idioma castellano

Lo que falta para terminar con el proceso es modificar algunos archivos originales de Adblock Plus para incluir la nueva funcionalidad que aporta la ventana. Dichos archivos son “contentPolicy.js” (carpeta “lib”) y “prefs.js” (carpeta “defaults”). El objetivo de modificar estos archivos es conseguir que el nivel de seguridad persista y que tenga efecto.

Vamos a ver, primeramente, cómo añadir una nueva preferencia que va a ser la que represente el nivel de seguridad. De esta manera se conservará el nivel de seguridad de ABP junto con el resto de sus preferencias aunque la sesión se termine. Las preferencias se encuentran en el archivo “pref.js” y, por tanto, la nueva preferencia habrá que añadirla al mismo. Por defecto, pondremos el nivel de seguridad en 3. La modificación en el archivo es la siguiente (seleccionado en la imagen):

```

pref("extensions.adblockplus.savestats", true);
pref("extensions.adblockplus.composer_default", 2);
pref("extensions.adblockplus.security_level", 3);
pref("extensions.adblockplus.clearStatsOnHistoryPurge", true);
pref("extensions.adblockplus.report_submiturl", "https://reports.adblockplus.org/submi

```

Ilustración 48: Modificación en el archivo prefs.js

Ahora hay que crear una variable en la clase Policy (en la que parece que tiene más sentido), que represente el nivel de seguridad. Esta variable se inicializará cuando se inicialice la clase, es decir, en la función “init”. Además, hay que crear una función para guardar el valor del slider cuando se clique en aceptar. Las modificaciones pertinentes serían las siguientes (seleccionadas en la imagen):

```

* @type Object
*/
nonVisual: {},

/**
 * Contains an int that represents the security level.
 * @type Int
 */
secLevel: 3,

/**
 * Map containing all schemes that should be ignored by content policy.
 * @type Object
 */
whitelistSchemes: {},

/**
 * Called on module startup, initializes various exported properties.
 */
init: function()
{
  // type constant by type description and type description by type constant
  let iface = Ci.nsIContentPolicy;
  for (let typeName of contentTypes)
  {
    if ("TYPE_" + typeName in iface)
    {
      let id = iface["TYPE_" + typeName];
      this.type[typeName] = id;
      this.typeDescr[id] = typeName;
      this.localizedDescr[id] = Utils.getString("type_label_" + typeName.toLowerCase());
    }
  }

  this.secLevel = Prefs.security_level;
  this.type.ELEMHIDE = 0xFFFFD;
  this.typeDescr[0xFFFFD] = "ELEMHIDE";
  this.localizedDescr[0xFFFFD] = Utils.getString("type_label_elemhide");

  this.type.POPUP = 0xFFFE;
  this.typeDescr[0xFFFE] = "POPUP";
  this.localizedDescr[0xFFFE] = Utils.getString("type_label_popup");

  for (let type of nonVisualTypes)
    this.nonVisual[this.type[type]] = true;

  // whitelisted URL schemes
  for (let scheme of Prefs.whitelistSchemes.toLowerCase().split(" "))
    this.whitelistSchemes[scheme] = true;

  // Generate class identifier used to collapse node and register corresponding
  // stylesheet.
  let offset = "a".charCodeAt(0);
  for (let i = 0; i < 20; i++)
    collapsedClass += String.fromCharCode(offset + Math.random() * 26);

  let collapseStyle = Services.io.newURI("data:text/css," +
    encodeURIComponent("." + collapsedClass +
      "{-moz-binding: url(chrome://global/content/bindings/general.xml#fooobarbazdummy) !import");
  Utils.styleService.loadAndRegisterSheet(collapseStyle, Ci.nsIStyleSheetService.USER_SHEET);
  onShutdown.add(function()
  {
    Utils.styleService.unregisterSheet(collapseStyle, Ci.nsIStyleSheetService.USER_SHEET);
  });
},

/**
 * Changes security level
 * @param lvl {Int}
 */
changeLevel: function(lvl)
{
  this.secLevel = lvl;
  Prefs.security_level = lvl;
},

```

Ilustración 49: Modificaciones en contentPolicy.js (1ª parte)

El último paso para conseguir el objetivo es hacer que al procesar las peticiones se tenga en cuenta el nivel de seguridad. Como se pudo comprobar en el apartado en el que se desmenuza el proceso por el que se analizan las peticiones, las funciones claves son “processNode” de la clase Policy y “shouldLoad” de la clase PolicyImplementation. Por esto, hay que introducir una serie de condiciones en estas funciones. Una condición por nivel de seguridad.

Las condiciones correspondientes al nivel 1 y 2 de seguridad se introducirán en la función “processNode”. Las condiciones introducidas son las siguientes (seleccionadas en la imagen):

```

* @param location {nsIURI}
* @param collapse {Boolean} true to force hiding of the node
* @return {Boolean} false if the node should be blocked
*/
processNode: function(wnd, node, contentType, location, collapse)
{
  // Ignore all if security level is 1
  if (this.secLevel == 1)
    return true;

  //Blocks POPUPs if security level is higher than 1
  if (!(contentType == Policy.type.POPUP || contentType == Policy.type.SCRIPT) && this.secLevel == 2)
    return true;

  let topWnd = wnd.top;
  if (!topWnd || !topWnd.location || !topWnd.location.href)
    return true;

```

Ilustración 50: Condiciones introducidas en la función processNode

La condición correspondiente al nivel 3 de seguridad se añadirá en la función shouldLoad.

```

shouldLoad: function(contentType, contentLocation, requestOrigin, node, mimeTypeGuess,
{
  if (Policy.secLevel <= 3)
    return Ci.nsIContentPolicy.ACCEPT;

  // Ignore requests without context and top-level documents
  if (!node || contentType == Policy.type.DOCUMENT)
    return Ci.nsIContentPolicy.ACCEPT;

```

Ilustración 51: Condición introducida en la función shouldLoad

6.2.3 Instalando la extensión modificada

Cuando ya se han hecho las modificaciones en el código, basta con comprimir los archivos de nuevo en un .zip y cambiar manualmente la extensión a .xpi. Sin embargo, da problemas a la hora de instalarla porque sale un error.

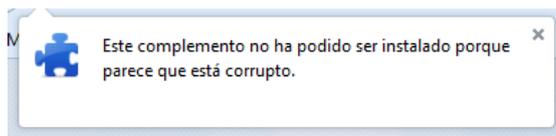


Ilustración 52: Mensaje de error a la hora de instalar la extensión modificada.

Para evitar este problema, hay que ir a la carpeta donde se encuentran instaladas las extensiones de Firefox e identificar cuál de estas es ABP. El nombre del .xpi que contiene AdBlock Plus es una combinación de letras y números entre corchetes. Suele ser: {d10d0bf8-f5b5-c8b4-a8b2-2b9879e08c5d}.

Una vez localizado ABP, se copia el nombre y se borra. Luego, se cambia el nombre del archivo .xpi de la extensión modificada por el que hemos copiado y se mueve la extensión modificada a la ubicación en la que se encontraba el archivo original de AdBlock Plus.

Al terminar este proceso la extensión estará instalada sin problemas en Firefox.

6.3 El resultado

Tras completar el proceso, la extensión ya ha sido modificada y está lista para funcionar. A continuación vamos a ver su efecto en una web que tiene diversos tipos de anuncios y que, por tanto, permite ver los diferentes grados de seguridad. La página elegida es “animeflv.net”. Al margen del contenido que ofrece, los diferentes tipos de anuncios que contiene son muy interesantes. Se puede encontrar publicidad por popup, frames, imágenes fijas, e incluso imágenes dinámicas. Dicho esto vamos a ver el nuevo comportamiento de la extensión.

Lo primero es abrir Firefox y cargar una página. En este caso abriremos la página que se ha mencionado antes.

Ahora vamos a ver cómo ha quedado el menú de Firefox tras la modificación.

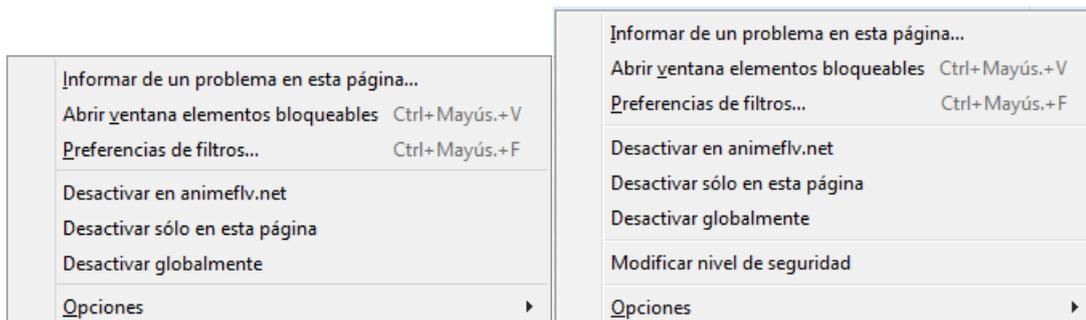


Ilustración 53: Menú antes (izquierda) y después (derecha) de la modificación

Una vez visto que, efectivamente, se ha añadido una nueva opción al menú, vamos a observar que efecto produce seleccionarla. Al clicar encima de la opción, se abre un diálogo que es lo más semejante posible a la idea original.

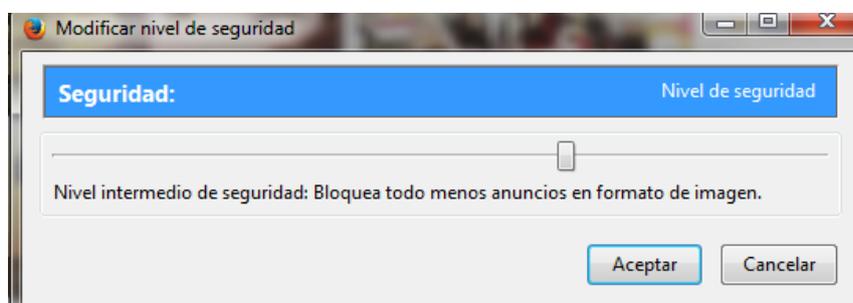


Ilustración 54: Diálogo de modificación del nivel de seguridad (Tercer nivel activo)

Si es la primera vez que se abre esta ventana, se mostrará lo mismo que en esta ilustración. Sin embargo, si se ha abierto anteriormente y se ha modificado el nivel, se mantendrá guardado y los valores corresponderán a los de la última actualización. Las siguientes tres imágenes corresponden a los otros niveles de seguridad:

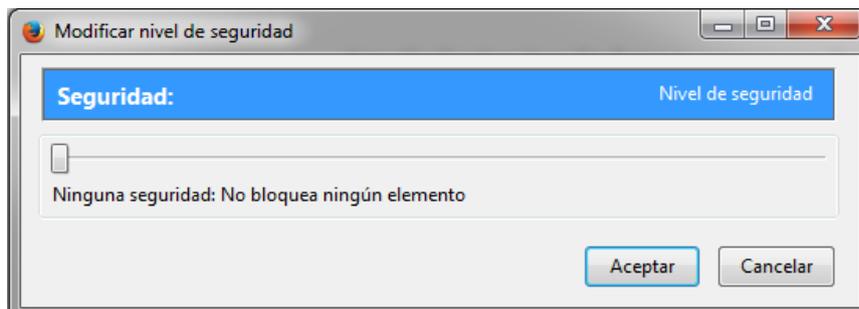


Ilustración 55: Diálogo de modificación del nivel de seguridad (Primer nivel activo)

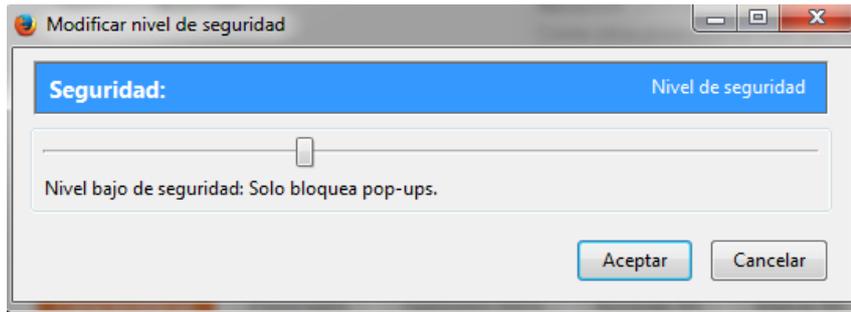


Ilustración 56: Diálogo de modificación del nivel de seguridad (Segundo nivel activo)

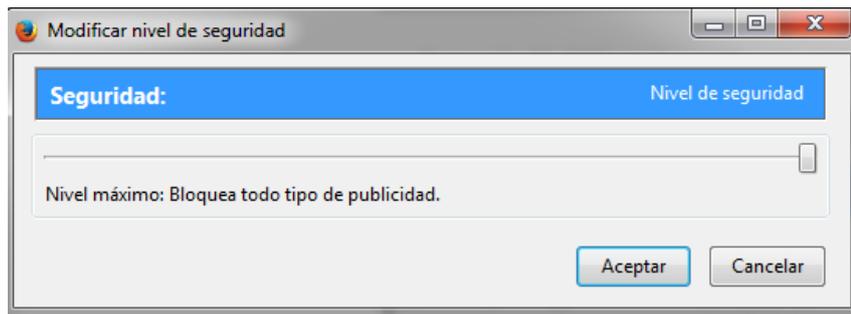


Ilustración 57: Diálogo de modificación del nivel de seguridad (Cuarto nivel activo)

Lo único que queda por ver es el efecto en la web de los distintos niveles de seguridad. Empezaremos viendo cómo se comporta la web con ABP desactivado. En las imágenes se ha desenfocado lo que no hay que tener en cuenta, viéndose mejor la publicidad en la que hay que mantener la atención.

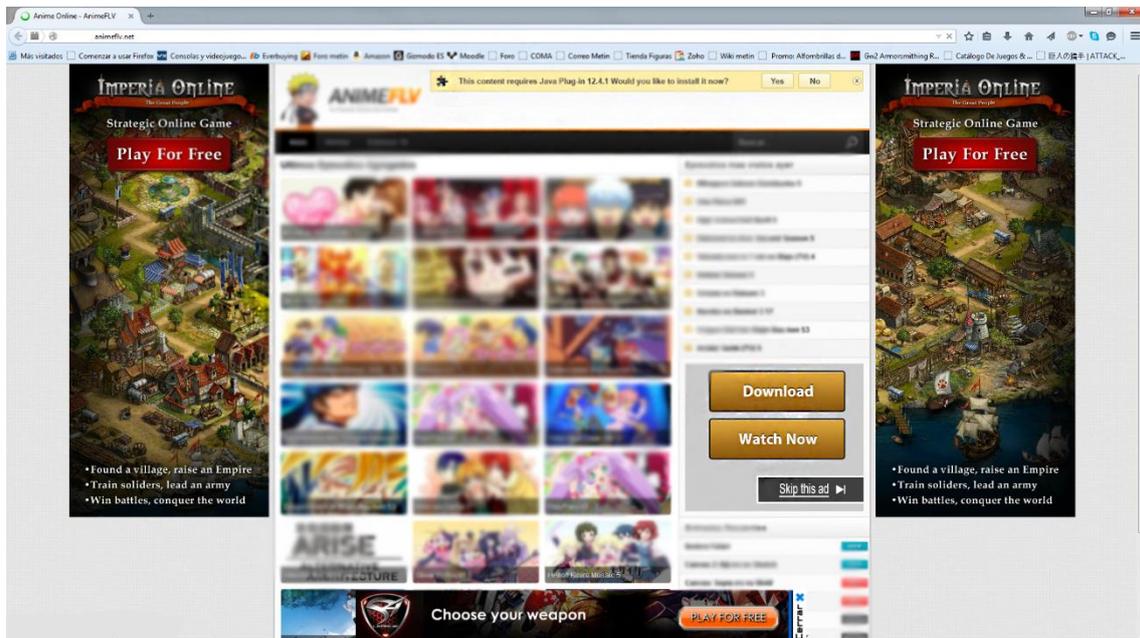


Ilustración 58: Efecto de tener ABP desactivado en animev.net

Además de esto, hay que tener en cuenta que clicando en prácticamente cualquier sitio, se abren popups con publicidad que no tiene nada que ver con la página.



Ilustración 59: Popups (a partir del segundo empezando por la izquierda) que se abren con ABP desactivado en animeflv.net

Con el primer nivel de seguridad el efecto debe ser el mismo a tener Adblock Plus desactivado. Se podrá observar en las imágenes que la publicidad no es la misma que la que se ve en las imágenes del funcionamiento con ABP desactivado. Esto se debe a que en cada visita a la web se ve una publicidad diferente. Lo que hay que hacer es no fijarse en el contenido de los cuadros de publicidad sino en su situación en la web, es decir, cómo afectan a la navegación.

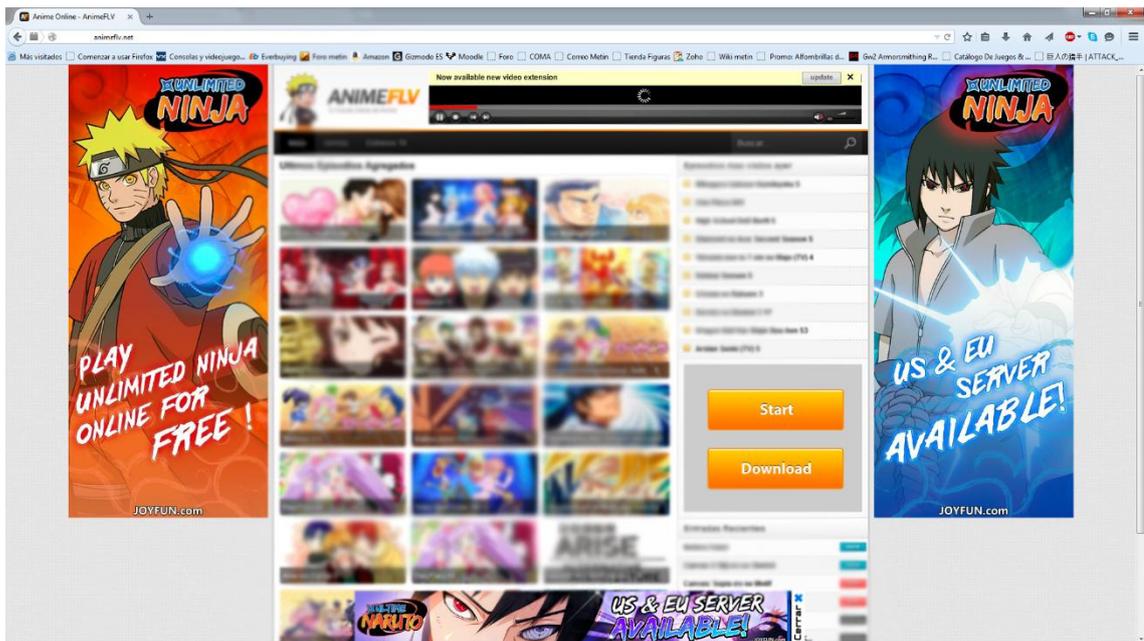


Ilustración 60: Efecto del nivel 1 de seguridad en animeflv.net

Asimismo siguen abriéndose popups. Aunque también hay que advertir que el número de popups que se abren y su contenido varían de una visita a otra. Lo importante es comprobar que siguen abriéndose.



Ilustración 61: Popups abiertos (a partir del segundo empezando por la izquierda) cuando se aplica el nivel 1 de seguridad en animeflv.net

El siguiente paso es ver cómo se comporta la web con el nivel 2 de seguridad. El efecto debería ser el mismo que con el 1, salvo que no se abrirán popups.

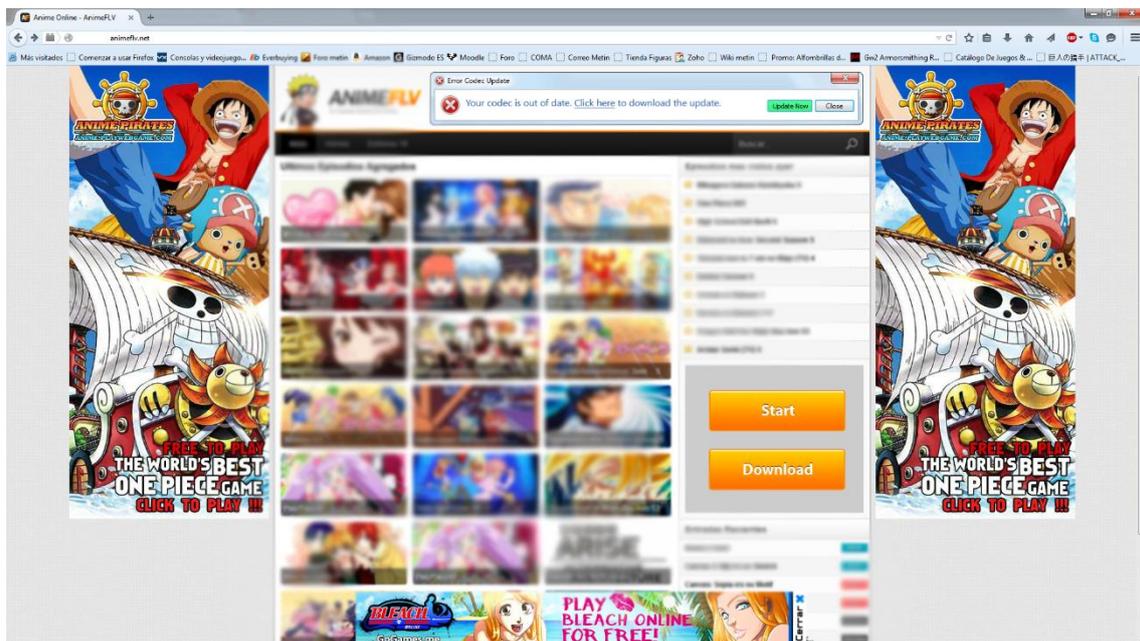


Ilustración 62: Efecto del nivel 2 de seguridad en animeflv.net

Aunque parezca que todo está igual, si clicamos en cualquier lugar de la web, los popup intentarán abrirse pero ABP los cerrará, por lo que únicamente aparecerá en nuestra barra de inicio la ventana en la que estamos navegando.



Ilustración 63: Efecto sobre los popups del nivel 2

En el siguiente nivel de seguridad (nivel 3) se ve un paso más grande en cuando a lo que se bloquea. Ahora deberán desaparecer todos aquellos anuncios que ocupen frames que

molestan. De esta manera, deberán quedar solo los anuncios en formato de imagen que no molestan en el contenido.

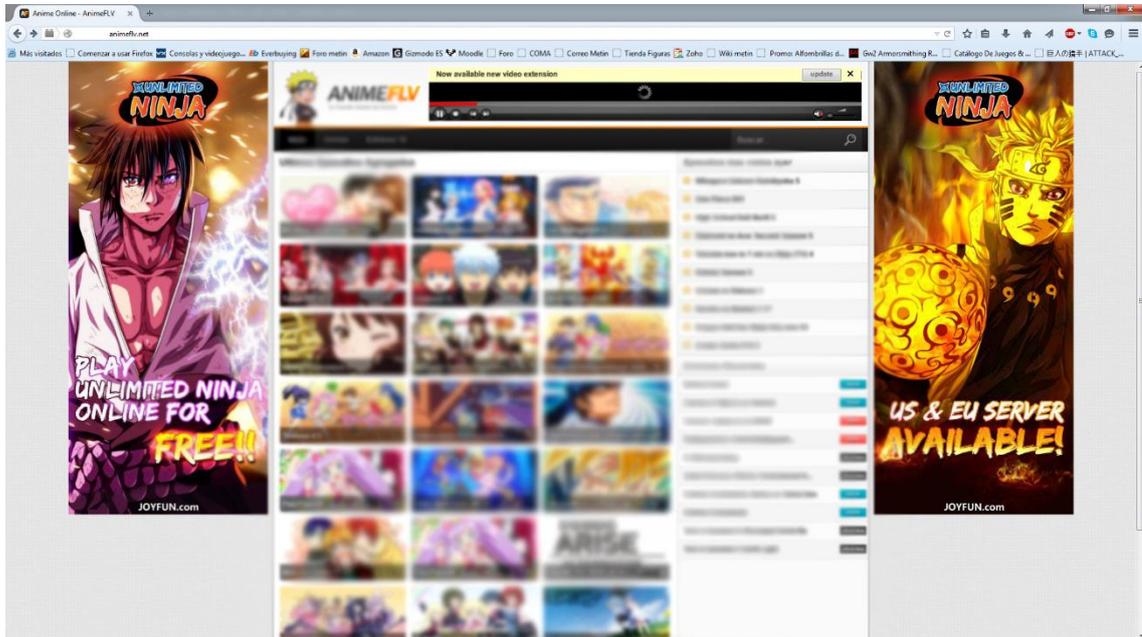


Ilustración 64: Efecto del nivel 3 de seguridad en animeflv.net

En este nivel tampoco se permite que se abran popups, es decir, en ese sentido, el resultado sería el mismo que en el nivel 2.

Por último, el nivel 4 ofrece el mismo nivel de protección que tener el ABP original activado.

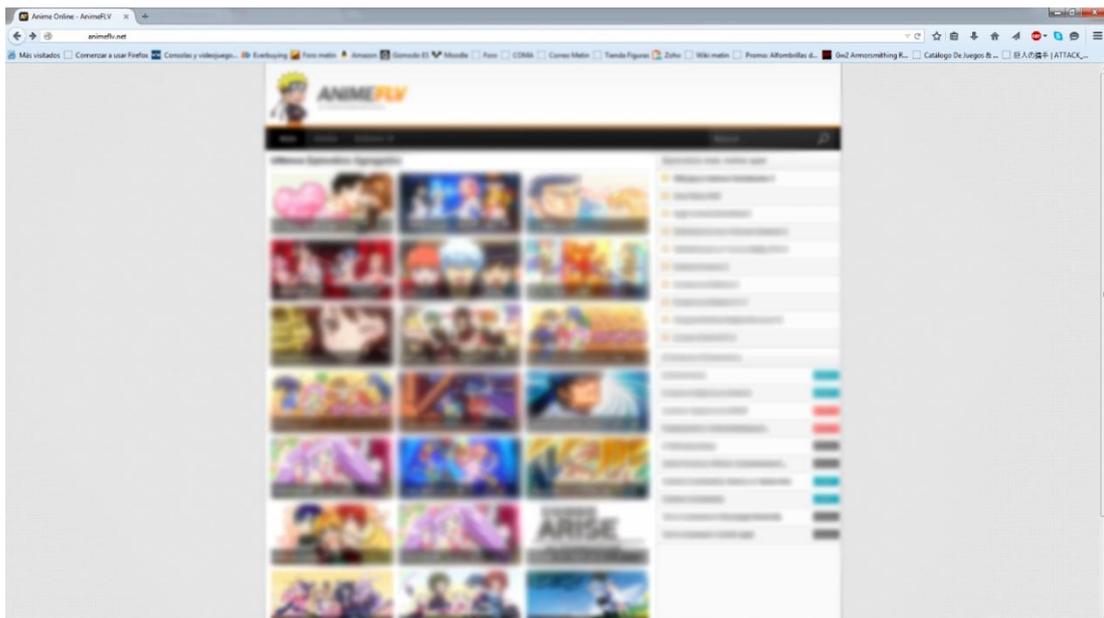


Ilustración 65: Efecto del nivel 4 de seguridad en animeflv.net

7 Gestión del proyecto

En el primer apartado de este capítulo se mostrará la planificación concebida para este proyecto. Dicha planificación incluirá la identificación de las tareas, un diagrama de Gantt, el EDT y unos subapartados dedicados a la gestión de riesgos y comunicaciones.

En el segundo apartado se recogerán aquellas desviaciones en el proyecto con respecto a la planificación inicial y se especificará el motivo.

7.1 Planificación

En este apartado se especificará la planificación pensada para realizar el proyecto.

7.1.1 Identificación de tareas

Las tareas a realizar a lo largo del proyecto serán las siguientes:

Nombre de la actividad:	A05: Tutorías
Fecha de inicio:	1 de Septiembre de 2014
Fecha de finalización	7 de Julio de 2015
Descripción:	Asistencia a tutorías para explicar mis dudas y problemas y recibir orientación para avanzar en el trabajo.
Dedicación:	1 hora por tutoría

Nombre de la actividad:	A10: Planificar el proyecto
Fecha de inicio:	1 de Septiembre de 2014
Fecha de finalización	10 de Septiembre de 2014
Descripción:	Se realizará una planificación inicial del proyecto. En esta planificación se identificarán las tareas a realizar y que dedicar a cada tarea.
Dedicación:	10 horas

Nombre de la actividad:	A20: Analizar la importancia del problema
Fecha de inicio:	10 de Septiembre de 2014
Fecha de finalización	31 de Diciembre de 2014
Descripción:	Esta tarea incluye hacer un análisis de la situación actual de la publicidad y del bloqueo de publicidad. No basta solo con analizar el bloqueo de publicidad porque hay que saber el porqué de su origen. Incluirá la búsqueda en Internet de artículos, lectura, resumen y transcripción a la memoria.
Dedicación:	80 horas

Nombre de la actividad:	A30: Obtener una idea general de qué es Adblock Plus
Fecha de inicio:	31 de Diciembre de 2014
Fecha de finalización	31 de Enero de 2015
Descripción:	Comenzar a entender de manera superficial qué es Adblock Plus y cómo funciona. Buscar información sobre la extensión.
Dedicación:	50 horas

Nombre de la actividad:	A32: Descubrir cómo está construido Adblock Plus
Fecha de inicio:	1 de Febrero de 2015
Fecha de finalización	10 de Febrero de 2015
Descripción:	Se analizarán los módulos que ABP utiliza, en que lenguajes está implementado y qué tipo de aplicación es.
Dedicación:	30 horas

Nombre de la actividad:	A34: Extraer las clases que implementa Adblock Plus
Fecha de inicio:	10 de Febrero de 2015
Fecha de finalización	18 de Febrero de 2015
Descripción:	Se extraerán las diferentes clases implementadas en la extensión ABP y se explicará cuál es su utilidad de manera muy breve.
Dedicación:	20 horas

Nombre de la actividad:	A36: Descubrir cómo funciona AdBlock Plus
Fecha de inicio:	18 de Febrero de 2015
Fecha de finalización	15 de Marzo de 2015
Descripción:	Se intentará comprender cómo realiza AdBlock Plus sus funciones principales, es decir, cómo bloquea los elementos de las webs.
Dedicación:	50 horas

Nombre de la actividad:	A40: Buscar soluciones al problema del bloqueo de anuncios
Fecha de inicio:	1 de Febrero de 2015
Fecha de finalización	15 de Marzo de 2015
Descripción:	A partir de la información obtenida en la primera actividad se propondrán una serie de medidas para llegar a un win-win para usuarios y creadores de contenido.
Dedicación:	15 horas

Nombre de la actividad:	A50: Implementar una solución
Fecha de inicio:	16 de Marzo de 2015
Fecha de finalización	30 de Abril de 2015
Descripción:	Se intentará implementar una solución de las propuestas anteriormente valiéndose de los conocimientos adquiridos al realizar ingeniería inversa sobre AdBlock Plus.
Dedicación:	80 horas

Nombre de la actividad:	A60: Preparar la defensa del proyecto
Fecha de inicio:	1 de Septiembre de 2014
Fecha de finalización	31 de Mayo de 2015
Descripción:	Preparación de la presentación pública del proyecto. Se realizarán una serie de transparencias que sirvan de apoyo para la presentación y se practicará la misma. Durante el proceso de realización del proyecto irán surgiendo ideas que añadir a la presentación. Por eso, esta actividad se sitúa durante todo el periodo que ocupa el proyecto.
Dedicación:	35 horas

Nombre de la actividad:	A70: Seguimiento y control
Fecha de inicio:	1 de Septiembre de 2014
Fecha de finalización	31 de Mayo de 2015
Descripción:	Actividad que incluye todas las subactividades que comprenden el seguimiento y control del proyecto. También se incluye la dedicación invertida en redactar el informe de seguimiento.
Dedicación:	20 horas

Sumando la dedicación de cada una de las actividades la cifra de horas invertidas alcanza las 390 horas. Hay que destacar también que dentro de la dedicación de cada actividad también se incluye la redacción de esa parte en la memoria.

7.1.2 Diagrama de Gantt

En el siguiente diagrama se plasmará de manera gráfica el periodo de tiempo pensado para cada actividad así como las dependencias entre las distintas actividades. Las dependencias entre actividades se indicarán con flechas.

Cada mes se dividirá en tres periodos de tal manera que las actividades que solo ocupan una parte del mes no reflejen una ocupación completa del mes. Si alguna actividad ocupa parte de uno de estos periodos se reflejará como que lo ocupa al completo, ya he hacer más divisiones puede hacer que se pierda comprensión en el diagrama.

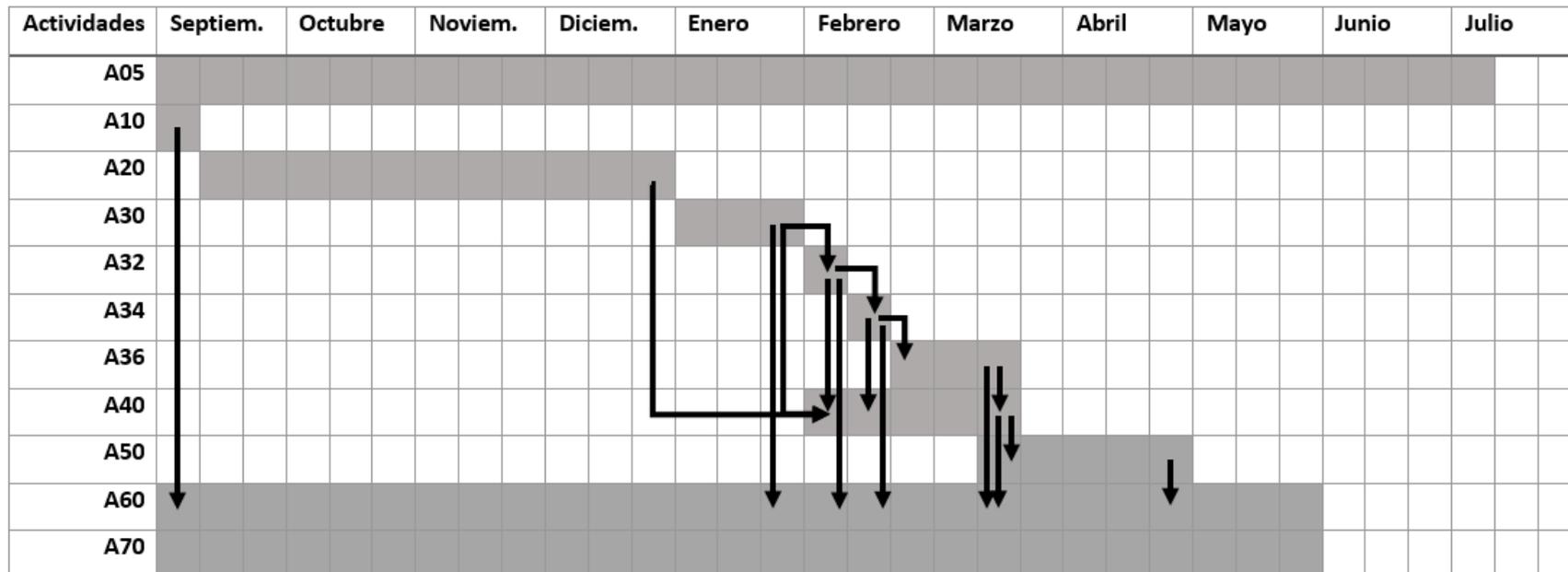


Ilustración 66: Diagrama de Gantt

7.1.3 EDT



Ilustración 67: EDT

7.1.4 Gestión de Riesgos

En este apartado se analizarán los diferentes riesgos que hay en la realización del proyecto y se especificará una manera para solventarlos.

1. **Enfermedad:** En caso de que se contraiga una enfermedad que incapacite proseguir con el trabajo, la dedicación prevista se aplazará hasta que la enfermedad desaparezca o se pueda continuar.
2. **Perdida de datos:** Para evitar este riesgo se crearán a menudo diferentes copias de la memoria y del código de la solución a implementar. Estas copias se guardarán en el propio PC y en Dropbox, para que en caso de que falle el PC siempre quede la versión de la nube.
3. **Perder la plataforma de trabajo:** En caso de que el ordenador desde el que se está realizando el proyecto sufra una avería, se intentará buscar un sustituto entre las personas cercanas. Por suerte, al ser un proyecto que, en principio, no requiere instalar entornos de trabajo (tan solo el programa NotePad++), será muy sencillo cambiar de máquina.
4. **No cumplirse los plazos:** Si a medida que avanza el proceso de realización del proyecto no se van cumpliendo los plazos en los que se deberían haber terminado ciertas tareas, se reajustarán los plazos para poder seguir cumpliendo el objetivo de completar satisfactoriamente el proyecto.

7.1.5 Gestión de comunicaciones

Para mantener la comunicación con Oscar (tutor del proyecto), se utilizarán los correos electrónicos y, se concertarán algunas reuniones para tratar temas que sería más complicado tratar a través de correo.

7.1.6 Análisis de viabilidad

Tras identificar las diferentes tareas, estimar la dedicación que será necesaria para cada una de ellas, identificar los riesgos y establecer un método de comunicación, conociendo la disponibilidad para el periodo de tiempo del proyecto, se puede afirmar que lograr los objetivos del proyecto es viable.

7.2 Seguimiento y control

A lo largo de la realización del proyecto se han realizado una serie de ajustes sobre la planificación original para poder alcanzar el objetivo de completar el proyecto. Dichos ajustes hacen referencia a la dedicación y espacio en el tiempo que ocupan las actividades. Vamos a ver la comparación entre lo previsto y lo realmente sucedido. El motivo de las desviaciones significativas (10 horas o más) se especificará en cada caso.

Primero vamos a ver la variación en la dedicación.

Actividad	Estimado	Realidad	Desviación	Motivo
A05	---	3h	3h	No se sabía de antemano el número de tutorías a las que se iba a acudir.
A10	10h	10h	---	
A20	80h	90h	+10h	Había numerosos artículos e información a analizar. Además había varias fuentes en inglés, por lo que eso ralentizó el progreso.
A30	50h	60h	+10h	Costó mucho comenzar a comprender cómo funcionaba ABP.
A32	30h	20h	-10h	Fue más sencillo de lo previsto descubrir que módulos usaba y en que lenguajes estaba desarrollado ABP.
A34	20h	20h	---	
A36	50h	65h	+15h	Comprender a fondo el funcionamiento de la extensión fue bastante complicado. También se invirtió más tiempo del esperado al describir el proceso en la memoria. Sobre

				todo en realizar los diagramas de secuencia.
A40	15h	15h	---	
A50	80h	70h	-10h	Una vez analizada la extensión fue más sencillo de lo esperado elaborar la mejora sobre el código original.
A60	35h	40h	+5h	
A70	20h	20h	---	
Total	390h	413h	+23h	

Por último vamos a ver cómo se han desplazado en el tiempo las actividades. Para ello se va a poner el mismo diagrama de Gantt de la planificación, pero sin indicar las dependencias. Las desviaciones se indicarán en rojo. En las actividades que solo se haya extendido la fecha de finalización, se prolongará la representación en el diagrama en rojo. Sin embargo, si también se ha afectado su fecha de origen, se indicará la actividad completa en rojo.

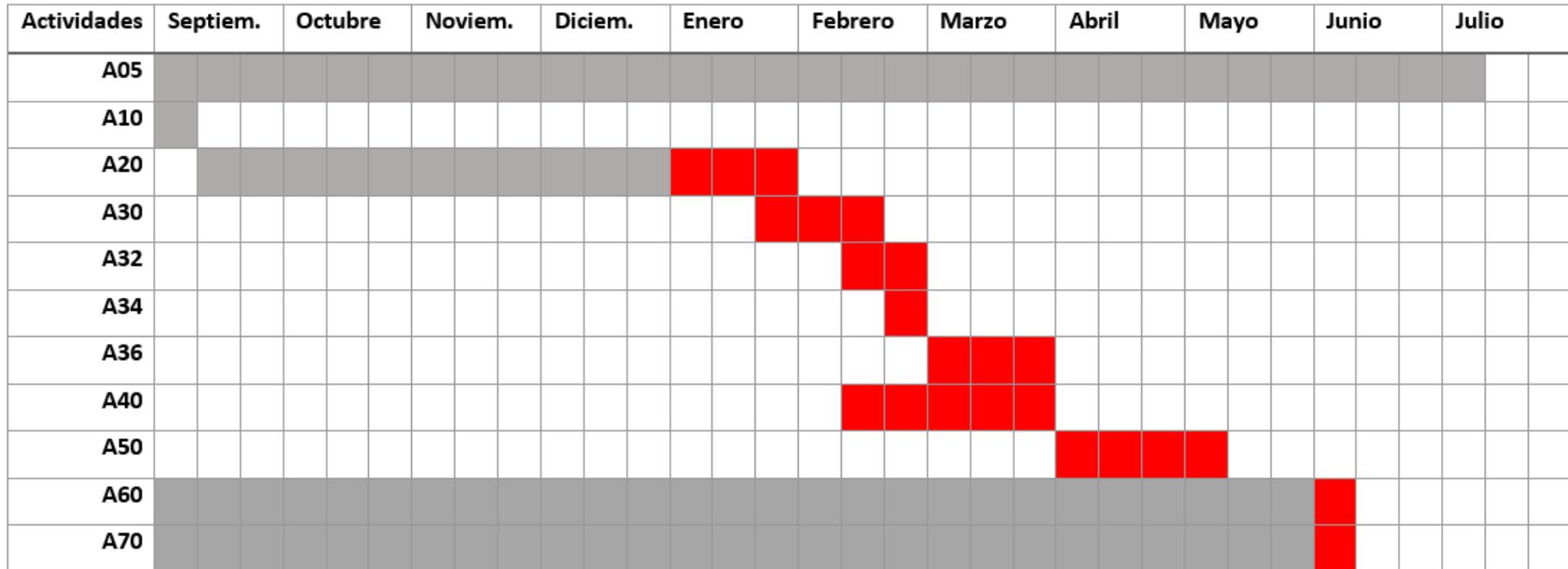


Ilustración 68: Diagrama de Gantt con desviaciones respecto a la planificación

El motivo del retraso en las actividades fue que durante el primer cuatrimestre de curso había asignaturas con las que compatibilizar el proyecto. El tiempo disponible aparte de las asignaturas no se calculó bien, por lo que hubo una baja actividad en el proyecto desde octubre hasta diciembre.

8 Conclusiones

Para terminar, vamos a ver las conclusiones a las que se ha llegado tras la realización del proyecto.

- Al igual que Internet, la publicidad está en constante evolución. Hoy en día existen muchas clases de publicidad y algunas son molestas u ofensivas por su contenido o por cómo se muestran. Además, la publicidad es cada vez más agresiva. Por esto, los anunciantes deben plantearse moderar el nivel de agresividad y controlar más el contenido que ofrecen para no minar aún más la confianza de los consumidores.
- El problema del adblocking está muy de moda y hay nuevos artículos de opinión cada poco tiempo acerca de este fenómeno.
- El problema actual del bloqueo de anuncios puede traer consigo el fin del Internet que conocemos. Hay que encontrar una solución que haga ganar a usuarios, autores de contenido y anunciantes. Y, esta solución, no debe demorarse mucho.
- Las extensiones para navegadores son un campo aún por explotar. Son una realidad y se utilizan cada vez más, pero hay muy poca información para poder comenzar a trabajar en este campo. Es necesario sentar unas bases que sirvan de suelo para los que quieran iniciarse en el ámbito de las extensiones.
- El uso de extensiones como AdBlock Plus se está disparando. De hecho, la progresión es ya de carácter geométrico. Por lo que a no ser que cambie algo, y muy pronto, la mayoría de los usuarios de Internet las usarán como método de protección ante la publicidad.
- Una vez entendido cómo funciona AdBlock Plus es mucho más sencillo pensar en soluciones que adapten su comportamiento para satisfacer a los distintos usuarios. Una idea interesante (además de la implementada) sería invertir el funcionamiento de AdBlock Plus. De esta manera, en vez de bloquear todos los anuncios de primeras e ignorar solo las excepciones que se encuentran en la lista blanca, pasaría a permitir todo salvo los elementos que se encuentren en la lista negra. Así, los usuarios darían una oportunidad a la web antes de decidir bloquear sus anuncios. Por otra parte, también puede ser muy interesante mejorar la interfaz de modificación de los filtros de ABP, ya que ahora mismo esta tarea no

es nada intuitiva. Hay que tener conocimientos avanzados para poder modificar manualmente los filtros ya que están expresados en una sintaxis muy concreta.

- La propuesta de mejora de AdBlock Plus propuesta en este proyecto da a los usuarios mayor control sobre lo que bloquean y da una oportunidad a los proveedores de contenido de obtener ingresos con los anuncios que realmente no molestan.

9 Referencias

AdBlock (2015). Recuperado de <https://getadblock.com/>

AdBlock Plus (2015). Recuperado de <https://adblockplus.org/>

AdBlock Plus (2015). *Permitir publicidad aceptable en Adblock Plus*. Recuperado de <https://adblockplus.org/es/acceptable-ads>

AdMuncher (2015). Recuperado de <https://www.admuncher.com/>

Alegsa (2015). *Ingeniería inversa*. Recuperado de <http://www.alegsa.com.ar/Dic/ingenieria%20inversa.php>

BBC. (2013). *Your views: Are targeted online adverts too intrusive?*. Recuperado de <http://www.bbc.com/news/business-22202406>

Bernal Arrieta, R. M., Márquez Rodríguez, A. C. (1997). *Psicología de la publicidad*. Nuevo León: Universidad Autónoma de Nuevo León

Código Penal Español [CP] Ley Orgánica 10 de 1995. Noviembre 23 de 1995 (España)

Dans, E. (2014). *La hora de la verdad para la publicidad en la web*. Recuperado de <http://www.enriquedans.com/2014/09/la-hora-de-la-verdad-para-la-publicidad-en-la-web.html>

Dans, E. (2015). *Snapchat Discover: la importancia del contenido*. Recuperado de <http://www.enriquedans.com/2015/01/snapchat-discover-la-importancia-del-contenido.html>

De la Encarnación Gabin, M. A. (2005). *Políticas de marketing*. España: McGraw-Hill.

Estatuto 61 de 1964. Por la que se aprueba el Estatuto de la Publicidad. Junio 11 de 1964. BOE N° 143.

Guzmán Elísea, J. (2003). *Desarrollo de campaña publicitaria*. Nuevo León: Universidad Autónoma de Nuevo León.

Hispalinux (2015). *¿Qué es el software libre?*. Recuperado de <http://hispalinux.es/SoftwareLibre>

Josemi (2010) *Gracias por no usar Adblock Plus*. Recuperado de <http://blog.josemi.net/2010/08/gracias-por-no-utilizar-adblock-plus/>

Mozilla (2015). Recuperado de <https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XPCOM/Reference/Interface>

Mozilla (2015). Recuperado de <https://dxr.mozilla.org/mozilla-central/source/toolkit/modules/LoadContextInfo.jsm>

Mozilla (2015). Recuperado de https://developer.mozilla.org/en-US/docs/Mozilla/JavaScript_code_modules

Ley de 1902. Propiedad industrial. Mayo 16 de 1902.

Ley 34 de 1988. General de Publicidad. Noviembre 11 de 1988. BOE N° 274.

Ley 7 de 1998. Sobre condiciones generales de la contratación. Abril 13 de 1998. BOE N° 89.

Ley 34 de 2002. De servicios de la sociedad de la información y de comercio electrónico. Julio 11 de 2002. BOE N° 166.

Palant, W. (2010). *The devastating effects of ad-blocking*. Recuperado de <https://adblockplus.org/blog/the-devastating-effects-of-ad-blocking>

Puromarketing (2014). *La inversión publicitaria en internet seguirá creciendo*. Recuperado de <http://www.puromarketing.com/66/18187/inversion-publicitaria-internet-seguira-creciendo.html>

Puromarketing (2015). *Internet y publicidad digital experimentan las mayores subidas en la inversión publicitaria*. Recuperado de <http://www.puromarketing.com/66/24053/internet-publicidad-digital-experimentaran-mayores-subidas-inversion-publicitaria.html>

RAE (2015). *Publicidad*. Recuperado de <http://lema.rae.es/drae/?val=publicidad>

Real decreto 1 de 1996 [Ministerio de Cultura]. Por medio del cual se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia. Abril 12 de 1996.

Real decreto 1 de 2007 [Ministerio de Presidencia]. Por medio del cual se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios y otras leyes complementarias. Noviembre 16 de 2007.

Snapchat (2014). *Advertising on Snapchat*. Recuperado de <http://blog.snapchat.com/post/100255857340/advertising-on-snapchat>

Soria Ruíz, R., (2013). La ingeniería inversa es legal en España. *Abanlex*. Recuperado de <http://www.abanlex.com/2013/12/la-ingenieria-inversa-es-legal-en-espana/>

Taskjs (2015). Recuperado de <http://taskjs.org/>

Vilches, A. (2012). Publicidad y cultura. *Revista Universitaria de Treballs Academics*, (4)

µBlock (2015). Recuperado de <https://github.com/gorhill/uBlock>

10 Anexo 1: Glosario de términos

En este apartado se introducirán una serie de términos cuyo significado no es obvio con el objetivo de facilitar la lectura del proyecto. Los términos, que están ordenados alfabéticamente, son los siguientes:

- **Algoritmo**: conjunto de instrucciones o reglas bien definidas y ordenadas que permite realizar una actividad mediante pasos sucesivos que conducen a obtener la solución a un problema. En este caso el problema es bloquear los anuncios.
- **API**: también conocida como interfaz de programación de aplicaciones, es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software.
- **Asíncrono**: que no tiene un intervalo de tiempo constante entre cada evento.
- **Categoría**: lista de pares nombre-valor, donde cada valor es una lista de cadenas de texto. Cada cadena de texto es lo que se llama **entrada**. Para acceder a ellas se debe hacer a través del **gestor de categorías** (categoryManager).
- **Clase**: es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje. Cada clase es un modelo que define un conjunto de variables y métodos apropiados para operar con dichos datos. Cada objeto creado a partir de la clase se denomina **instancia** de la clase.
- **Clase abstracta**: clase que no puede ser instanciada para crear un objeto y tienen al menos un método abstracto (no está implementado). Estas clases deben ser extendidas por otras de tal manera que implementen los métodos abstractos de la clase abstracta.
- **Clase concreta**: clase que puede ser instanciada para crear un objeto.
- **Debug**: también denominado depuración de programas. Es el proceso de identificar y corregir errores de programación.
- **DTD**: es un fichero que define la estructura de un documento XML.
- **Expresión regular**: es una secuencia de caracteres que forma un patrón de búsqueda. Con ellas se pueden expresar cosas como “todas las cadenas que empiezan o acaban por...”, o “todas las cadenas que contienen el carácter...”.
- **Firefox Sync**: es una aplicación que permite a los usuarios sincronizar sus marcadores, historial de navegación, preferencias, contraseñas, las últimas 25 pestañas abiertas y datos de formularios a través de múltiples dispositivos.

- **Hoja de estilo (style sheet):** son conjuntos de instrucciones, a veces en forma de archivo anexo, que se asocian a los archivos de texto y se ocupan de los aspectos de formato y de presentación de los contenidos: tipo, fuente y tamaño de letras, alineación y posicionamiento del texto, colores y fondos, etc. El lenguaje de hoja de estilo más conocido es el CSS (Cascading Style Sheets).
- **Hoja de estilo de usuario:** son hojas de estilo que puede aplicar el usuario mediante su navegador. Aunque la inmensa mayoría de usuarios no utiliza esta característica, en teoría es posible que los usuarios establezcan el tipo de letra, color y tamaño de los textos y cualquier otra propiedad CSS de los elementos de la página que muestra el navegador.
- **HTML:** es un lenguaje de marcado cuya base es XML. Se emplea en el desarrollo de páginas web y se encarga de estructurar la información proveyendo diferentes etiquetas para los elementos a representar. Unos ejemplos de etiquetas son <image> y <a> que sirven para imágenes y enlaces respectivamente.
- **IFrame:** es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Algo así como una página web dentro de otra página web.
- **Interfaz (programación):** contiene las definiciones de un grupo de funciones relacionadas que una clase pueda implementar.
- **Interfaz de usuario (user interface):** son elementos gráficos que permiten al usuario acceder a los contenidos, navegar e interactuar. Para lograr que un usuario se quede y vuelva, el diseño de la interfaz es importante.
- **Librería:** es un paquete que contiene varias funciones para lograr un propósito bien definido y específico. Estas librerías están diseñadas de tal forma que son fácilmente integradas a otros programas que requieren usar la funcionalidad que la librería ofrece. Las librerías no están pensadas para funcionar de manera autónoma, es decir, su fin es servir de apoyo para otros programas.
- **Listener:** es un objeto creado dinámicamente por código que se utiliza para detectar eventos.
- **Método get:** método que permiten acceder a propiedades de objetos.
- **Módulo:** es una porción de un programa. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas. Podría decirse que es un paquete.
- **Observador (observer):** es un objeto que se mantiene a la espera de que el elemento que está observando cambie. Cuando lo hace, se notifica para realizar las acciones necesarias.

- **Página HTML**: página web compuesta por elementos HTML.
- **Petición HTTP**: es un mensaje de texto creado por un cliente en un formato especial conocido como HTTP. **HTTP** (HyperText Transfer Protocol) es un formato basado en texto que permite a dos máquinas comunicarse entre sí.
- **Política de contenidos**: son básicamente una serie de reglas que establecen si un contenido es adecuado o no.
- **Propiedades exportadas**: propiedades que serán accedidas desde otro fichero.
- **Punto de ruptura**: son utilizados para el debug de programas. Introducir un punto de ruptura en una instrucción de un programa produce que el programa, cuando está en ejecución, se pause para poder ver el estado actual de las variables. También puede ser usado para saber si en ejecución del programa se accede a cierta sección del código.
- **Regla CSS**: estas reglas son cada una de las instrucciones que componen las hojas de estilo en lenguaje CSS.
- **Script**: programa usualmente simple, que por lo regular se almacena en un archivo de texto.
- **URI**: es una cadena de caracteres que identifica los recursos de una red de forma inequívoca. Un caso particular de URI son las URL, que, además de identificar el recurso, también permiten localizarlo.
- **Variable**: está formada por un espacio en el sistema de almacenaje y un identificador que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un valor.
- **XML**: es un meta-lenguaje que nos permite definir lenguajes de marcado adecuados a usos determinados. Un lenguaje de marcado sirve para representar información estructurada, de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos. Un par de ejemplos de lenguajes, que se ven en este documento, que tienen base en XML son XUL y HTML.