Chromium extension tutorial

Por supuesto que no estamos hablando del elemento químico cromo ("chromium" en inglés) sino del popular navegador web de código abierto para todos. Lo que desarrollaremos es una "extensión", una herramienta integrada al navegador y que tiene muy diferentes propósitos. Con fines didácticos haremos una extensión muy sencilla que nos redirige hacia una página web que nos devuelve simplemente nuestra propia dirección IPv4, ¡manos al teclado!

¿Qué es una extensión para Chromium?

[SVG: Chromium Web Browser Logo]

Como ya lo definimos al principio, una herramienta puede tener fines diversos, uno o más:

- Pueden ser usadas para notificarnos si tenemos nuevos mensajes de correo electrónico por medio del pequeño icono que se coloca al lado de la barra de direcciones.
- Si no hemos iniciado sesión en nuestra cuenta de correo-e basado en web, podemos hacer click en el icono para ir a la página donde introduciremos nuestras credenciales ***(¡Cuidado con las páginas falsas o "phishing"!).***
- Podría alertarnos si una palabra clave en particular aparece en la página que estemos visitando.
- Podríamos programarla para NO cargar las imágenes de un sitio determinado -o todos los que visitemos-.
- Tal vez sería útil para alertarnos de la modificación de una página (aunque para ellos hay lectores RSS específicos para ese trabajo) o su cambio de estado.
- ¡Incluso, si somos aventureros, nos permitiría programar una calculadora!

Como ven los usos son muchos y variados, nuestra imaginación no tiene límites *más sin embargo el entorno donde va a funcionar nuestra novel extensión sí que está supeditada al navegador* ***Chromium.*** ***Esencialmente una extensión es una pequeñita página web que está alojada en un navegador web - en este caso Chromium - y que además puede acceder a ciertas funciones API ("Application Programming Interface":*** conjunto de clases, métodos y funciones ***normalizadas).***

La que nosotros vamos a realizar es bastante modesta y a medida que avanzemos la iremos profundizando más, pero la idea es introduciros en esta área de programación en un entorno de desarrollo bastante peculiar.

Creando el proyecto.

Pues que para comenzar crearemos en nuestro disco duro, donde gustemos, una carpeta con el nombre **"KS7000_ConoceTuIPv4"** - o el nombre que vosotros querais, sed de libre albedrío, lo importante es que aprendamos los conceptos y procedimientos-.

Primer componente de nuestra extensión en Chromium.

Como somos inclinados al dibujo técnico más que al dibujo artístico (¡Ea, que por allí viene nuestro tutorial sobre **Inkscape**!) lo que hicimos fue redimensionar un logotipo en formato PNG (que ya teníamos rodando por internet) a un tamaño de 16 por 16 píxeles ***ya que necesitados una pequeña interfaz gráfica para el botón donde nuestros usuarios harán click y abrirán nuestro trabajo.*** *Cuando "subamos" - y especifiquemos- este icono a Chromium no tendremos que programar línea alguna: ya el navegador sabrá qué hacer al evento de hacer click sobre nuestra pequeña obra de arte microscópica (935 bytes "pesa" el fichero):*

Precisamente queremos hacer hincapié en este punto: esa imagen pequeñita que vamos a especificar ya Chromium le añade el evento **click** para cargar y ejecutar los siguientes componentes que vamos a crear, **eso es, en sí, una Interfaz de Programación de Aplicaciones**

"API" -normalizadas en un ambiente determinado (Chromium, en este caso) -.

Creando un archivo de manifiesto.

Un manifiesto es simplemente una declaración de propósitos o programas. Para este caso es un fichero llamado **manifest.json** y por su extensión deducireis que está en lenguaje **JSON: JavaScript Object Notation**, el cual permite el intercambio de información entre cliente y servidor. Aunque está basado en la norma ECMA-262 3ª edición de 1999 nosotros consideramos que es una norma *de facto.* En todo caso no representa mayor misterio, tal como lo es XML, el JSON (a menos que guarde datos binarios) es plenamente legible tanto a humanos como a máquinas, pasemos a analizar su estructura mínima necesaria:

```
{   "manifest_version": 2,   "name": "KS7000_ConoceTuIPv4",   "version":
"1.0", }
```

Explicado línea por línea, **los tres primeros elementos son obligatorios**:

1. **"manifest_version"**: toma el valor 2 -sin comillas- para indicar que es a partir de la versión 18 de Chromium que funcionará la extensión. Al momento de escribir este reportaje vamos por la versión 59 así que ya sabeis que la versión 1 está obsoleta e inútil, de hecho. *Notad cada coma para indicar separación de componentes, lo colocamos antes del fin de línea y retorno de carro.*
2. **"name"**: con hasta un máximo de 45 caracteres, es como identificaremos nuestra extensión y tratad que sea bien original y único si es que quereis algún día hacerlo famoso por internet y evitar malentendidos con otros programadores *o incluso poderosas corporaciones.*
3. **"version"**: pues la numeración que utilizemos para identificar los cambios que le hagamos, y aunque está entrecomillado *no os equivoquéis: son cuatro enteros de ntreo cero y 65535 separados por puntos* y sirven para identificar - y actualizar a nuestros usuarios si hacemos una mejora -. En nuestro ejemplo colocamos "1.0" **¿Dónde diablos están los otros dos números faltantes?** Pues en realidad nuestra versión es "1.0.0.0", tened mucho cuidado al numerar porque como dijimos se usa para comparar y actualizar, de ser necesario, a nuestros usuarios.

Más adelante ahondaremos en otras propiedades que *deberíamos* agregar y que implican otros ficheros en nuestra mini aplicación, per antes de finalizar esta sección os contamos que hay otras propiedades interesantes:

- **"description"**: la descripción de lo que realiza nuestra extensión, para lo que sirve, sed breve y conciso; entrecomillad.

- **"short_name**": que soporta hasta 12 caracteres y es especialmente útil para titular las pestañas y el lanzador de aplicaciones (en teoría).
- **"version_name"**: un nombre no necesariamente numérica, si queremos colocarle un nombre afectuoso, deberiamos tratar de que sea único.

Así las cosas nuestro archivo de manifiesto va quedando como sigue:

```
{   "manifest_version": 2,   "name": "KS7000_ConoceTuIPv4",   "short_name
": "KS7000_IP",   "description": "Esta extensión os permitirá saber vuest
ra dirección IPv4.",   "version": "1.0",   "version_name": "TuIPv4", }
```

¿Acciones de página o acciones de navegador?

Aunque dijimos que **Chromium** sabría qué hacer con nuestra extensión, esto no es totalmente cierto ya que en esta **API** desarrollaron dos aciones principales de la cual debemos especificar que nuestra extensión realize alguna de las dos (o ninguna: si es ninguna *pues no obtenemos ningún resultado, es una extensión inútil*).

- **Acción de navegador:** se ejecutará en cualquier pestaña y pa?ina que abramos *y permite que nuestros usuarios interactuen con nuestra extensión (ejecutar una orden o cambiar las preferencias, por ejemplo).*
- **Acción de página:** es un icono que indica un estado o condición a la página que estamos visitando (¿recuerdan que propusimos, al principio, una extensión que buesque y notifique la existencia de una palabra clave? *pues eso, se puede activar -o no- si esa palabra clave está presente*). Esta acción NO permite interactuar con los usuarios, es más bien informativa.

Nosotros usaremos la **acción de navegador** para lo cual debemos escribir:

```
"browser_action": {}
```

y colocamos entre los corchetes los elementos que queremos especificar y que a continuación especificamos:

```
 "default_icon": "KS7000_ConoceTuIPv4.png",  "default_popup": "aviso.html
"
```

La primera línea indicamos que utilice el ícono que os mostramos al principo, ***pero !ah,caramba! ¿habiais dicho que Chromium sabrá que hacer por nosotros?*** pues aquí el detalle no es culpa de **Chromium** ya que debido a los múltiples dispositivos (hardware) que pueden ejecutar este navegador web son muy diversos, *los tamaños de pantalla son muy diversos también* ***así que debemos -en teoría- hacer íconos de diferentes tamaños Y CHROMIUM SE ENCARGARÁ DE MOSTRAR EL QUE SEA DEL TAMAÑO ADECUADO AL DISPOSITIVO y tal como prometimos nos "harán" el trabajo a nosotros los programadores.***

Acá nos quedaremos con un solo ícono, queda para vosotros estudies cómo especificar más iconos. La segunda línea establece cuál archivo .html queremos mostrar cuando el usuario haga click, **y de nuevo es cuestión de tener variedad de opciones ya que podemos hacer varios ficheros .html (según la complejidad de nuestra extensión)** ***pero debemos decirle al navegador cual queremos que muestre primero, así solo sea una la que "subamos"*** ***¡NO ESTAMOS PROGRAMANDO EN PYTHON, TAMPOCO ESPERÉIS MILAGROS!***

Archivo de manifiesto, versión final, debidamente indentado.

```
{   "manifest_version": 2,   "name": "KS7000_ConoceTuIPv4",   "short_name
": "KS7000_IP",   "description": "Esta extensión os permitirá saber vuest
ra dirección IPv4.",   "version": "1.0",   "version_name": "TuIPv4",    "
browser_action": {     "default_icon": "KS7000_ConoceTuIPv4.png",     "de
fault_popup": "aviso.html"   }  }
```

Elemento emergente de interfaz.

Ya sabemos que nos falta ahora un elemento, el **"default_popup**" al cual le asignamos el nombre "**aviso.html**" y como este tema no es un tutorial de HTML5 pues de una os colocamos el código del fichero de marras:

```
# KS7000_ConoceTuIPv4

    ¡Conoce tu dirección IPv4!
```

> ¡ IMPORTANTE ! : declarad vuestro guion externo en JavaScript DESPUÉS de los elementos, de lo contrario arrojará un error "null" indicando que los elementos no existen. Esto es así debido a que el analizador "parser" va de arriba hacia abajo, línea por línea, y el proceso asíncrono que lanzamos debe tener con qué trabajar. En este caso hay dos elementos que "pasaremos": el botón identificado como "conoceIPv4" y el párrafo sin texto o "en blanco" identificado como"IPv4".

Ahora vamos a crear nuestro elemento escrito en JavaScript. un guión que, por ahora, simplemente colocará un mensajito de texto en el párrafo identificado como "IPv4":

```
document.getElementById("conoceIPv4").addEventListener("click", buscaIPv4
);   function buscaIPv4() {     document.getElementById("IPv4").innerHTML
 = "vuestra IPv4 es:";   }
```

Como veis es algo bien sencillo:

- En la primera línea declaramos que al **documento** identificado como "conoceIPv4" le agregue un evento que esté en función de si el usuario hace click en el botón y que ejecute la función "buscaIPv4".
- A su vez la función buscaIPv4 POR AHORA lo único que hace es sustituir al elemento (párrafo) con el identificador "IPv4" y lo sustituya con el mensaje "Vuestra IPv4 es:".

Al guardar y recargar con CTRL+R las extensiones instaladas va a volver a mostrarnos el mensaje de que los hash NO coinciden, copiamos el hash actualizado y lo pegamos en el fichero **manifiest.json** para volver a recargar y ejecutar, *podréis observar algo muy parecido a esto:*

¡LISTO, HEMOS CREADO UN PROCESO ASÍNCRONO! *MUY FÁCIL, ¿CIERTO? ?*

Creando

Comprobando compatibilidad en Mozilla Firefox.

Por medio de la cuenta Twitter de un desarrollador web de Mauritania tuvimos la excelente oportunidad de revisar cómo se desarrolla una extensión en Mozilla Firefox (versión 45 o superior)

y además cómo preparse para presentarla al público, todo muy detallado y bien pensado.

https://twitter.com/onlygan/status/852125379632541696

Como **Chromium** y **Firefox** *ambos* comparten el entorno de programación **chrome** hace que compartan la misma estructura de datos pero se "llega" de diferente manera. Primero debemos introducir en la barra de direcciones el siguiente comando:

```
about:debugging
```

Presionamos la tecla intro y seleccionamos complementos y hacemos click en el botón "**Cargar complementario temporario**" (sí, suena extraño en castellano pero si queremos colaborar con las traducciones, pues las puertas están abiertas). Seleccionamos la carpeta con nuestro mini proyecto y listo ¡a comprobar!

Fuentes consultadas.

En idioma francés.

- «Interface de programmation» au Wikipédia.

En idioma italiano.

- «JavaScript Object Notation» en Wikipedia.

En idioma inglés.

- «How to create a Chrome Extension in 10 minutes flat» by John Sonmez.
- «Extensions overview» at Google Chrome.
- «How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript» by Chris Williams.
- «CRX Package Format» at Chrome.
- «crxmake» at GitHub.
- «Chrome Extension - Content Security Policy - executing inline code»
- «Content Security Policy Level 2» by W3C.
- «Your first extension» at MDN.
- «I wanted real time #GitHub push notifications. So I built a Chrome extension.» by Stacy Goh.
- «How to Create and Publish a Chrome Extension in 20 minutes»