



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO E IMPLEMENTACIÓN DE UNA EXTENSIÓN (PLUGIN) PARA EL
NAVEGADOR WEB GOOGLE CHROME PARA ANÁLISIS Y EVALUACIÓN DE
ACCESIBILIDAD DE UNA PÁGINA WEB COMO COMPLEMENTO DE LA
HERRAMIENTA DE ANÁLISIS AUTOMÁTICO DE ACCESIBILIDAD
WEB DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

Trabajo de titulación previo a la obtención del
título de Ingeniero de Sistemas

AUTOR: GALO FELIPE GUARTATANGA INGA

TUTOR: ING CRISTIAN FERNANDO TIMBI SISALIMA, MSc.

Cuenca - Ecuador

2022

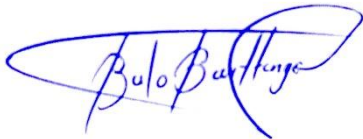
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Galo Felipe Guartatanga Inga con documento de identificación N° 0105236772, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 03 de octubre de 2022

Atentamente,



Galo Felipe Guartatanga Inga
0105236772

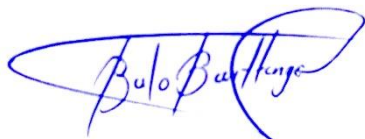
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Galo Felipe Guartatanga Inga con documento de identificación N° 0105236772, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: “Desarrollo e implementación de una extensión (plugin) para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 03 de octubre de 2022

Atentamente,



Galo Felipe Guartatanga Inga
0105236772

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Cristian Fernando Timbi Sisalima con documento de identificación N° 0103709911, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el proyecto técnico: DESARROLLO E IMPLEMENTACIÓN DE UNA EXTENSIÓN (PLUGIN) PARA EL NAVEGADOR WEB GOOGLE CHROME PARA ANÁLISIS Y EVALUACIÓN DE ACCESIBILIDAD DE UNA PÁGINA WEB COMO COMPLEMENTO DE LA HERRAMIENTA DE ANÁLISIS AUTOMÁTICO DE ACCESIBILIDAD WEB DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, realizado por Galo Felipe Guartatanga Inga con documento de identificación N° 0105236772, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 03 de octubre de 2022

Atentamente,



Ing. Cristian Fernando Timbi Sisalima, MSc.
0103709911

Dedicatoria

Dedico este trabajo con todo cariño y amor a mis padres, Aida del Rosario y Galo Gustavo por su apoyo constante y paciencia me han permitido llegar a culminar una meta más en mi vida, gracias por enseñarme que con esfuerzo y perseverancia todo se puede lograr.

A la memoria de mi abuelita Ana María Uyaguari que desde el cielo ha venido guiando cada paso que daba para culminar este trabajo, además que en vida ella me apoyó incondicionalmente y me inculcó el valor de la responsabilidad

A Gabriela Granda Tonato porque con sus consejos y palabras de aliento han hecho de mi una mejor persona y de una u otra forma me acompañaron en todos mis sueños y metas.

Finalmente quiero dedicar esta tesis a todas mis amigas y amigos, por apoyarme cuando más las necesito, por extenderme su mano en momentos difíciles y por el cariño brindado día a día.

Galo

Cuenca, octubre del 2022

Agradecimientos

El presente trabajo de tesis primeramente me gustaría agradecer a nuestro Dios por bendecirme, para llegar hasta donde he llegado y porque hice realidad este sueño anhelado.

De manera especial quiero agradecer a mi padre y mi madre quienes me supieron dar su ejemplo de trabajo y honradez y me dieron el apoyo que necesitaba para no decaer cuando todo parecía complicado e imposible.

A mi tutor de tesis, que con esfuerzo y dedicación supo guiarme correctamente para el desarrollo del presente trabajo.

A la Universidad Politécnica Salesiana por haberme abierto las puertas para seguir mis estudios, y poder cumplir con mis metas propuestas.

A todas aquellas personas que de una u otra manera colaboraron para hacer de este sueño una realidad.

Galo Felipe Guartatanga Inga

ÍNDICE DE CONTENIDOS

1.	Introducción	1
2.	Problema	1
3.	Objetivos	1
3.1.	Objetivo general	2
3.2.	Objetivos específicos	2
4.	Fundamentos teóricos	2
4.1.	Analizador Automático Web	3
4.2.	Api	7
4.3.	Api rest	8
4.4.	Json	8
5.	Marco Metodológico	9
6.	Resultados	9
6.1.	Adecuaciones de la Api	10
6.2.	Desarrollo del plugin	16
6.3.	Análisis de resultados del plan de pruebas	17
7.	Cronograma	20
8.	Presupuesto	23
9.	CONCLUSIONES	24
10.	RECOMENDACIONES	24
	REFERENCIAS	24
	ANEXOS	26

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de archivo Manifest.json	5
Figura 2. Menú Extensiones de Google Chrome	6
Figura 3. Página principal del objeto de aprendizaje	11
Figura 4. Página principal del objeto de aprendizaje	12
Figura 5. Ejemplo de una petición en el software Postman	13
Figura 6. Respuesta de una petición en formato JSON (resumen)	14
Figura 7. Respuesta de una petición en formato JSON (reglas)	15
Figura 8 Prototipo de la pantalla principal de la extensión	20
Figura 9 Lista de actividades realizadas	22

ÍNDICE DE TABLAS

Tabla.1. Página principal del objeto de aprendizaje	8
Tabla 2. Análisis de resultados del plan de pruebas	19
Tabla 3. Cronograma de actividades	21
Tabla 4. Presupuesto real del proyecto	22

Desarrollo e implementación de una extensión (plugin) para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana

RESUMEN

El objetivo principal de este proyecto es incentivar a los desarrolladores a construir sitios web con estándares de accesibilidad web, brindando una herramienta adicional como lo es el Plugin de accesibilidad web para el navegador Google Chrome.

El plugin ayudará a los desarrolladores web a identificar el cumplimiento o no de normativas WCAG de manera más rápida y amigable.

Palabras clave.

W3C: Acrónimo de World Wide Web Consortium, es un consorcio internacional en donde organizaciones, usuarios y empleados desarrollan estándares y especificaciones para la web[22].

WCAG: Acrónimo de Web Content Accessibility Guidelines, define pautas y recomendaciones con el objetivo de hacer la web mucho más accesible [23].

Extensión o plugin: es una aplicación que agrega una funcionalidad adicional o una nueva característica al navegador web[27].

ABSTRACT

The main objective of this project is to encourage developers to build websites with web accessibility standards by providing a Web Accessibility Plugin for the Google Chrome browser.

The plugin will help web developers to identify and comply WCAG regulations in a faster and more friendly way.

Keywords

W3C: Acronym for World Wide Web Consortium, is an international consortium where organizations, users and employees develop standards and specifications for the web[22].

WCAG: Acronym for Web Content Accessibility Guidelines, defines guidelines and recommendations with the aim of making the web much more accessible[23].

Extensión o plugin: Is an application that adds additional functionality or a new feature to the web browser[27].

1. INTRODUCCIÓN

Actualmente se observa una gran cantidad de avances tecnológicos que van en beneficio de mejorar el estilo de vida de las personas y de manera especial a aquellas personas con capacidades especiales[27]. Con el fin de mejorar su experiencia y uso de la información que contiene la Internet, en especial con las páginas que prestan servicios en línea (educación, banca virtual, compras, ventas entre otras) se han creado normativas que apunta a un mejor desarrollo de sitios y contenidos web, las cuales beneficiaran a un gran segmento de personas que presentan dificultades al momento de utilizar la Internet [1].

2. PROBLEMA

Hay que reconocer que en el mundo informático, específicamente en el desarrollo web, en su mayoría no toma en cuenta a personas con capacidades especiales, llevando consigo algunas dificultades para navegar dentro de un sitio e interactuar de manera correcta con la misma, consecuentemente acceder a la web se convierte en una limitante para dichas personas [30].

Actualmente se encuentra en funcionamiento el “Analizador Automático de Accesibilidad Web”, básicamente es un sistema informático capaz de analizar el código de una página web y devolver un listado de fallos encontrados con sus respectivas sugerencias.

Dicho analizador tiene la necesidad de abarcar y llegar a más evaluadores o desarrolladores (usuarios) con el objetivo de que se logre construir mayor cantidad de sitios web con las Directrices de Accesibilidad para Contenido WEB WCAG 2.0; dicha metodología describe el procedimiento a seguir y las consideraciones necesarias para guiar a los usuarios, promover buenas prácticas o evitar errores comunes que se dan al construir sitios web[1].

Se pretende crear una herramienta adicional a la que ya se dispone en la actualidad que brinde mayor facilidad de uso, mostrando el análisis de un sitio web directamente en el navegador con el uso de una extensión para el navegador Google Chrome.

Desarrollar una extensión para el navegador Google Chrome, uno de los navegadores más populares en la actualidad, ayudaría en gran medida a realizar un análisis rápido y eficiente de un sitio en proceso de construcción con normas de accesibilidad, sin la necesidad de ingresar al observatorio web y registrar la url de un determinado sitio.

Los usuarios podrían hacer testing directamente en su navegador, comprobar errores o seguir las recomendaciones dadas mientras se está construyendo un sitio web. Con

esta iniciativa se puede llegar a conseguir que más sitios web puedan ser accesibles para personas con discapacidad.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar e implementar una extensión (plugin) para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana

3.2 OBJETIVOS ESPECÍFICOS

- 1.1.1. Estudiar y desarrollar un objeto de aprendizaje que documente el proceso de creación y publicación de una extensión (plugin) para Google Chrome.
- 1.1.2. Examinar la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana y realizar las adecuaciones que sean necesarias para que el API se adapte a los requerimientos del proyecto.
- 1.1.3. Desarrollar la extensión de evaluación de accesibilidad de páginas web que haga uso del API de la herramienta de Análisis Automático de Accesibilidad WEB de la UPS
- 1.1.4. Publicar el plugin desarrollado para su descarga y uso desde un navegador web Chrome.

4. FUNDAMENTOS TEÓRICOS

4.1. Analizador Automático Web. El Analizador Automático Web fue desarrollado para realizar una revisión del nivel de accesibilidad (A, AA, AAA) que posee un sitio Web[2]. Para poder llegar a obtener este resultado se tomó en cuenta los parámetros que se tiene como referencia en la normativa “NTE INEN ISO/IEC 40500:2012”, en esta se encuentran contenidas las Directrices de Accesibilidad para el contenido Web de la W3C, estas fueron aplicadas en el Analizador Web [2].

La WCAG 2.0 dispone de algunas directrices sobre la accesibilidad web que se detallan a continuación:“

1. **Perceptible:** *la información y los componentes de la interfaz de usuario deben ser mostrados a los usuarios en formas que ellos puedan entender.*
2. **Operable:** *Los componentes de la interfaz de usuario y la navegación debe ser manejable*

3. **Comprensible:** *la información y las operaciones de usuarios deben ser comprensibles.*
4. **Robustez:** *el contenido debe ser suficientemente robusto para que pueda ser bien interpretado por una gran variedad de agentes de usuario, incluyendo tecnologías de asistencia” [3].*

Además, la WCAG 2.0 exige ciertas normativas que se las conoce como nivel de conformidad, las cuales se detallan a continuación:

1. Nivel de conformidad:”

- a. Nivel A: Es el mínimo a cumplir, la página web debe satisfacer todos los criterios de conformidad del nivel A o proporcionar una versión alternativa conforme.
- b. Nivel AA: Para lograr conformidad con el Nivel AA, la página web debe satisfacer todos los Criterios de Conformidad de los Niveles A y AA, o proporcionar una versión alternativa conforme al Nivel AA.
- c. Nivel AAA: Para lograr conformidad con el Nivel AAA, la página web debe satisfacer todos los Criterios de Conformidad de los Niveles A, AA y AAA, o proporcionar una versión alternativa conforme al Nivel AAA”[24].

2. Páginas web completas: La norma pide estrictamente que los niveles de conformidad se apliquen a absolutamente todo el sitio web sin excluir ninguna parte de la página del mismo[23].

3. Procesos completos: En el caso de una página web tenga una secuencia de pasos para realizar cierta actividad, por ejemplo, registrarse, llenar un formulario, ingresar datos personales, etc y a su vez estos pasos se encuentren distribuidos en varias páginas web, dichas páginas de este proceso deben cumplir con el mismo nivel de conformidad (A , AA, AAA) sin excepción, no es posible lograr conformidad con un nivel en particular si una de las páginas del proceso no cumple con ese nivel o uno superior [24].

4. Uso exclusivo de tecnologías de modo compatible con la accesibilidad: Es necesario que todas las tecnologías usadas en la página web sean compatibles con la accesibilidad para satisfacer los criterios de conformidad[18].

Existen dos métodos que pueden ayudar a los evaluadores, diseñadores y desarrolladores a valorar los contenidos Web para garantizar que sean accesibles: revisiones con herramientas de evaluación automática y controles manuales[5].

Las herramientas de evaluación de la accesibilidad de la Web poseen una gran ventaja, ya que pueden reducir significativamente el tiempo y el esfuerzo necesarios para llevar a cabo las evaluaciones, estas herramientas, con su correcto uso durante las fases de diseño, implementación o mantenimiento del sitio Web, pueden ayudar a prevenir barreras de accesibilidad, reparar obstáculos encontrados y mejorar la calidad general de los sitios Web [5].

Las herramientas de accesibilidad web están basados en los estándares de accesibilidad que se pueden automatizar para dar una estimación del cumplimiento de las normas de accesibilidad y su reglas expuestas. Pero hay que considerar que no todas las normas de los estándares de accesibilidad son automatizables, por ejemplo en el caso de una imagen y su texto alternativo, la herramienta verifica de manera automática que exista, pero es necesario que se verifique manualmente que dicho texto corresponde a la imagen que se está mostrando y este a su vez sea lo suficientemente descriptivo[19].

Como se mencionó anteriormente, las herramientas de evaluación automática de la accesibilidad Web son un método rápido y oportuno, pero para conocer el grado de cumplimiento real de las recomendaciones de diseño es necesario complementarlo con el juicio personal de un experto[5]. Lo ideal es que se realice una propuesta metodológica que combine técnicas manuales y automáticas para evaluar fidedignamente el nivel de accesibilidad de los sitios Web [8].

Plug-in (Extensión): “Son pequeños programas auxiliares o dispositivos de hardware que permiten a sistemas mayores extender sus capacidades normales o aportar una función, generalmente muy específica, de manera que no se afecten las funciones ya existentes ni se complique el desarrollo del programa principal” [4].

Los Plug-ins (Extensiones) en navegadores, es un software empaquetado que le agrega características o nuevas funcionalidades al navegador. En la actualidad el uso de plugins en navegadores web se han venido utilizando como un apoyo para el funcionamiento de aplicaciones web que tienen contenido como música, videos o juegos interactivos que requieren de código en la parte del cliente. El objetivo es facilitar al usuario utilizar un recurso adicional dentro del navegador [11].

Plug-in para Chrome. En el caso de Google Chrome a estos plugins se los conoce como extensiones, la interfaz de usuario es minimalista y de fácil instalación.

Las extensiones tienen todos sus archivos fuente agrupados en un sólo archivo que el usuario descarga e instala, este archivo cuya extensión es .crx es un

paquete comprimido que contiene todos esos archivos HTML, CSS, JavaScript y todos los recursos que sean necesarios tales como imágenes o audio[7].

A continuación se detalla la arquitectura de una extensión de Chrome:

1. “Manifest.json”:Un archivo en formato json que contiene las configuraciones de dicha extensión [7].
2. Uno o varios archivos “.html” que contendrá la interfaz gráfica de la extensión [7].
3. Uno o más archivos de JavaScript que contenga la funcionalidad y la lógica de la extensión [7].
4. Archivos de recursos adicionales necesarios para la extensión, como por ejemplo imágenes [7].

Como se indicó anteriormente el archivo de manifiesto (manifest.json), es un archivo donde se especifica cada una de las configuraciones requeridas para el plugin.

```
{  
  "manifest_version": 2,  
  "name": "OAW",  
  "version": "1.0",  
  "description": "Observatorio de accesibilidad web",  
  "devtools_page": "devtools.html",  
  "background": {  
    "scripts": ["background.js"],  
    "persistent": false  
  },  
  
  "browser_action": {  
    "default_icon": "icon_128.png"  
  },  
  
  "web_accessible_resources": ["background.js"],  
  
  "icons": {  
    "16": "icon_16.png",  
    "48": "icon_48.png",  
    "128": "icon_128.png"  
  },  
  "permissions": ["activeTab", "background"]  
}
```

Figura.1. Ejemplo de archivo Manifest.json

En este archivo se especifica el número de versión “manifest_version”, el nombre de la extensión y una descripción detallada que indique de qué se trata dicha extensión, así como los iconos en diferentes tamaños (16,48,128) pixeles ,se

incluye archivos javascript que sean necesarios, en este caso se incluyó el archivo "background.js", los datos que son obligatorios son "name" y "versión" (el cual debe ser un número distinto y por cada actualización se debe incrementar el valor en 1 para que sea aceptado en la Google Chrome Store). Los demás parámetros del archivo manifest pueden ser opcionales dependiendo las necesidades de nuestra extensión.

Por otro lado, debemos indicar con la directiva 'permissions' las combinaciones de URLs válidas para interactuar con nuestra extensión, estos permisos ayudan a establecer un marco de seguridad en el cual se ejecutará nuestra extensión, protegiéndola así de páginas que pudieran vulnerar nuestro código[20].

Una extensión está formada por al menos un archivo de manifiesto y una o varias páginas más albergados todos estos archivos bajo una carpeta única. Cuando se distribuye la extensión, el contenido de esta carpeta se empaqueta en un archivo especial comprimido ZIP que tiene un sufijo .crx[7]. Chrome tiene una opción para facilitar este trabajo a los desarrolladores de extensiones, ya que provee un interfaz para la carga de extensiones con base en carpetas que en ellas tengan un manifest.json, y en definitiva contengan una extensión, sin la necesidad de tener que empaquetar en un .crx para instalar.[7]

Para acceder a esta opción antes mencionada es necesario indicar en la barra de direcciones la siguiente url: "chrome://extensions/", una vez ahí, aparecerá la lista de extensiones que se tiene instaladas y adicional aquello se tendrá disponible 3 botones como se muestra a continuación.

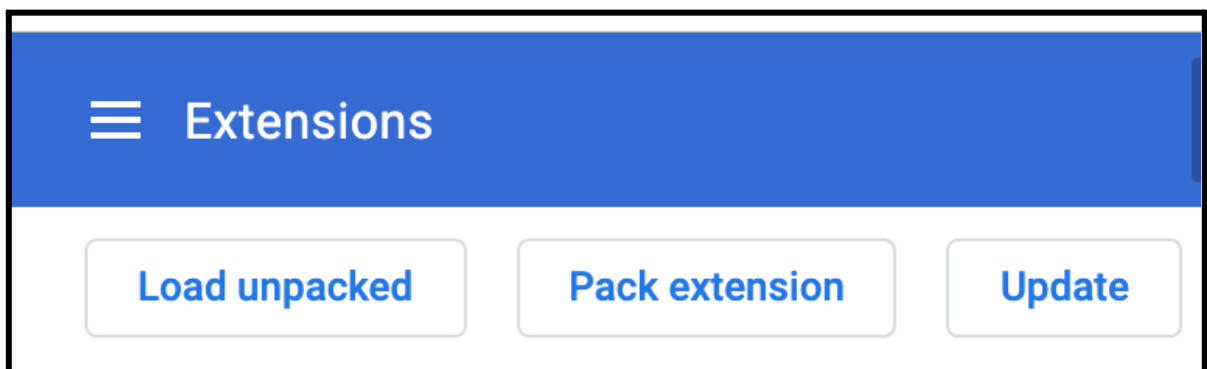


Figura.2. Menú Extensiones de Google Chrome

Load unpacked: A través de este botón se puede testear rápidamente una extensión, ya que se instala en el navegador y se puede probar de manera rápida lo que se esté desarrollando.

Pack extensión: Esta opción sirve para generar un archivo “.crx” con su respectiva firma digital para subir a la Chrome Web Store.

Update: Este botón es de gran ayuda mientras se está desarrollando, ya que permite actualizar los cambios realizados en los archivos y mostrarlos en el navegador.

Plug-ins de Accesibilidad Web: Actualmente, existen distintas páginas que realizan un análisis de accesibilidad con evaluaciones heurísticas, dichas páginas disponen de extensiones para diferentes navegadores que facilita su uso. En la Internet se pueden encontrar numerosas herramientas que permiten hacer una evaluación automática de accesibilidad a un sitio Web, en las que destacan páginas como WAVE, AXE, TAW, Examinator, entre otras[10]. Estas herramientas hacen una evaluación rápida y sistemática generando un resultado inicial de accesibilidad. En el siguiente gráfico se muestra una tabla comparativa de las distintas herramientas de evaluación de accesibilidad web con sus valoraciones por niveles de conformidad, este análisis fue realizado en julio 2017 [10].

Lista de herramientas de evaluación de accesibilidad web, con sus valoraciones total por niveles de conformidad, julio 2017				
#	Herramientas	Calificación Nivel de conformidad		
		A	AA	AAA
1	TAW	67%	49%	35%
2	Analizador Web Ecuatoriano	62%	40%	23%
3	Functional Accessibily Evaluator	62%	31%	13%
4	aChecker	60%	69%	31%
5	Ainspector	59%	31%	17%
6	Access Monitor	57%	38%	25%
7	eXaminator	57%	31%	25%
8	Vamolá	53%	62%	32%
9	Tingtun Accessibily Checker	51%	22%	0%
10	aXe	49%	13%	0%
11	Total Validator	46%	7%	8%
12	SortSite	42%	47%	13%
13	Tenon	41%	0%	4%
14	Wave	37%	18%	0%
15	Cynthia Says	36%	31%	26%
16	WCAG Contrast Checker	0%	7%	4%
17	Juicio Studio Toldbar	0%	7%	4%

Tabla.1. Página comparativa de herramientas de accesibilidad web.

- 4.2. API.** Una API es una abstracción que describe una interfaz para la interacción con un conjunto de funciones utilizadas por los componentes de un sistema de software[6]. El software que proporciona las funciones descritas por un API se dice que es una implementación de la API [6].
- 4.3. API REST.** Se accede a través del protocolo HTTP. Su comunicación es en una sola dirección por medio de direcciones web en la que enviamos los datos de nuestra consulta, como respuesta a la consulta sobre la API REST se obtienen datos en diferentes formatos, como pueden ser texto plano, XML, JSON, entre otros[25].

Para acceder y manipular los recursos, HTTP nos dota de los siguientes métodos con los cuales debemos operar:

- GET: Para consultar y leer recursos
- POST: Para crear recursos
- PUT: Para editar recursos
- DELETE: Para eliminar recursos [12].

- 4.4. JSON.** Es un formato ligero basado en texto, trabaja de forma independiente del lenguaje, cuya sintaxis está basada en un subconjunto de funcionalidades y palabras reservadas del lenguaje JavaScript, debido a su sencillez es fácil generar y procesar un documento con este formato; Esta característica hace de JSON un lenguaje ideal para el intercambio de datos entre aplicaciones cliente servidor o aplicaciones que se ejecuten en diferentes escenarios o lenguajes[25].

Un formato JSON está constituido por dos estructuras básicas: la primera es una colección de pares denominadas clave-valor, en otros lenguajes populares tales como java, python o Ruby se conocen como diccionarios, tablas, hash Maps, arreglos, etc.

Todas estas estructuras antes mencionadas son usadas para construir los mensajes en formato JSON, estos mensajes son interpretados por el cliente y servidor y codificados de acuerdo al protocolo interno definido, para lo cual se requiere el uso de librerías que permitan interpretar la sintaxis propia del lenguaje JSON[21]. En el caso de JAVA es muy popular utilizar la librería GSON para el intercambio de mensajes[21].

5. MARCO METODOLÓGICO

Para el cumplimiento de los objetivos propuestos se utilizó una metodología llamada Scrum, ya que ofrece grandes ventajas para el desarrollo de cualquier tipo de proyecto de software y un gran impacto al finalizar el mismo, básicamente la manera que se trabaja con esta metodología es de forma iterativa e incremental[26]. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 1 y 4 semanas, obteniendo como resultado una versión del software con nuevas funcionalidades listas para ser usadas, en cada nuevo sprint se va ajustando la funcionalidad ya construida y se añaden nuevas funcionalidades, priorizando siempre aquellas que aporten mayor valor al producto final [9].

Por lo tanto, en primera instancia se recopila la lista de características que va a tener el producto final (Product Backlog), una vez concluido, se definirán la tarea a desarrollar del proyecto con su respectiva prioridad (Sprint Backlog), de esta manera se dará inicio al desarrollo del producto [14].

Una de las características de los sprints es que sus entregables mejoran progresivamente conforme se lo vaya iterando, así como también se recibirá retroalimentación continua en el proceso de desarrollo del producto hasta que cumplan con el criterio de aceptación definido en un principio.

De esta manera se asegura que se tendrá bajo control el avance del proyecto para cumplir con el tiempo establecido y los objetivos planteados, obteniendo como resultado final un producto de calidad.

Una vez definido la metodología de desarrollo de software (Scrum) para este proyecto, se investigó acerca de cómo empezar con el desarrollo del mismo, conocimientos previos requeridos, herramientas necesarias, entre otras. Para iniciar con el desarrollo fue necesario tener conocimientos básicos de html, css y javascript.

En el caso de javascript se utilizó un framework llamado ReactJS, para optimizar el tiempo de trabajo reutilizando componentes.

React es una framework para javascript desarrollado por Facebook, actualmente se encuentra en la versión 1.7, una de las características más importantes de esta librería es que hace el conjunto mínimo de cambios necesarios en el DOM, dado que utiliza un nuevo concepto llamado el DOM virtual que básicamente es una representación virtual de la página donde se almacena en el estado actual, React lo que hace es compararlo con el estado actual del DOM del navegador y encontrar las diferencias y manipular la menor cantidad posible, manteniendo así los datos actualizados y mejorando el rendimiento[15].

Además, se requirió de un editor de texto o un IDE que ayude en el desarrollo del mismo, en este caso se eligió Visual Studio Code, el cual es un IDE muy popular dentro del ámbito del desarrollo web.

Algo requerido previamente es conocer las Herramientas de desarrollo de Google Chrome en la cual muestra algunas pestañas destacando entre las más importantes la pestaña de Elementos y Consola, además es aquí donde se aloja la extensión desarrollada en este proyecto.

6. RESULTADOS

6.1. Estudiar y desarrollar un objeto de aprendizaje que documente el proceso de creación y publicación de una extensión (plugin) para Google Chrome.

En este punto se creó un objeto de aprendizaje con ayuda de la herramienta EXELEARNING, que es un software gratuito de código abierto que permite la creación de contenidos educativos de manera muy sencilla y práctica[16].

Los objetos de aprendizaje son generalmente entendidos como entidades digitales que se pueden distribuir por Internet, significando esto que cualquier número de personas puede acceder y usarlos, los objetos de aprendizaje son fundamentales para el presente y futuro de la educación a distancia, ya que proveen la forma de crear contenidos educativos en línea [13].

Cabe destacar que el objeto de aprendizaje es importante para este proyecto ya que permite:

1. El contenido puede ser interactivo y de fácil entendimiento para quien desee usarlo.
2. Portabilidad del mismo, con la posibilidad de catalogar los contenidos y publicarlos en diferentes formatos, como html o pdf [16].

La página oficial de esta esta herramienta es la siguiente: <http://exelearning.net/>

El plugin es compatible con la versión 30 o superior de Google Chrome.

El plugin se encuentra disponible para todos los usuarios que deseen utilizarlo en la Chrome Web Store, su última versión es la “1.32”.

El resultado de la creación del objeto de aprendizaje es el siguiente:

Menú Siguiete »

Como crear un plugin para el navegador Google Chrome

INSTRUCTIVO PARA CREAR UNA EXTENSION O PLUGIN PARA EL NAVEGADOR GOOGLE CHROME

Instructivo para crear una extension o plugin para el navegador Google Chrome

- Objetivos
- Mapa Conceptual
- Introduccion
- Bloque de Contenidos



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)

Figura 3. Página principal del objeto de aprendizaje.



Figura 4. Menu "Objetivos" del objeto de aprendizaje.

6.2. Examinar la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana y realizar las adecuaciones que sean necesarias para que el API se adapte a los requerimientos del proyecto.

Para cumplir con este objetivo, en primera instancia fue necesario realizar la conexión a la API del Observatorio de accesibilidad Web para su respectivo análisis, para realizar esto es necesario conocer cuales son los parámetros de entrada, los cuales deben ser enviados en formato JSON.

- **Host:** Dirección web del servidor en el cual está alojada la API, en este caso es:
["http://observatorioweb.ups.edu.ec/owa/srv/wcag2"](http://observatorioweb.ups.edu.ec/owa/srv/wcag2)
- **Path:** Identificador del servicio web, en este caso es: "/análisis"
- **Width:** Ancho de la pantalla desde la cual se realiza la petición, la unidad de medida en la que se debe enviar es en pixeles por defecto es "1366"
- **Height:** Alto de la pantalla, por defecto es "768"
- **Token:** Código único generado en cuenta personal del usuario.
- **Lang:** Idioma en el desea que se le de respuesta.
- **Url:** URL de la página web a analizar.

Para conectarse a la API y realizar las respectivas pruebas se utilizó un software que brinda soporte para la conexión de APIs como lo es Postman, esta herramienta permite conectarse a APIs de manera muy rápida y sencilla.

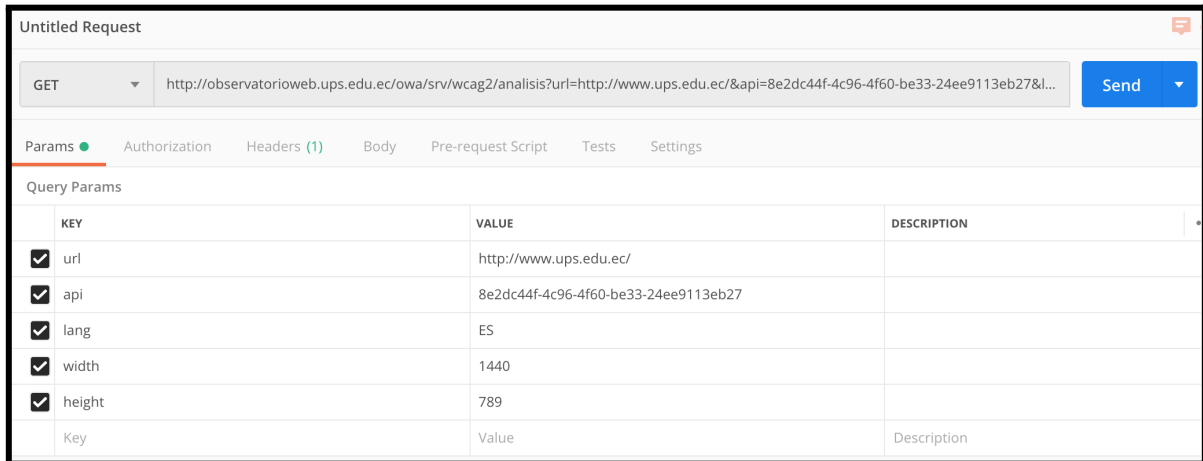


Figura 5. Ejemplo de una petición en el software Postman

Los parámetros de salida que retorna son también en formato JSON y tiene la siguiente información dividida básicamente en dos partes que son:

- **Resumen general de la petición.**

En esta sección se muestra un resumen del análisis realizado, indicando qué página web fue analizada, el número elementos como tablas, formularios, archivos de audio, etc que en dicha página web fueron encontrados.

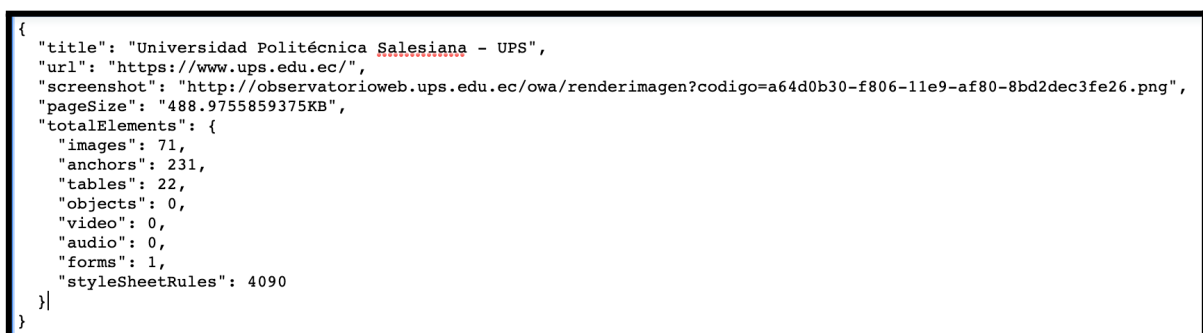


Figura 6. Respuesta de una petición en formato JSON (resumen)

- **Reglas de la norma OAW:**

En esta sección se muestra ya el análisis en sí de la página web, devolviendo información como:

- **Code:** Indica el código del elemento analizado de acuerdo con la norma WCAG 2.0
- **title:** Indica el título del elemento analizado de acuerdo con la norma WCAG 2.0
- **description:** Muestra una descripción detallada de lo anteriormente expuesto.
- **evaluated_elements:** Muestra un resumen del total de elementos analizados clasificados en correctos, incorrectos, advertencias y errores.
- **related_criteria:** Criterios relacionados a la técnica WCAG 2.0 analizada.
- **related_techniques:** Técnicas relacionadas a la norma WCAG 2.0 analizada.
- **results.** Resultados de la página en base a la técnica analizada

```

{
  "rules": [
    {
      "code": "OAW11",
      "title": "Proporcionar un título usando el elemento de título",
      "description": "Todos los documentos HTML y XHTML, incluidos aquellos en marcos individuales en un conjunto de marcos, tienen un elemento",
      "evaluation": "En la fuente del documento HTML o XHTML se verifica que exista un elemento <title> no vacío en la cabecera.",
      "evaluated_elements": {
        "total": 1,
        "ok": 0,
        "okr": 1,
        "war": 0,
        "err": 0
      },
      "related_criteria": [
        {
          "code": "2.4.2",
          "title": "Páginas tituladas (A)",
          "url": "https://www.w3.org/TR/WCAG20/#contents"
        }
      ],
      "related_techniques": [
        {
          "code": "H25",
          "title": "Proporcionando un título usando el elemento title",
          "url": "https://www.w3.org/TR/WCAG20/#contents"
        },
        {
          "code": "F25",
          "title": null,
          "url": "https://www.w3.org/TR/WCAG20-TECHS/F25.html"
        }
      ],
      "results": [
        {
          "resultCode": "OKR",
          "selector": "html:nth-child(1) > head:nth-child(2) > title:nth-child(1)",
          "source": "<title>Universidad Politécnica Salesiana - UPS</title>",
          "comment": "La página tiene título. Asegúrese que éste sea descriptivo y que esté acorde al contenido de la página",
          "question": "¿El título de la página describe y está acorde a su contenido?"
        }
      ]
    }
  ]
}

```

Figura 7. Respuesta de una petición en formato JSON (reglas)

Luego de una revisión del API del observatorio de Accesibilidad web, se llegó a la conclusión que requería de algunas adecuaciones, con el objetivo de lograr mostrar de manera más óptima y precisa la información de análisis de una determinada página.

Uno de los cambios más importantes fue necesario que la API del observatorio de Accesibilidad web devuelva un *selector*, básicamente un selector es una función de la librería de javascript JQuery [29], la cual obtiene un path de donde está ubicado un elemento dentro del DOM de la página html.

Este selector lo que hace es ayudar a identificar el elemento y mostrarle exactamente donde está ubicado, resaltando con un color diferente. Para el resaltado de la imagen se utilizó la librería highlight de JavaScript.

Un selector tiene el siguiente formato:

`html:nth-child(1) > head:nth-child(2) > title:nth-child(1)`

Este selector que se tomo como ejemplo hace referencia a la etiqueta "Title de la página web"

`<title>Universidad Politécnica Salesiana - UPS</title>`

Otra adecuación que se tuvo que realizar fue el módulo de respuestas, ya que algunos resultados del análisis no se los puede clasificar automáticamente como un error o acierto, para esto se requiere el criterio del usuario para clasificarlo correctamente.

La API no ofrecía esta funcionalidad por lo que se implementó para que el plugin sea más eficiente en su análisis.

Cabe aclarar que el usuario debe haber iniciado sesión en el plugin del navegador para poder acceder a esta funcionalidad, ya que su respuesta se almacena en una base de datos asociado a su id de usuario.

6.3. Desarrollar la extensión de evaluación de accesibilidad de páginas web que haga uso del API de la herramienta de Análisis Automático de Accesibilidad WEB de la UPS

En primera instancia, es necesario adquirir un api key, para ello se debe registrarse en el observatorio de accesibilidad web, disponible en el siguiente enlace:

<http://observatorioweb.ups.edu.ec/owa/login.jsf?>, en donde se debe rellenar un formulario corto con información personal básica como nombre de usuario, correo y contraseña.

Una vez registrado en la página del observatorio, esta la redirigirá al perfil correspondiente en donde estará disponible el Api Key para ser utilizado.

Como segundo paso se elaboró un prototipo que es un proceso clave para la construcción de software, el realizar un prototipo en la fase de diseño ayuda a evitar problemas posteriores de tiempo, esfuerzo y dinero.

Para cumplir con el objetivo planteado en este punto se utilizó la herramienta Gravity Designer, que es un software especializado en la creación de prototipos y diseño de interfaces.

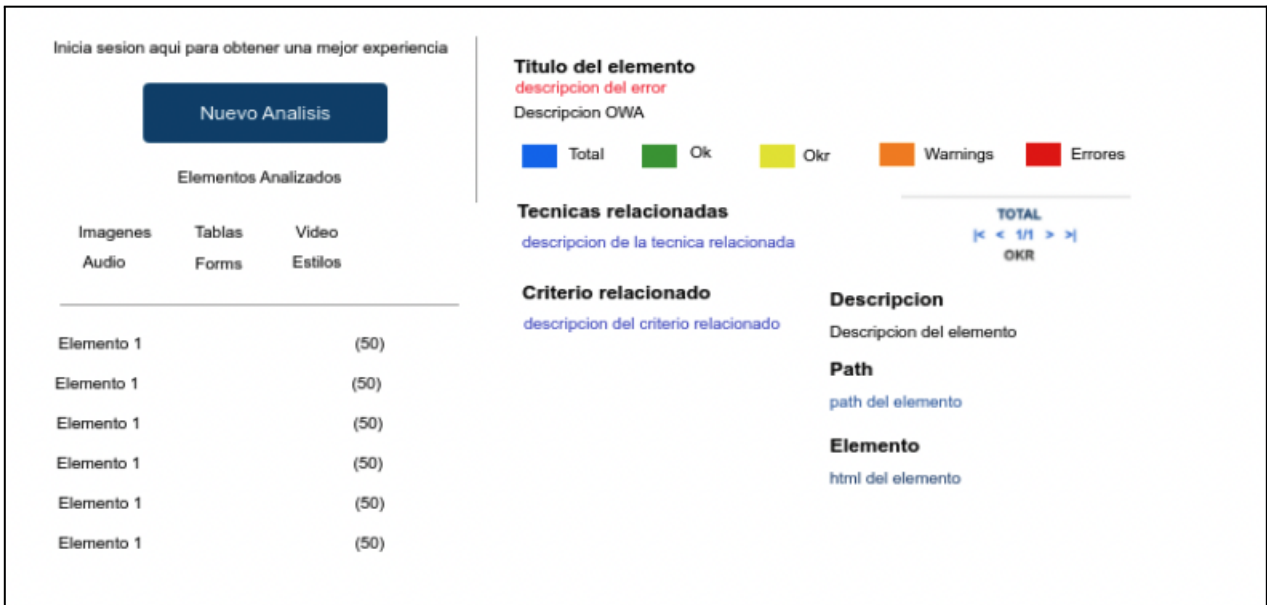


Figura 8 Prototipo de la pantalla principal de la extensión

Una vez terminada la fase de diseño se procedió a la fase de construcción del software en sí, ayudándose de herramientas indispensables como git, alojando en un servidor de versiones en este caso se utilizó bitbucket y source tree.

Otra de las librerías que se utilizó en el proyecto fue JQuery para manejar la interacción con el DOM y manipularlo en caso de ser necesario.

Posterior a esto se elaboró un plan de pruebas que permita realizar el testing del plugin y corregir sus respectivos errores que se presenten para que al concluir el proyecto se logre tener un producto de calidad y funcional.

El plan de pruebas que se elaborado se basó en el Estándar IEEE-829, que es uno de los documentos más utilizados al momento de crear un plan de pruebas de software, ya que ha sido diseñada especialmente para la documentación de las mismas y por medio de su utilización se puede abarcar de una forma más amplia y precisa las actividades de documentación de pruebas, aprovechando al máximo los datos que podemos obtener de las pruebas que se ejecutan [17].

El documento del plan de pruebas se encuentra como un anexo a este documento.

6.4. Análisis de resultados del plan de pruebas.

Tipo de prueba	Caso de prueba	Resultado	Criterio de finalización
Funcional	Descarga e instalación del plugin en el navegador web chrome.	Esperado	Aprobado
Funcional	Prueba de conexión entre el plugin con el servidor	Esperado	Aprobado
Funcional	Inicio de sesión desde el plugin hacia el analizador de accesibilidad Web	Esperado	Aprobado
Interfaz de usuario	Responsive Design General	Esperado	Aprobado
Interfaz de usuario	Responsive Design (Visualización de resultados)	Esperado	Aprobado
Funcional	Envío correcto de parámetros al solicitar un nuevo análisis	Esperado	Aprobado
Funcional	Manejo de errores de la página	Esperado	Aprobado
Funcional	Verificación de enlaces o hipervínculos a la página WCAG2	Esperado	Aprobado
Funcional	Verificación de los datos que se muestra en el	Esperado	Aprobado

	resumen de la búsqueda realizada sean correctos.		
Funcional	Clasificación de resultados (errores, warnings, correctos y total)	Esperado	Aprobado
Funcional	Comprobar que se almacene las preguntas de verificación de forma correcta.	Esperado	Aprobado
Funcional	Verificar que si alguna pregunta que ya fue contestada anteriormente no aparezca nuevamente como pendiente de indicar su respuesta, esto cuando se realice un nuevo análisis a una misma página web.	Esperado	Aprobado
Funcional	Cuando se seleccione "ubicar elemento" comprobar que este resalte en la página web correctamente	Esperado	Aprobado
Funcional	Cuando se seleccione "ubicar elemento" comprobar que la	Esperado	Aprobado

	página se desplaza automáticamente hacia donde esté el elemento resaltado		
--	---	--	--

Tabla 2. Análisis de resultados del plan de pruebas

7. CRONOGRAMA

Objetivo Específico 1		Estudiar y desarrollar un objeto de aprendizaje que documente el proceso de creación y publicación de una extensión (plugin) para Google Chrome.
Actividad	1	Investigación del proceso y requerimientos previos para la creación de extensiones para el navegador google Chrome.
Actividad	2	Investigación de herramientas de programación para la creación de plugins de google Chrome.
Actividad	3	Construir un objeto de aprendizaje que de manera práctica y sencilla explique como crear una extensión para Google Chrome.
Actividad	4	Documentar los resultados de la investigación realizada.

Objetivo Específico 2		Examinar la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana y realizar las adecuaciones que sean necesarias para que el API se adapte a los requerimientos del proyecto.
Actividad	1	Revisión de Analizador Web desarrollado en una primera fase por el Observatorio de Accesibilidad WEB.
Actividad	2	Análisis de las normativas implementadas para el análisis
Actividad	3	Realizar un análisis e implementar las adecuaciones necesarias en la herramienta y el API

Objetivo Específico 3		Desarrollar la extensión de evaluación de accesibilidad de páginas web que haga uso del API de la herramienta de Análisis Automático de Accesibilidad WEB de la UPS
Actividad	1	Obtener la Api Key de la Api Rest del observatorio de accesibilidad Web.
Actividad	2	Elaborar un prototipo básico de la interfaz del plugin en donde se visualizarán los resultados.
Actividad	4	Implementación del plugin para el navegador utilizando las herramientas definidas en el OE1 y el diseño en la actividad previa.
Actividad	5	Elaboración de un plan de pruebas para el plugin que considere una muestra bajo criterios estadísticos.
Actividad	6	Ejecución del plan de testing del plugin en el navegador Chrome con diferentes versiones.
Actividad	7	Análisis de los resultados obtenidos de la ejecución del plan de testing

Objetivo Específico 4		Publicar el plugin desarrollado para su descarga y uso desde un navegador web Chrome.
Actividad	1	Obtener una cuenta de desarrollador de Google.
Actividad	2	Publicar la extensión en Chrome Web Store y verificar su disponibilidad para el usuario final
Actividad	3	Documentar el proceso de publicacion y actualizacion del plugin dentro de Chrome Web Store

Actividades Adicionales		
Actividad	1	Preparación de documentos e informes de las actividades realizadas.

Tabla 3. Cronograma de actividades

Nombre de la tarea	Fecha de Inicio	Fecha final	Duración
Estudiar y desarrollar un objeto de aprendizaje que documente el proceso de creación y publicación de una extensión (plugin) para Google Chrome.	04/16/18		
Investigación del proceso y requerimientos previos para la creación de extensiones para el navegador google Chrome.	04/16/18	04/18/18	18h
Investigación de herramientas de programación para la creación de plugins de google Chrome.	04/19/18	04/23/18	18h
Construir un objeto de aprendizaje que de manera práctica y sencilla explique como crear una extensión para Google Chrome.	04/24/18	05/01/18	32h
Documentar los resultados de la investigación realizada.	05/01/18	05/04/18	18h
Examinar la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana y realizar las adecuaciones que sean necesarias para que el API se adapte a los requerimientos del proyecto.	05/04/18	05/18/18	10d
Revisión de Analizador Web desarrollado en una primera fase por el Observatorio de Accesibilidad WEB.	05/04/18	05/09/18	18h
Análisis de las Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0	05/09/18	05/14/18	18h
Realizar un análisis e implementar las adecuaciones necesarias en la herramienta y el API	05/14/18	05/18/18	24h
Desarrollar la extensión de evaluación de accesibilidad de páginas web que haga uso del API de la herramienta de Análisis Automático de Accesibilidad WEB de la UPS, considerando las adecuaciones que sean necesarios en la herramienta y el API.	05/18/18	07/26/18	49d
Obtener la Api Key de la Api Rest del observatorio de accesibilidad Web.	05/18/18	05/21/18	6h
Elaborar un prototipo básico de la interfaz del plugin en donde se visualizarán los resultados.	05/21/18	05/24/18	18h
Implementación del plugin para el navegador utilizando las herramientas definidas en el OE1 y el diseño en la actividad previa.	05/24/18	06/27/18	144h
Elaboración de un plan de pruebas para el plugin que considere una muestra bajo criterios estadísticos.	06/27/18	06/29/18	12h
Ejecución del plan de testing del plugin en el navegador Chrome con diferentes versiones.	06/29/18	07/04/18	18h
Análisis de los resultados obtenidos de la ejecución del plan de testing	07/04/18	07/09/18	18h
Publicar el plugin desarrollado para su descarga y uso desde un navegador web Chrome.	07/09/18	07/19/18	8d
Obtener una cuenta de desarrollador de Google.	07/09/18	07/10/18	6h
Publicar la extensión en Chrome Web Store y verificar su disponibilidad para el usuario final	07/10/18	07/13/18	18h
Documentar el proceso de publicación y actualización del plugin dentro de Chrome Web Store	07/13/18	07/19/18	24h
Preparación de documentos e informes de las actividades realizadas.	07/19/18	07/26/18	30h

Figura 9 Lista de actividades realizadas.

8. PRESUPUESTO

Elemento	Valor	Cantidad	Total
Desarrollador de software	8 USD	440	3520 USD
Laptop Mac book pro 2015	1800 USD	1	1800 USD
Internet 10 GB	22 USD	3	66 USD
Cuenta de google developer	5 USD	1	5 USD
		Total	5391 USD

Tabla 4. presupuesto real del proyecto

9. CONCLUSIONES

Como conclusión se puede decir que el trabajo que se realizó logrando desarrollar el plugin tal como se planteó en el objetivo principal de este proyecto, algo importante a destacar es que los desarrolladores de software tendrán a su alcance una herramienta más para facilitar la construcción de sitios web accesibles, ya que con este plugin podrán identificar fácilmente qué tan accesible es el mismo, así como también los errores, advertencias o sugerencias según sea el caso.

Al tener una herramienta que facilite el trabajo a los desarrolladores en el campo de la accesibilidad web, el plugin puede tener otro beneficio, que sería ayudar a promover la construcción de sitios con estas características y a vez más personas que así la requieran puedan ser las beneficiadas.

10. RECOMENDACIONES

Es recomendable que se considere de alguna manera la posibilidad de que el plugin desarrollado sea compatible con otros navegadores, como por ejemplo, Firefox, Opera o Safari, con esto se puede llegar a más usuarios que utilicen esta herramienta.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. [consejodiscapacidades.gob.ec](http://www.consejodiscapacidades.gob.ec). (2017). Biblioteca | Accesibilidad Universal y Diseño para Todos (Normas Técnicas Ecuatorianas). [online] Available at: <http://www.consejodiscapacidades.gob.ec/biblioteca/>
- [2]. INEN. (2014). TECNOLOGÍA DE LA INFORMACIÓN - DIRECTRICES DE ACCESIBILIDAD PARA EL CONTENIDO WEB DEL W3C (WCAG)2.0 (ISO/IEC 40500:2012,IDT). 03/2017, de MINISTERIO DE DEFENSA NACIONAL Sitio web: http://www.defensa.gob.ec/wp-content/uploads/downloads/2015/12/inen_iso_ice_40500-unidov-1.pdf
- [3]. Sergio Luján Mora. (2006 - 2018). Accesibilidad Web - Pautas de accesibilidad al contenido web . 04/2018, de UNIVERSIDAD DE ALICANTE Sitio web: <http://accesibilidadweb.dlsi.ua.es/?menu=pautas-2.0>

- [4]. Rodríguez Rodríguez, R. E., & Escobar Ravelo, D. C. (2014). *Integración de Plugin a un navegador web para análisis de vulnerabilidades en JavaScript* (Bachelor's thesis).
- [5]. López Zambrano, J. H., Pico, M., Joffre, R., & Alava Cagua, N. V. (2018). Metodología para valorar y clasificar herramientas de evaluación de accesibilidad web. *e-Ciencias de la Información*, 8(1), 1-18.
- [6]. González Orellana, R. A. (2011). Implementación de una API de Información Bancaria para la SBIF.
- [7]. Ruiz, J. V. B. (2011). *Chrome webfiltering toolbar: migración de un addon de Firefox a una extensión de Google Chrome*(Doctoral dissertation).
- [8]. Mascaraque, E. S., Ocaña, A. M., & Martos, I. O. (2010). Métrica para la evaluación de la accesibilidad en Internet: propuesta y testeo. *Revista española de documentación científica*, 33(3), 378-396.
- [9]. Rodríguez, C., & Dorado, R. (2015). ¿ Por qué implementar Scrum?. *Revista Ontare*, 3(1), 125-144.
- [10]. López Zambrano, J. H., Pico, M., Joffre, R., & Alava Cagua, N. V. (2018). Metodología para valorar y clasificar herramientas de evaluación de accesibilidad web. *e-Ciencias de la Información*, 8(1), 1-18.
- [11]. Rodríguez Rodríguez, R. E., & Escobar Ravelo, D. C. (2014). *Integración de Plugin a un navegador web para análisis de vulnerabilidades en JavaScript* (Bachelor's thesis).
- [12]. Lugo, M., & Antonio, J. (2017). *Aplicación web y móvil con tecnologías Rest y geoposicionamiento para el control de recepción y entrega de prendas lavadas de la empresa Lavanderías Monserrat* (Bachelor's thesis).
- [13]. Velázquez, C., Muñoz, J., Álvarez, F., & Arévalo, C. (2005). La Importancia de la Definición de la Calidad del Contenido de un Objeto de Aprendizaje. *Avances en la ciencia de la computación*, 329-33.
- [14]. Gallego, M. T. (2012). Metodologia scrum. *Universitat Oberta de Catalunya*.
- [15]. Aguirre Benavides, C. D. (2019). *Estudio de la integración de Yii2 y REACT. JS para la elaboración de un sistema de tipificación para la cruz roja ecuatoriana junta provincial de Imbabura* (Bachelor's thesis).

- [16]. Exelearning.net. (2019). *Características de eXeLearning* | *eXeLearning.net*. [online] Available at: <https://exelearning.net/caracteristicas/> [Accessed 20 Oct. 2019].
- [17]. Martínez, N. E. L., García, J. L. A., Morales, E. F. V., & Florez, L. C. G. (2013). Herramienta computacional para la documentación de pruebas de software enmarcado en actividades de investigación. *Scientia et Technica*, 18(4), 682-689.
- [18]. Campoverde Molina, M. (2019). La accesibilidad web. Un reto en el entorno educativo ecuatoriano. *Revista Científica y Tecnológica UPSE*, 2016, vol. 3, num. 3, p. 90-98.
- [19]. Mascaraque, E. S. (2009). Herramientas para la evaluación de la accesibilidad Web/Tools for the evaluation of Web accessibility. *Documentación de las Ciencias de la Información*, 32, 245.
- [20]. BENEYTO RUIZ, J. V. (2011). Chrome webfiltering toolbar: migración de un addon de Firefox a una extensión de Google Chrome (Doctoral dissertation).
- [21]. Chanchí, G. E., Campo, W. Y., Amaya, J. P., & Arciniegas, J. L. (2011). Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON. *Cadernos de Informática*, 6(1), 233-240.
- [22]. Brooks, T. A. (2010). World wide web consortium (W3C). *Encyclopedia of library and information sciences*, 5695-5699.
- [23]. Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)*, 290, 1-34.
- [24]. Sarmiento Basurto, D. B., & Zhizhpon Tacuri, C. E. (2022). *Diseño, desarrollo e implementación de una herramienta web de análisis estadístico y monitoreo para evaluar la accesibilidad en páginas web según los criterios de conformidad de la WCAG 2.1* (Bachelor's thesis).
- [25]. Marrs, T. (2017). *JSON at work: practical data integration for the web*. "O'Reilly Media, Inc."
- [26]. Ramírez, M. R., Soto, M. D. C. S., Moreno, H. B. R., Rojas, E. M., Millán, N. D. C. O., & Cisneros, R. F. R. (2019). Metodología SCRUM y desarrollo de Repositorio Digital. *Revista Ibérica De Sistemas e Tecnologías De Informação*, (E17), 1062-1072.

- [27]. Liu, L., Zhang, X., Yan, G., & Chen, S. (2012, February). Chrome Extensions: Threat Analysis and Countermeasures. In *NDSS*.
- [28]. Cabrera, M. P., & López, F. J. (2014). Estudio de la Normativa WCAG 2.0 y análisis de accesibilidad web para la creación de un prototipo de sistema experto basado en casos, que permita generar reportes de accesibilidad de páginas web, tomando como base un sitio web ecuatoriano. *Universidad Politécnica Salesiana, Cuenca. Recuperada de <http://www.dspace.ups.edu.ec/bitstream/123456789/6290/1/UPSCT002849.pdf>*.
- [29]. Alvarez, M. A. (2010). Manual de jQuery. *Recuperado el, 17*.
- [30]. Viñanzaca Toledo, R. F. (2014). *Propuesta para la implementación de buenas prácticas de usabilidad y accesibilidad web para el portal y AVAC de la Universidad Politécnica Salesiana* (Bachelor's thesis)

ANEXOS

Manual de uso del Plugin de Accesibilidad Web

Índice.

Manual de uso del Plugin de Accesibilidad Web	1
Introducción	2
Objetivos	2
Requerimientos	2
Uso del plugin	2
Instalación del plugin	2
Uso del plugin en la página Web.	3
Inicio de sesión	6

1. Introducción

a. Objetivos

El objetivo de este documento es otorgar a los usuarios una guía de como utilizar el plugin de accesibilidad web en el navegador Google Chrome.

b. Requerimientos

- i. Tener el Navegador Google Chrome instalado en la versión 30 o superiores.
- ii. Acceso a internet.

2. Uso del plugin

a. Instalación del plugin

1.- Ingresar a la tienda de Chrome y en el buscador poner **OAW**.

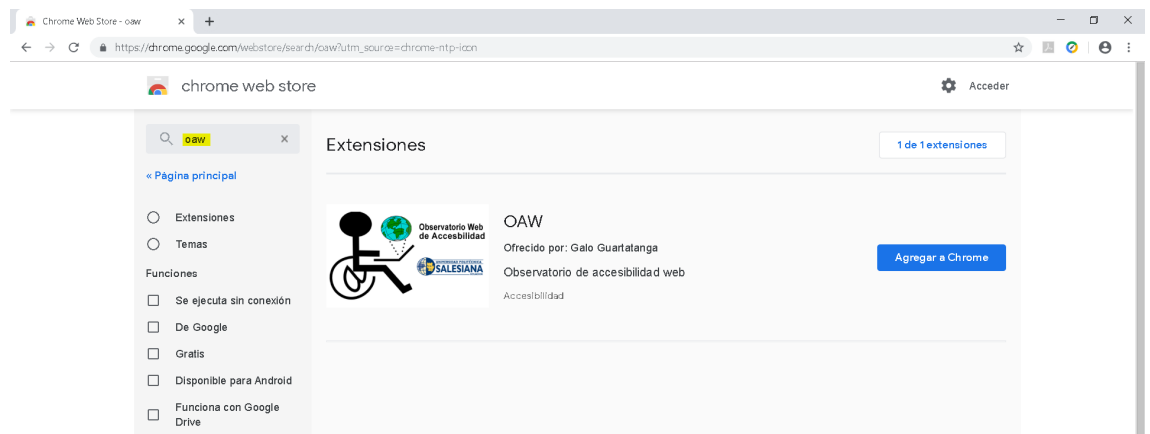


Imagen 1. Extensión OAW.

2.- Proceder a la Instalación, que en este caso será la opción de **Agregar a Chrome**.

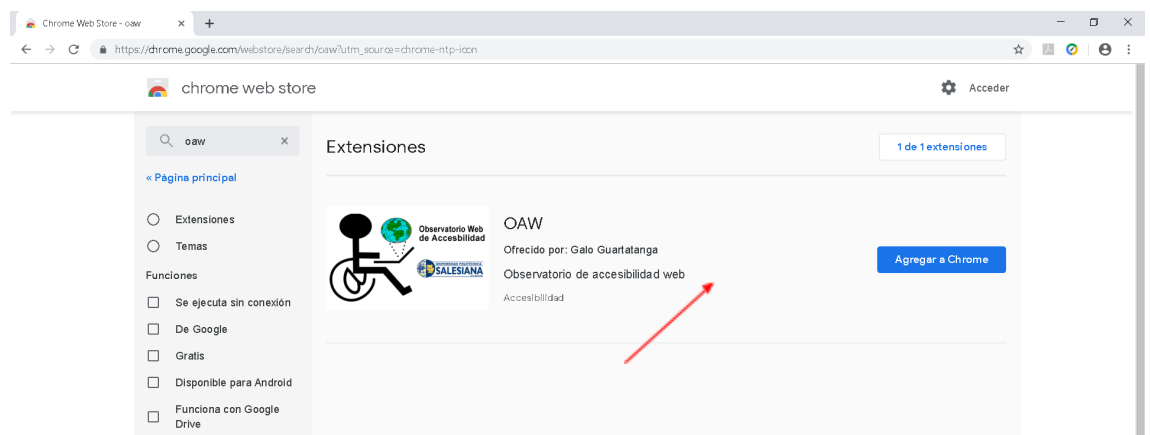


Imagen 2. Extensión OAW

3.- Validar la extensión y esperamos a que se realice la instalación.

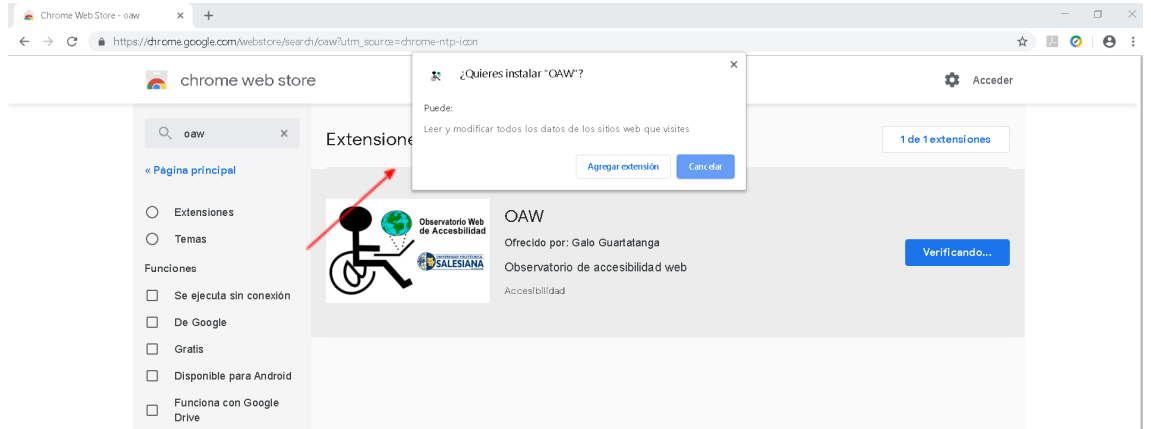


Imagen 3. Aprobación Plugin AOW

4.- Finalizada la instalación nos aparecerá el icono de la extensión que fue instalada en el navegador, como se muestra a continuación.

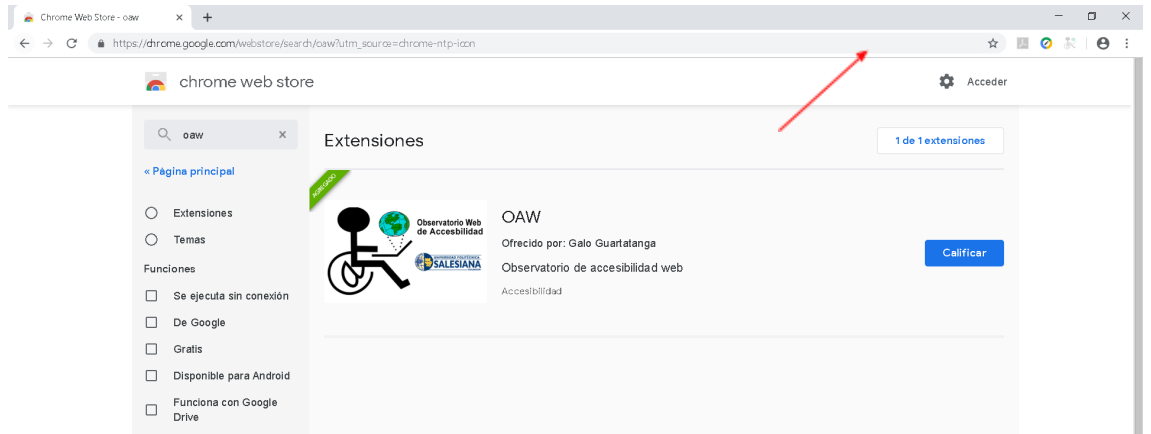


Imagen 4. Plugin Instalado.

b. Uso del plugin en la página Web.

1.- El uso del plugin tiene el propósito de analizar las páginas web e informar las necesidades de accesibilidad del sitio web. Lo cual para comenzar con el análisis damos clic derecho en cualquier parte del sitio web y en el menú que se visualizará elegir la opción de inspeccionar.



Imagen 5. Menú Opciones Navegador Chrome

2.- Después elegir la opción del plugin **OAW** y nos mostrará el contenido para el análisis necesario, por lo mediante se pondrá **Analizar** para que comience el funcionamiento del plugin.

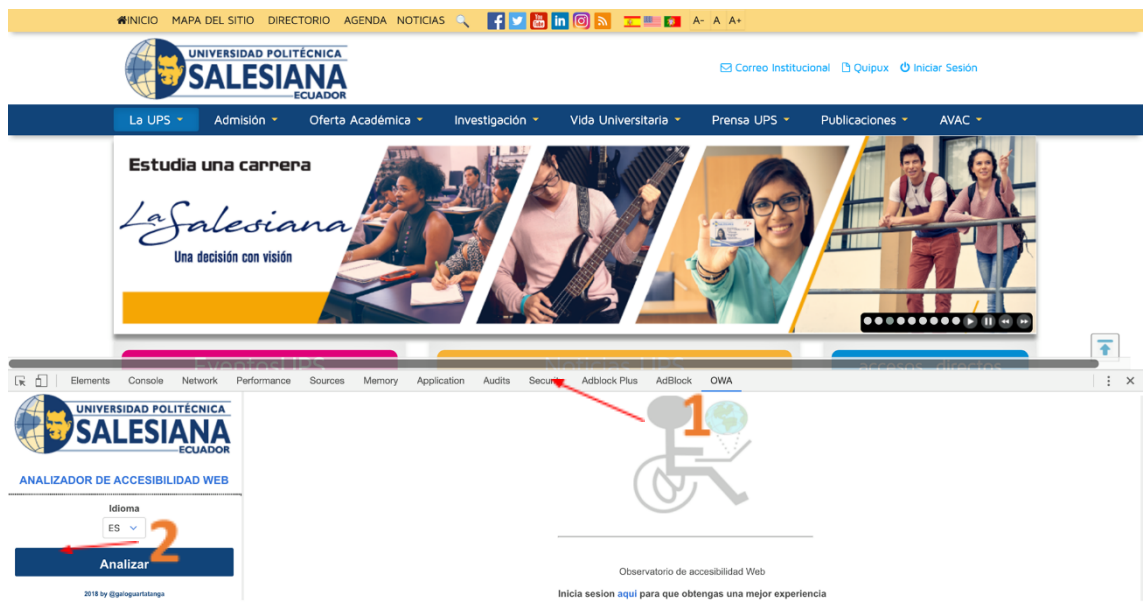


Imagen 6. Pantalla Principal de OAW.

3- Cuando comienza el análisis de la página web, cambia el logo del plugin y muestra los resultados obtenidos, si es necesario o conveniente se procede a realizar un **Nuevo Análisis**.

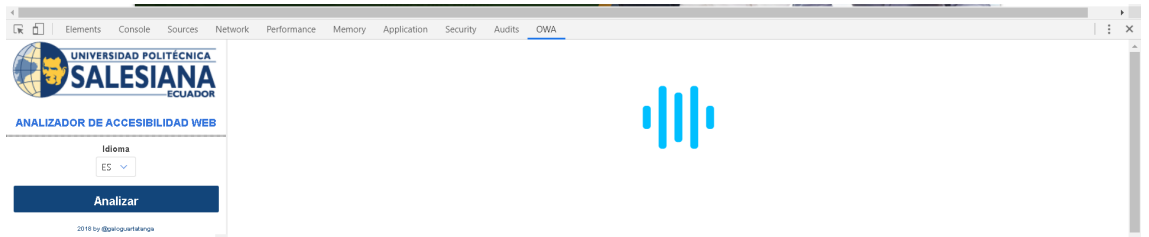


Imagen 7. Pantalla de Análisis

E

Imagen 8. Resultado Análisis

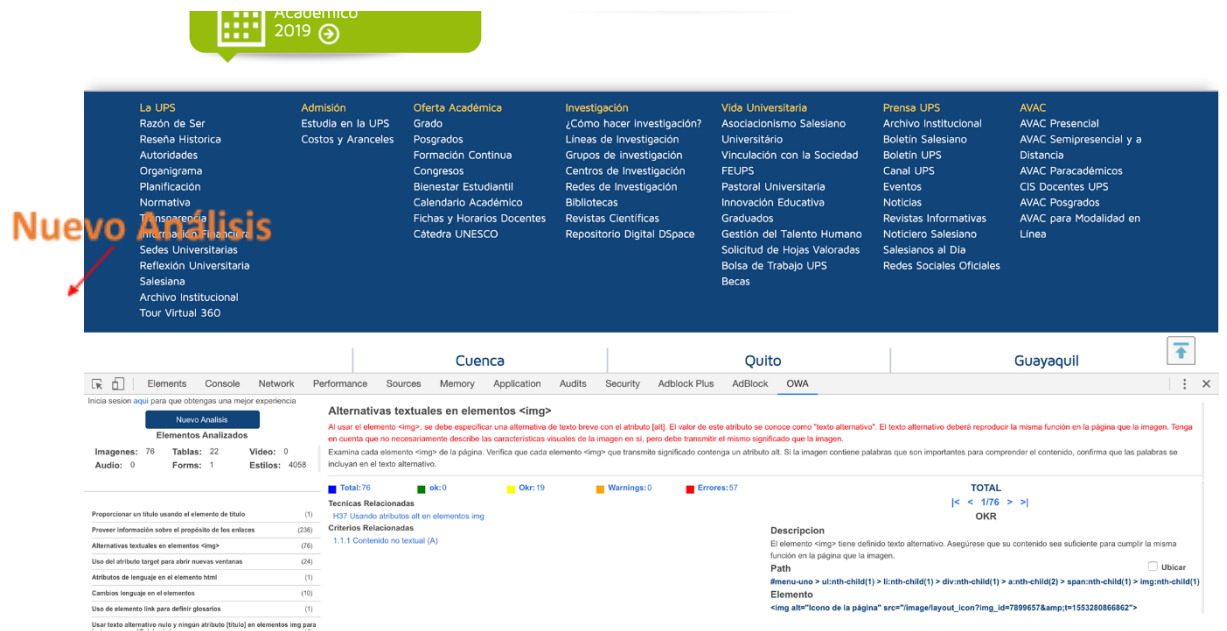


Imagen 9. Resultado de Análisis

c. Inicio de sesión

1.- Para una mayor experiencia, es necesario registrarse en el plugin en la opción que se muestra en la imagen.

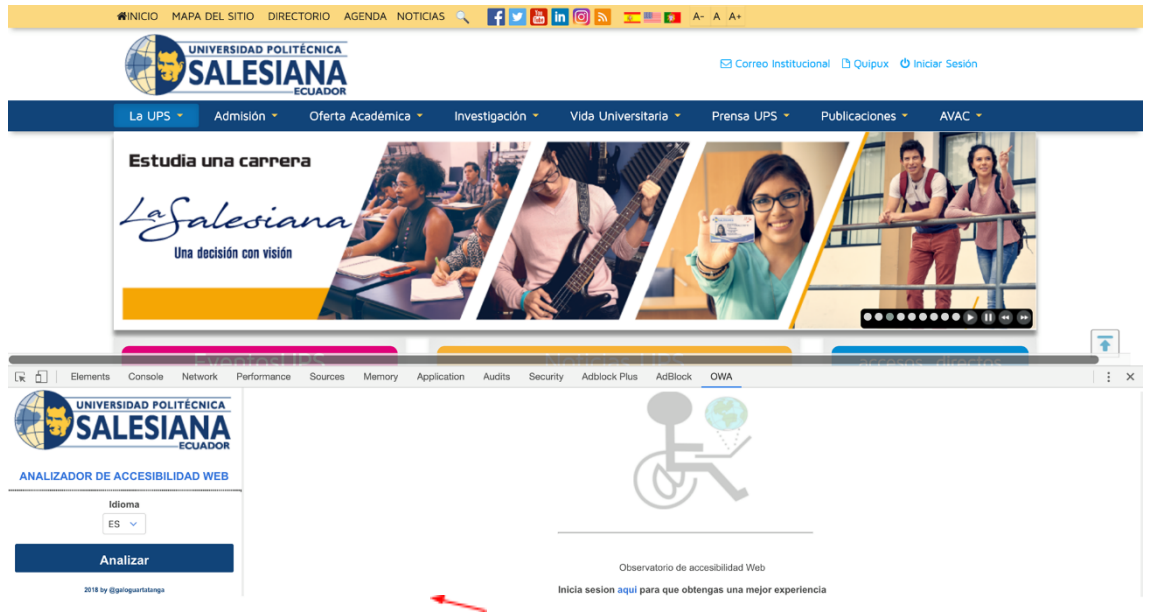


Imagen 10. Pantalla Opción de Registro

2.- En el caso de ya encontrarse registrado solo es necesario ingresar el usuario y contraseña, caso contrario llenar el formulario de Registro y poner crear cuenta

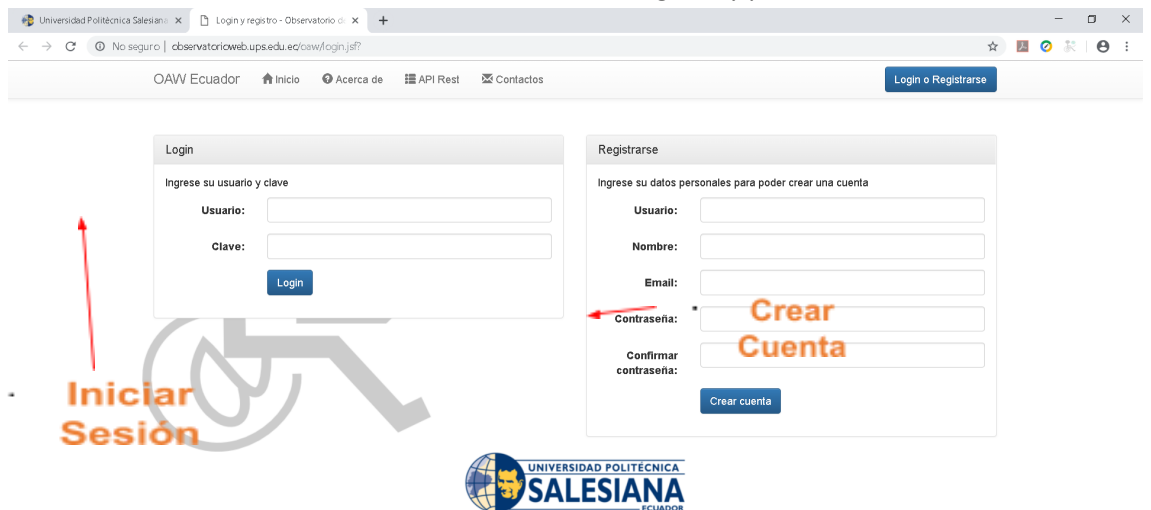


Imagen 11. Pantalla Inicio Sesión y Registro.

3.- Una vez registrado e iniciado sesión se mostrará los datos del perfil.

The screenshot shows a web browser window with two tabs: 'Universidad Politécnica Salesiana' and 'Mi Cuenta - Observatorio de Ac...'. The address bar shows the URL 'observatorioweb.ups.edu.ec/oaw/login.jsf'. The navigation menu includes 'OAW Ecuador', 'Inicio', 'Acerca de', 'API Rest', 'Contactos', and 'Mi Cuenta'. The user is logged in as 'edwinqs' with a 'Salir' button.

Mi Cuenta

- Información General
- Cambiar contraseña
- Estadísticas

Información General

API_KEY: 943858bc-61ca-4ae5-99f4-3bc11c5cacaee

Usuario: edwinqs
Nombre: Edwin Ouishpe
Email: edwinqs5@gmail.com
Registrado desde: jueves 2 de mayo de 2019 17:09:19 GMT




Imagen 12. Perfil Usuario

Desarrollo e implementación de una extensión (plugin) para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana.

PLAN DE PRUEBAS

**Universidad Politécnica Salesiana
Ingeniería de Sistemas
24 de febrero del 2019**

CONTENIDO

	Pág.
1. <u>INTRODUCCIÓN</u>	4
2. <u>OBJETIVO</u>	4
3. <u>ALCANCE DE LAS PRUEBAS FUNCIONALES</u>	4
4. <u>CRITERIOS DE ENTRADA Y SALIDA</u>	5
4.1. <u>Criterio de Ejecución del Plan de Pruebas</u>	5
4.2. <u>Criterio de Terminación del Plan de Pruebas</u>	6
4.3. <u>Criterio de Suspensión del Plan de Pruebas</u>	6
5. <u>ESCENARIOS PROPUESTOS DE PRUEBA</u>	7
5.1. <u>DESCARGA E INSTALACIÓN DEL PLUGIN EN EL NAVEGADOR WEB CHROME.</u>	7
5.2. <u>PRUEBA DE CONEXIÓN ENTRE EL PLUGIN CON EL SERVIDOR</u>	8

1. INTRODUCCIÓN

Por medio de este documento se pretende establecer la guía para la elaboración del Plan de Prueba del proyecto **“Desarrollo e implementación de una extensión (plugin) para el navegador web Google Chrome para análisis y evaluación de accesibilidad de una página web como complemento de la herramienta de Análisis Automático de Accesibilidad WEB de la Universidad Politécnica Salesiana”**.

2. OBJETIVO

El propósito del plan de pruebas es proveer la información necesaria para planear y controlar los esfuerzos de pruebas de un proyecto o iteración específicos. Describe el enfoque para probar el software.

3. CRITERIOS DE ENTRADA Y SALIDA

3.1. Criterio de Ejecución del Plan de Pruebas

- Set de pruebas documentado incluyendo escenarios claros para el desarrollo de las pruebas unitarias.
- Claridad en el procedimiento para la realización de las pruebas.
- El entorno de pruebas debe ser el adecuado para la realización de las pruebas.

3.2. Criterio de Terminación del Plan de Pruebas

- Todas las pruebas se ejecutan sin errores.

3.3. Criterio de Suspensión del Plan de Pruebas

- Una componente principal tiene un error que impide probar un área importante.
- El entorno de pruebas no es lo suficientemente estable como para confiar en los resultados.
- El entorno de pruebas es muy diferente del entorno de producción previsto y no se puede confiar en los resultados.

4. ESCENARIOS PROPUESTOS DE PRUEBA

4.1. DESCARGA E INSTALACIÓN DEL PLUGIN EN EL NAVEGADOR WEB CHROME.

Información General	
Identificador de caso de uso:	p-001
Nombre de caso de uso:	Descarga e instalación del plugin en el navegador web chrome.
Descripción Prueba:	Buscar el plugin en chrome store y descargar la aplicación (beta) e instalar en un navegador web chrome en su última versión disponible a la fecha.
Responsable:	Galo Guartatanga
Prerrequisitos	
<p>Se debe tener instalada la última versión de Google Chrome en la computadora.</p> <p>El usuario debe dirigirse a la tienda oficial de Google Chrome y buscar la extensión con el nombre OAW y posterior a eso instalar la extensión.</p>	
Descripción de Casos de Prueba	
Caso: Descarga e instalación del plugin en el navegador web chrome.	
Instrucciones de Prueba	
<ol style="list-style-type: none"> 1. Realizar la búsqueda del plugin con una cuenta registrada de Google, descargar e instalar, en caso de que exista ninguna coincidencia en el navegador debe verificarse que la cuenta de google asociada este registrada como tester del plugin. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> 1. En caso de que se encuentre la aplicación se debe proceder a la instalación en el navegador, posterior a eso automaticamente se agregara en la consola de desarrolladores del navegador una pestaña con el nombre del plugin. 	

4.2. PRUEBA DE CONEXIÓN ENTRE EL PLUGIN CON EL SERVIDOR.

Información General	
Identificador de caso de uso:	p-002
Nombre de caso de uso:	Prueba de conexión entre el plugin con el servidor.
Descripción Prueba:	Realizar la prueba de conexión entre el plugin y el servidor a través del botón "Analizar" y verificar que devuelva resultados.
Responsable:	Galo Guartatanga
Prerrequisitos	
<ol style="list-style-type: none">1. Plugin instalado en el navegador Google Chrome.2. Conexión a internet.	
Descripción de Casos de Prueba	
Caso: Prueba de conexión entre el plugin con el servidor.	
Instrucciones de Prueba <ol style="list-style-type: none">2. En una pestaña del navegador abrir la herramienta de desarrolladores, seleccionar la pestaña OWA y pulsar el botón analizar.	
Criterios de Aceptación <ol style="list-style-type: none">2. Debe devolver un resultado del análisis de la página web, en caso de devolver un mensaje de error no se considerara este test como válido.	

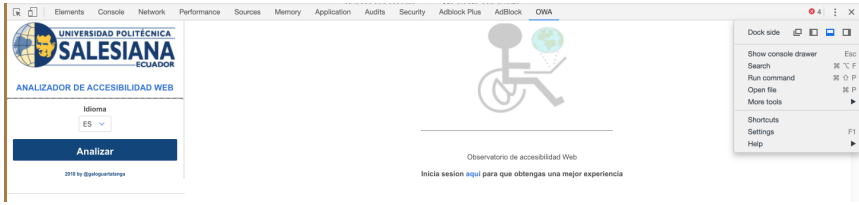
4.3. INICIO DE SESIÓN DESDE EL PLUGIN HACIA EL ANALIZADOR DE ACCESIBILIDAD WEB.

Información General

Identificador de caso de uso:	p-003
Nombre de caso de uso:	Inicio de sesión desde el plugin hacia el analizador de accesibilidad Web.
Descripción Prueba:	Probar el Inicio de sesión desde el plugin en el navegador.
Responsable:	Galo Guartatanga
Prerrequisitos	
<p>Crear una cuenta en el la página del observatorio de accesibilidad web. http://observatorioweb.ups.edu.ec/owa/index.jsf</p>	
Descripción de Casos de Prueba	
Caso: Inicio de sesión desde el plugin hacia el observatorio de accesibilidad Web	
Instrucciones de Prueba	
<ol style="list-style-type: none"> En una pestaña del navegador abrir la herramienta de desarrolladores, seleccionar la pestaña OWA, ingresar el usuario y la contraseña y finalmente pulsar en Login, este test se debe realizar tanto con las credenciales correctas como con credenciales incorrectas. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> Si el inicio de sesión es correcto se dará por concluida la prueba, en caso de ingresar las credenciales incorrectas, deberá mostrar un mensaje indicando que se ingresó datos incorrectos, en caso de producirse un error distinto no se considerará este test como válido. 	

4.4. RESPONSIVE DESIGN GENERAL

Información General	
Identificador de caso de uso:	p-004
Nombre de caso de uso:	Responsive Design General

Descripción Prueba:	Realizar el test en diferentes versiones de los navegadores y resolución de pantalla y comprobar que el contenido se muestre de manera amigable.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	
Caso: Responsive Design General	
Instrucciones de Prueba	
<p>4. Abrir el Chrome developers tools y en la pestaña OAW verificar que se muestre correcta y amigablemente, se debe interactuar cambiando la posición del dockside en distintas posiciones (abajo , arriba, izquierda o derecha), como se muestra en la siguiente imagen:</p>	
	
Criterios de Aceptación	
<p>4. Si la visualización es correcta y amigable de todos los componentes de interfaz gráfica de la extensión se da el test como válido.</p>	

4.5. RESPONSIVE DESIGN (VISUALIZACIÓN DE RESULTADOS)

Información General	
Identificador de caso de uso:	p-005
Nombre de caso de uso:	Responsive Design (Visualización de resultados)
Descripción Prueba:	Esta prueba consiste básicamente en verificar que los resultados que devuelve al analizar la página se muestren correctamente.

Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	
Caso: Responsive Design (Visualización de resultados)	
Instrucciones de Prueba	
<ol style="list-style-type: none"> 5. Realizar un análisis de una página cualquiera y verificar que el resultado se muestra correcta y amigablemente en distintas versiones del navegador Google Chrome de la misma forma que se realizó el test anterior. 	
Criterios de Aceptación	
<ol style="list-style-type: none"> 5. Si la visualización es correcta y amigable de todos los componentes de interfaz gráfica de la extensión se da el test como válido. 	

4.6. ENVÍO CORRECTO DE PARÁMETROS AL SOLICITAR UN NUEVO ANÁLISIS

Información General	
Identificador de caso de uso:	p-006
Nombre de caso de uso:	Envío correcto de parámetros al solicitar un nuevo análisis
Descripción Prueba:	Verificar que al pulsar analizar se envíe los parámetros correctos para el análisis de la página.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	

Caso: Actualización del plugin en la tienda de Google Chrome

Instrucciones de Prueba

6. Realizar una petición para el análisis de una página cualquiera, se debe verificar que se envíe correctamente parámetros como:
 - a. Idioma
 - b. Resolución de pantalla
 - c. Url de la página que se requiere analizar.

Criterios de Aceptación

6. Si todos los parámetros enviados concuerdan con lo requerido se considera esta prueba como válida.

4.7. MANEJO DE ERRORES DE LA PÁGINA

Información General	
Identificador de caso de uso:	p-007
Nombre de caso de uso:	Manejo de errores de la página
Descripción Prueba:	Se pretende verificar que cuando se produzca un error en la página está respuesta con un mensaje claro hacia el usuario indicando porque se produjo el error.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	
Caso: Actualización del plugin en la tienda de Google Chrome	
Instrucciones de Prueba	
<ol style="list-style-type: none">7. Realizar una petición para el análisis de una página cualquiera, y tratar de forzar a que se produzca un error realizando las siguientes acciones<ol style="list-style-type: none">a. Inhabilitado temporalmente el acceso a los servidores donde está alojado el API del sistema OAW	

- b. Realizar una petición sin conexión a internet.
- c. Realizar una petición con internet y en el proceso desconectarse del mismo.
- d. Realizar una petición con un pestaña en blanco (Sin ninguna URL en la barra de direcciones)

Criterios de Aceptación

7. Si se muestran los mensajes adecuados y correctos en cada uno de estos casos se consideraría esta prueba como válida.

Para el caso del literal 7 - a : Se debería mostrar un mensaje que indique lo siguiente:
“Se produjo un error de conexión”

Para el caso del literal 7 - b : Se debería mostrar un mensaje que indique lo siguiente:
“Compruebe su conexión a internet”

Para el caso del literal 7 - c : Se debería mostrar un mensaje que indique lo siguiente:
“Se perdió la conexión, compruebe su conexión e inténtelo nuevamente”

Para el caso del literal 7 - d : Se debería mostrar un mensaje que indique lo siguiente:
“No existe ninguna url en el navegador”

4.8. COMPROBAR QUE LAS TÉCNICAS RELACIONADAS PROPORCIONEN EL ACCESO A LA PÁGINA WCAG2.

Información General	
Identificador de caso de uso:	p-008
Nombre de caso de uso:	Verificación de enlaces o hipervínculos a la página WCAG2
Descripción Prueba:	Se pretende verificar que cuando se devuelva todas las técnicas relacionadas, en su descripción se muestre los accesos a los links proporcionados.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	
Caso: Actualización del plugin en la tienda de Google Chrome	

Instrucciones de Prueba

8. Realizar una petición para el análisis de una página cualquiera.
 - a. Verificar de forma aleatoria si en la sección de técnicas relacionadas se proporciona los links correctos hacia la página WCAG2

Criterios de Aceptación

8. Si en todos los casos se proporciona los links correctos se consideraría esta prueba como válida.

4.9. VERIFICACIÓN DE LOS DATOS QUE SE MUESTRA EN EL RESUMEN DE LA BÚSQUEDA REALIZADA SEAN CORRECTOS

Información General	
Identificador de caso de uso:	p-009
Nombre de caso de uso:	Verificación de los datos que se muestra en el resumen de la búsqueda realizada sean correctos.
Descripción Prueba:	Cuando se obtengan los resultados de una búsqueda comprobar que los datos que se muestran en el resumen sean correctos.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	
Caso: Verificación de los datos que se muestra en el resumen de la búsqueda realizada sean correctos.	
Instrucciones de Prueba	
9. Cuando se obtengan los resultados de una búsqueda comprobar que los datos que se muestran en el resumen sean correctos, esto se lo puede verificar de la siguiente manera:	

Ingresar a la pagina “<http://observatorioweb.ups.edu.ec/owa/index.js/>” y realizar el analisis respectivo de una pagina web, ejemplo: “<https://www.ups.edu.ec/>”, en dicho analisis retorna lo siguiente en la sección de Resumen:

Análisis WCAG 2.0 - Resumen

URI: <https://www.ups.edu.ec/>
 Título: Universidad Politécnica Salesiana - UPS
 Elementos: 1137
 Imágenes: 74
 Enlaces: 234
 Resolución de pantalla: 1366x768
 Nivel de conformidad WCAG: AAA
 Fecha/Hora: domingo 7 de abril de 2019 19:36:23 GMT

Se debe verificar que desde el plugin, ingresando desde la misma página web retorne los mismos valores que se muestran en la imagen anterior.

Crterios de Aceptación

9. Si los datos que se muestran coincide a plenitud con lo mostrado en la página web del observatorio de accesibilidad web. (excepto la fecha y hora)

4.10. CLASIFICACIÓN DE RESULTADOS (ERRORES, WARNINGS, CORRECTOS Y TOTAL)

Información General	
Identificador de caso de uso:	p-010
Nombre de caso de uso:	Clasificación de resultados (errores, warnings, correctos y total)
Descripción Prueba:	Cuando se obtengan los resultados de un análisis de una página web, comprobar que estos resultados esten clasificados correctamente a donde corresponde
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web.	
Descripción de Casos de Prueba	

Caso: Clasificación de resultados (errores, warnings, correctos y total)
Instrucciones de Prueba 10. Realizar un análisis de una página cualquiera y verificar en cada una de las clasificaciones (total, correctas, warnings, error) esten catalogadas a donde corresponde.
Criterios de Aceptación 10. Al verificar los resultados y comprobar que los resultados se muestran en donde corresponde se considera este prueba como válida.

4.11. COMPROBAR QUE SE ALMACENE LAS PREGUNTAS DE VERIFICACIÓN DE FORMA CORRECTA

Información General	
Identificador de caso de uso:	p-011
Nombre de caso de uso:	Comprobar que se almacene las preguntas de verificación de forma correcta.
Descripción Prueba:	Comprobar que se almacene las preguntas de verificación de forma correcta y no muestre ningún tipo de error
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web. Haber realizado el Login correspondiente en el plugin	
Descripción de Casos de Prueba	
Caso: Comprobar que se almacene las preguntas de verificación de forma correcta.	
Instrucciones de Prueba 11. Realizar un análisis de una página cualquiera y responder las preguntas de verificación	
Criterios de Aceptación 11. Al responder las preguntas de verificación asegurarse que estas se reorganicen clasificando como error o como correctas según se haya elegido en la respuesta.	

4.12. VERIFICAR QUE SI ALGUNA PREGUNTA QUE YA FUE CONTESTADA ANTERIORMENTE NO APAREZCA NUEVAMENTE COMO PENDIENTE DE INDICAR SU RESPUESTA, ESTO CUANDO SE REALICE UN NUEVO ANÁLISIS A UNA MISMA PÁGINA WEB.

Información General	
Identificador de caso de uso:	p-012
Nombre de caso de uso:	Verificar que si alguna pregunta que ya fue contestada anteriormente no aparezca nuevamente como pendiente de indicar su respuesta, esto cuando se realice un nuevo análisis a una misma página web.
Descripción Prueba:	Esta prueba permitirá verificar que si alguna pregunta que ya fue contestada anteriormente no aparezca nuevamente como pendiente de indicar su respuesta, esta prueba tiene que ser realizada haciendo un nuevo análisis a una misma página web
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web. Haber realizado el Login correspondiente en el plugin	
Descripción de Casos de Prueba	
Caso: Verificar que si alguna pregunta que ya fue contestada anteriormente no aparezca nuevamente como pendiente de indicar su respuesta, esto cuando se realice un nuevo análisis a una misma página web	
Instrucciones de Prueba	
12. Realizar un análisis de una página cualquiera y responder las preguntas de verificación, posteriormente realizar nuevamente un análisis y verificar que se cumpla con el objetivo propuesto en esta prueba.	
Criterios de Aceptación	
12. Si todas las preguntas que previamente fueron respondidas ya no aparecen nuevamente pendientes de responder, se considera este test como válido.	

4.13. CUANDO SE SELECCIONE "UBICAR ELEMENTO" COMPROBAR QUE ESTE RESALTE EN LA PÁGINA WEB CORRECTAMENTE

Información General

Identificador de caso de uso:	p-013
Nombre de caso de uso:	Cuando se seleccione "ubicar elemento" comprobar que este resalte en la página web correctamente
Descripción Prueba:	Esta prueba consiste en verificar que en la página se muestre el elemento previamente seleccionado de forma correcta y que resalte de manera que se pueda identificar correctamente.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web. Haber realizado el Login correspondiente en el plugin	
Descripción de Casos de Prueba	
Caso: Cuando se seleccione "ubicar elemento" comprobar que este resalte en la página web correctamente	
Instrucciones de Prueba	
13. Realizar un análisis de una página cualquiera y pulsar en el botón "Ubicar elemento" y verificar que se cumpla con el objetivo propuesto en este caso de prueba.	
Criterios de Aceptación	
13. Si al pulsar en el botón "Ubicar elemento" se muestra el elemento correcto y resaltado se considera este test como válido.	

4.14. CUANDO SE SELECCIONE "UBICAR ELEMENTO" COMPROBAR QUE LA PÁGINA SE DESPLAZE AUTOMÁTICAMENTE HACIA DONDE ESTÉ EL ELEMENTO RESALTADO

Información General	
Identificador de caso de uso:	p-014
Nombre de caso de uso:	Cuando se seleccione "ubicar elemento" comprobar que la página se desplace automáticamente hacia donde esté el elemento resaltado
Descripción Prueba:	Esta prueba consiste en verificar que la página se desplace de manera automática a donde está ubicado el elemento, de esta manera se podrá ubicar con mayor

	facilidad el elemento que se seleccionó para ser resaltado.
Responsable:	Galo Guartatanga
Prerrequisitos	
Tener instalado el plugin en el navegador web. Haber realizado el Login correspondiente en el plugin	
Descripción de Casos de Prueba	
Caso: Cuando se seleccione "ubicar elemento" comprobar que la página se desplace automáticamente hacia donde esté el elemento resaltado	
Instrucciones de Prueba	
14. Realizar un análisis de una página cualquiera y pulsar en el botón "Ubicar elemento" y verificar que se cumpla con el objetivo propuesto en este caso de prueba, cabe aclarar que este caso de prueba es necesario realizarlo desplazando la pagina al inicio o al final para que el desplazamiento automático sea más notorio.	
Criterios de Aceptación	
14. Si al pulsar en el botón "Ubicar elemento" la página se desplace hacia el elemento correcto y resaltado se considera este test como válido.	