Tesina de Licenciatura en sistemas

Detección automática de problemas de accesibilidad a partir de eventos de interacción de usuario

Autor: Maximiliano Jonathan Toledo

Mail: tole.maxi@hotmail.com

Directores: Dra. Alejandra Garrido
Dr. Julián Grigera

Resumen

Actualmente muchas de las actividades de nuestra vida cotidiana se encuentran integradas en una aplicación web. Toda la información de los hechos que acontecen en el mundo está a un simple clic, por eso es de suma importancia lograr que una gran parte de la sociedad tenga la posibilidad de acceder al contenido presente en la web. Aquí es donde la accesibilidad web se convierte en un recurso fundamental para combatir con la famosa brecha digital y permitir que el contenido web sea accesible a la mayor cantidad de personas posibles. Para brindar accesibilidad en un sitio web, es muy importante integrarla al proceso de desarrollo. Para facilitar esta integración y a la detección de problemas de accesibilidad, en esta tesina se desarrolló una herramienta automática para la detección y reporte de este tipo de problemas denominada **ABF** (*Accessibility BadSmells Finder*).

Código de la aplicación

ABF– Accessibility BadSmells Finder

Extensión Web – Link del repositorio público
https://github.com/tole22/tesinaLS/tree/master/Accessibility-web-extension
API REST – Link del repositorio público
https://github.com/tole22/tesinaLS/tree/master/rest-api-events-web-extension-Tesina
APP de reportes – Link del repositorio público
https://github.com/tole22/tesinaLS/tree/master/reportes-app-Tesina/app-reportes-tesina

Cómo ejecutar la herramienta

En esta sección se describen los pasos necesarios para poder instalar y ejecutar la herramienta de detección en un ambiente local. Para esto se definieron los pasos para poder ejecutar localmente todos los componentes de la herramienta.

Dividiremos esta sección por componente: Extensión Web, API REST - Base de Datos, y la Aplicación de Reportes.

Se necesita tener la extensión web, API REST y la Base de datos ejecutándose para lograr almacenar la información de los smells detectados por la extensión web.

Para lograr ver estos datos de una forma legible, se debe ejecutar la aplicación de reportes.

Tener pre-instalado:

- NODE JS: versión > 12.0.0, utilizada para este trabajo: v12.14.0
- Angular CLI: versión 10.2.1
- MONGODB
- Web Browser: Mozilla Firefox o Google Chrome.

Extensión Web

Para instalar la extensión web en nuestro navegador, primero necesitamos descargar su código fuente del repositorio.

Extensión Web – Link del repositorio público

Una vez descargado, podremos comenzar con la instalación en el web browser. Para este trabajo se pudo determinar que la extensión funciona correctamente tanto para **Mozilla Firefox** como para **Google Chrome**.

La forma de instalación en estos web browsers es la siguiente:

Para Mozilla Firefox

Debemos entrar a la siguiente url:

about:debugging#/runtime/this-firefox

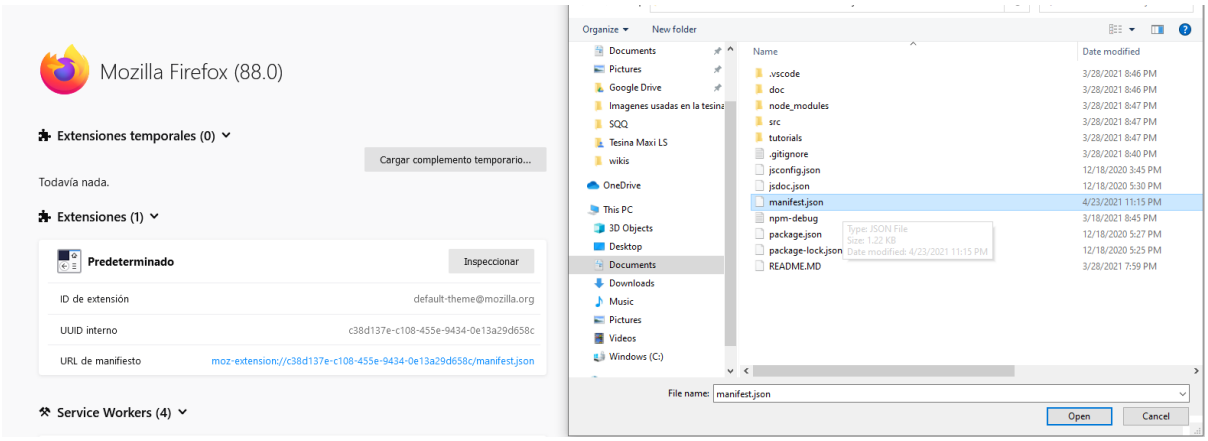

Figura 1: Cargar extensiones temporales en Firefox.

Como vemos en la Figura 1 podremos instalar extensiones temporales. Seleccionamos “Cargar complemento temporario” y seleccionamos el archivo *manifest.json* de nuestra extensión web.

Esto instalará la extensión web “*Accessibility-BadSmells-Finder*”, Figura 2, la cual ya estará reportando una vez que ingresemos a un sitio web.

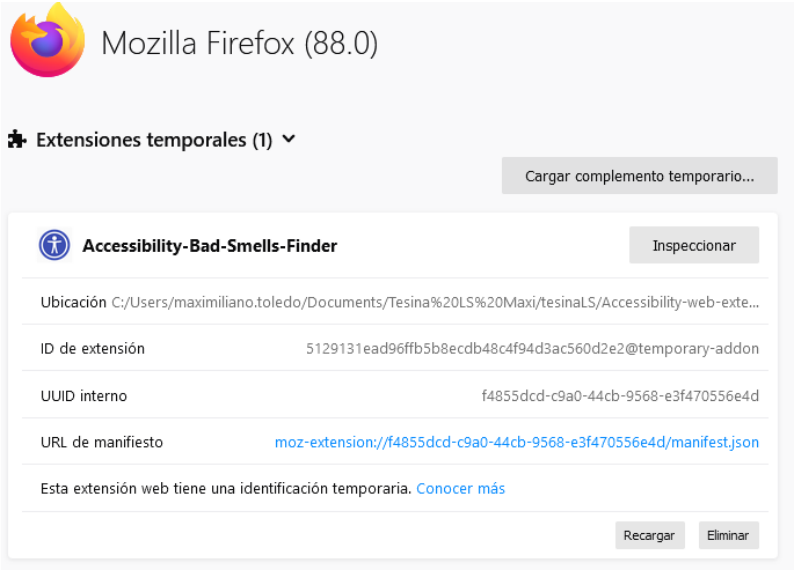

Figura 2: Web extension instalada.

En Google Chrome

Debemos entrar a la siguiente url dentro del browser:

chrome://extensions/

Luego, como vemos en la Figura 3, debemos seleccionar “Load unpacked” (paso #1). Nos pedirá seleccionar la carpeta donde se encuentra el código fuente de la extensión web (paso #2).

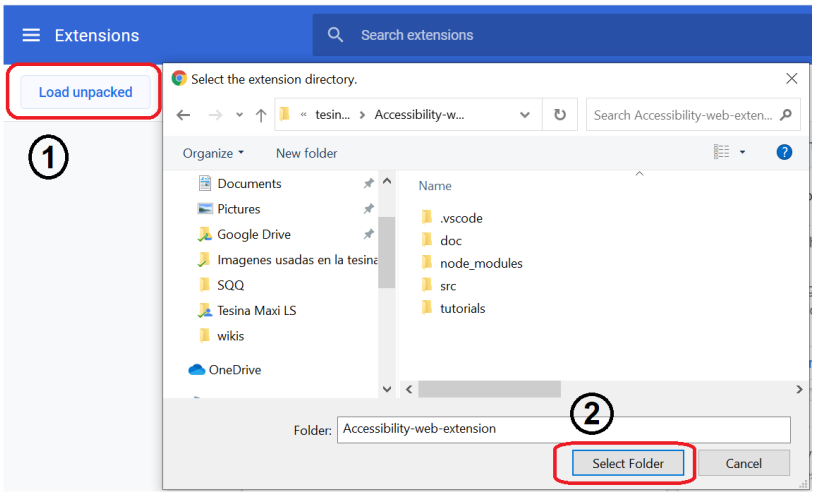

Figura 3: Cargar extensiones temporales en Chrome.

Luego de esto, la instalación de la extensión ya estará completada, Figura 4.

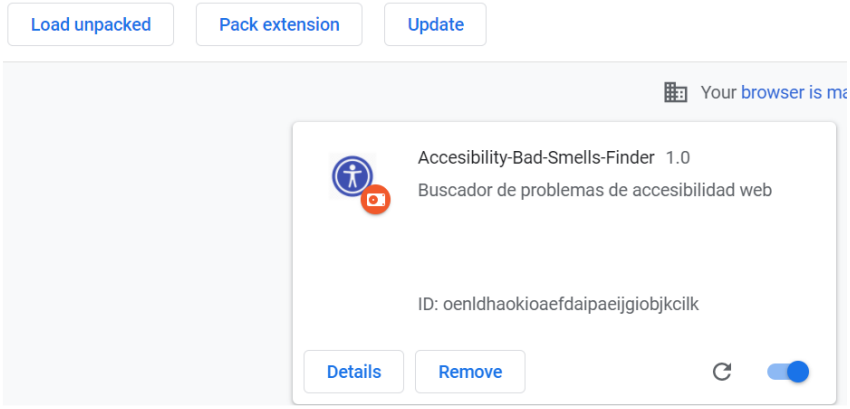

Figura 4: Extensión web instalada correctamente en Chrome.

Para validar si la extensión está funcionando, se puede comprobar en la consola del navegador los logs que la misma va logeando: ejemplo en la Figura 5.

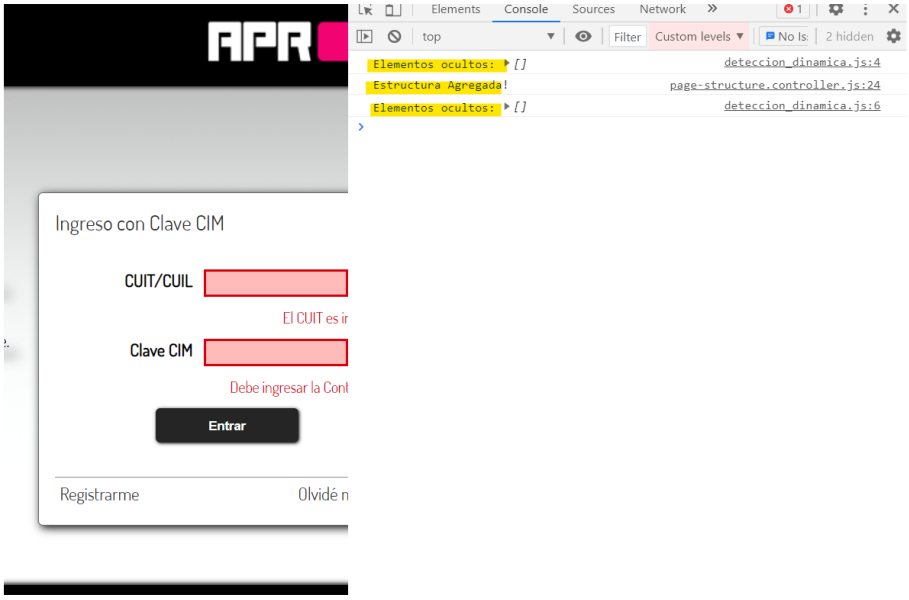

Figura 5: Ejemplo de los logs que va generando la extensión web del a herramienta de detección.

Nota: Por defecto la extensión web está configurada para ejecutarse en cualquier sitio web, pero esto se puede restringir a páginas web especificas en la sección "*matches*" del archivo *manifest.json*.

API REST y Base de datos

Base de datos

Solo se debe instalar MongoDB, y luego en una consola del sistema ejecutar el comando ***mongod***. Como observamos en la Figura 6.

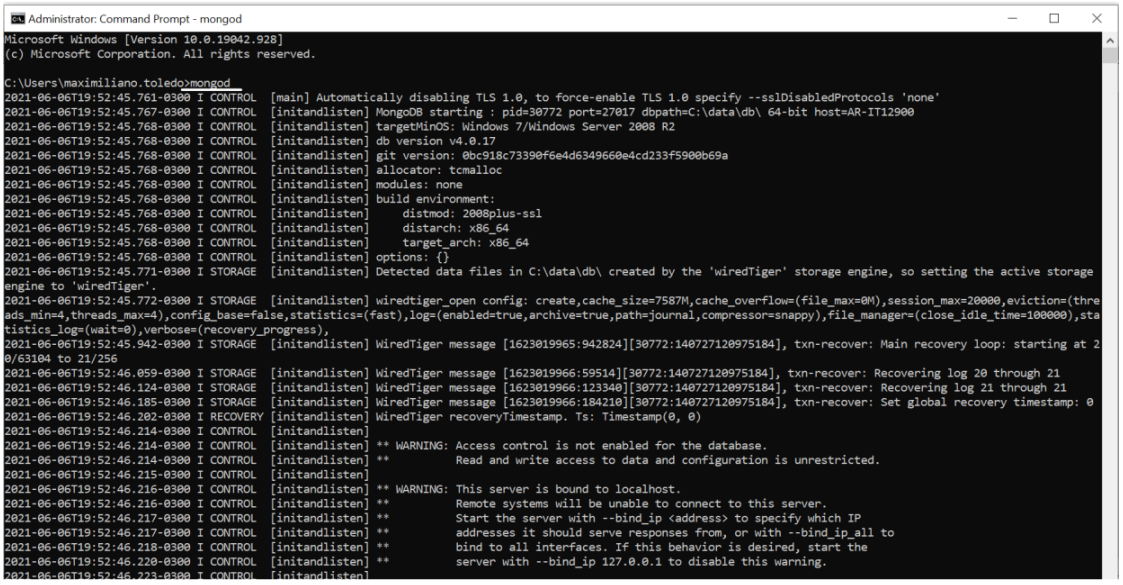

Figura 6: Ejecutando la base de datos localmente utilizando el comando *mongod*.

API REST

Para tener en funcionamiento la API REST, primero necesitamos descargar su código fuente del repositorio.

API REST – Link del repositorio público

Una vez descargado, podremos comenzar con la instalación de las dependencias de Node que utiliza esta API. Para esto necesitamos:

1. Abrir una consola de sistema en la carpeta del código fuente de la API.
2. En la consola ejecutar el comando ***npm install***
3. Una vez que termine la instalación, podremos ejecutar la API REST con el comando ***npm run dev***

Con esto ya estaríamos permitiendo que la extensión web almacene eventos, estructuras, smells y demás información en la base de datos a través de la API REST.

Aplicación de Reportes

Para tener el entorno completo de la herramienta en ejecución, debemos tener la aplicación de reportes instalada y ejecutándose para lograr visualizar el resultado final expresado en un reporte.

Los pasos son tan simples como con la API REST:

1. Descargamos su código fuente:

   APP de reportes – Link del repositorio público

2. Abrir una consola de sistema en la carpeta donde descargamos su código fuente.
3. En la consola ejecutar el comando ***npm install*** para descargar todas sus dependencias.
4. Una vez que termine la instalación, podremos ejecutar la aplicación de reportes con el comando ***ng serve***

Para visualizar la aplicación debemos ingresar a *`http://localhost:4200/`* en cualquier navegador web, como se observa en la Figura 7.

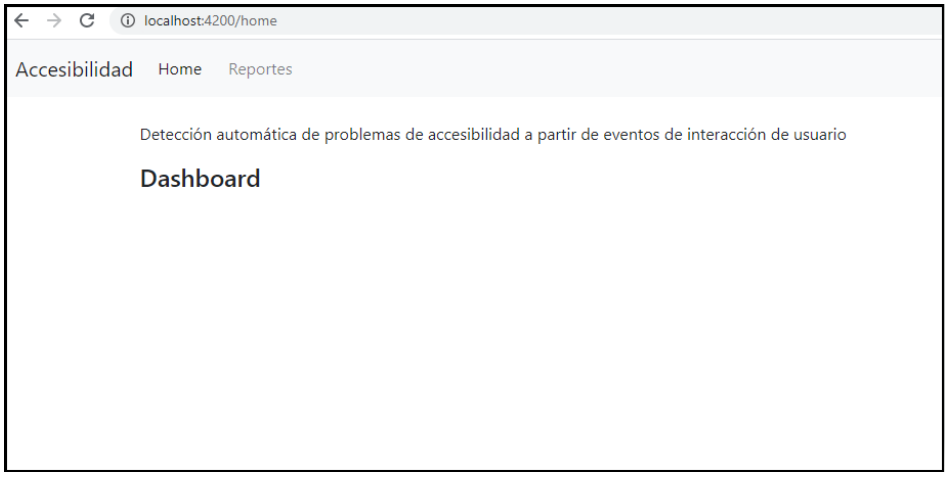

Figura 7: Ingresando a la Aplicación de reportes a través del navegador web.

Conclusiones

El trabajo desarrollado permite a los desarrolladores y dueños de aplicaciones web, obtener reportes de problemas de accesibilidad web complejos, que no pueden ser detectados mediante el análisis estático de código. La herramienta resultante es de gran utilidad para mejorar la accesibilidad web, debido a su facilidad para ser extendida con nuevos algoritmos de búsqueda, y a su adaptabilidad para diferentes escenarios de búsqueda.

Trabajos Realizados

- Análisis de herramientas online actuales de accesibilidad web.
- Búsqueda y análisis de problemas de accesibilidad utilizando un lector de pantalla.
- Desarrollo de los algoritmos “finder” para la detección de diferentes tipos de problemas de accesibilidad web.
- Desarrollo de una herramienta para la detección, almacenamiento y reporte de problemas de accesibilidad web, compuesta por una extensión web, una base de datos, una API REST y una aplicación de reportes.
- Realización de pruebas de la herramienta en diferentes sitios web para analizar el cumplimiento de sus objetivos.

Trabajos Futuros

- Ampliar el alcance de la herramienta para poder aplicar refactorings y así poder solucionar problemas de accesibilidad.
- Mejora de la interfaz de la aplicación de reportes: agregar roles de usuarios y mejora de los reportes.
- Escalar el alcance de la herramienta a un ambiente de producción.
- Continuar con la mejora de los finders desarrollados para ampliar el alcance a más tipos de elementos web.
- Continuar con el desarrollo de nuevos finders para otros tipos de elementos web inaccesibles.