Reporte de experiencia de la herramienta de pruebas para exploración de GUI de aplicaciones web RIPuppet

1. **Herramientas Utilizadas:**
   - Ghost cms: Instalamos la version de ghost que esta en docker (https://hub.docker.com/_/ghost/)
   - RIPuppet: Instalamos la herramienta proporcionada clonando el repositorio (https://github.com/TheSoftwareDesignLab/RIPuppetCoursera), utilizando la version 12.22.12 de NodeJs con el apoyo de NVM (Node version manager).
2. **Pasos de Instalación:**
   - Instalamos el contenedor que tiene la imagen de ghost con el siguiente comando: docker run -d --name some-ghost -e NODE_ENV=development -e url=http://localhost:3001 -p 3001:2368 ghost
   - Nos dirigimos a http://localhost:3001/ghost, y creamos el sitio con los siguientes datos:
     - Nombre del sitio: Pruebas
     - Usuario: ca.ariasv1@uniandes.edu.co
     - Contraseña: Carias12345
   - Clonamos el repositorio de RIPuppet, e instalamos las dependencias
   - En el archivo index, en la linea 84, antes de que se llame recursivamente al metodo "recursiveExploration", agregamos el codigo para que el script pueda realizar el login:

```
await page.goto(baseUrl);

await new Promise(r => setTimeout(r, 7000));

await page.type('id="identification"', 'ca.ariasv1@uniandes.edu.co');

await page.type('id="password"', ' Carias12345);

await page.screenshot({ path: '001.png' });

await page.click('id="ember5"');

await new Promise(r => setTimeout(r, 7000));

await page.screenshot({ path: '002.png' });

baseUrl = `${baseUrl}#/dashboard`
```

   - Ejecutamos el script: node index.js
3. **Resultados:**

   RiPuppet se ejecuta correctamente en los navegadores chromium y webkit, para lo cual la aplicación genera un reporte que se visualiza siguiendo los pasos:

   En una terminal, ubíquese en el directorio results. Desde allí, ejecute el siguiente comando para instalar la librería http-server de Node.js:

   ```
   npm install -g http-server
   ```

Ahora que cuenta con esta herramienta en su computadora, ejecute el siguiente comando para crear un servidor local con los archivos del directorio actual:

```
http-server
```

Al ejecutar los comandos anteriores se muestra lo siguiente:

Una vez se realice lo anterior podrá explorar el Index de resultados en cualquiera de los siguientes enlaces:

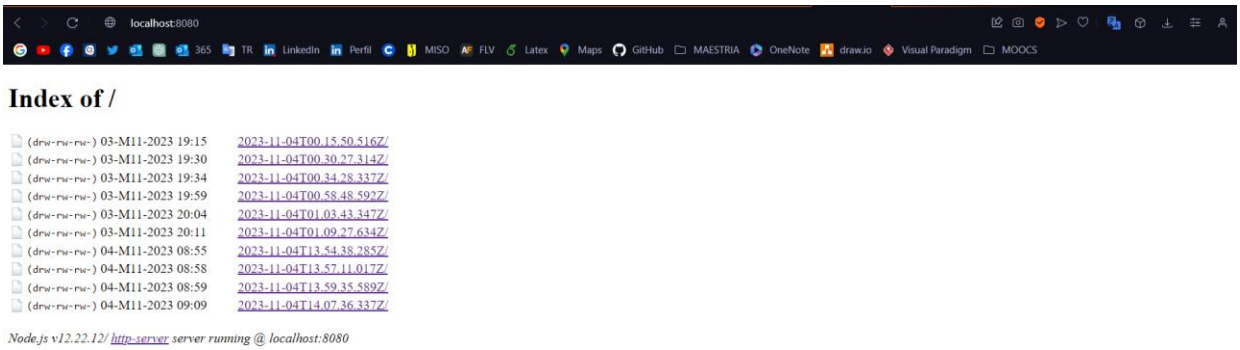
http://192.168.1.32:8080

http://127.0.0.1:8080

http://192.168.96.1:8080

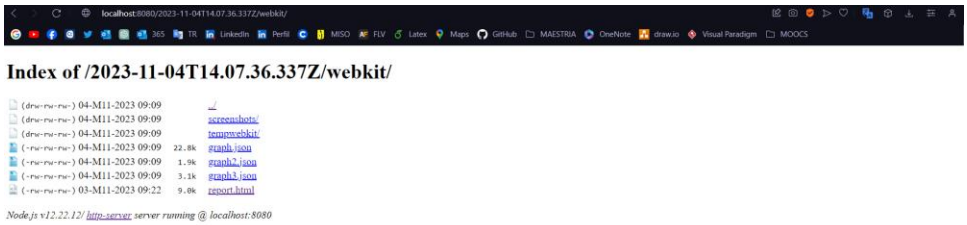
http://localhost:8080

Lo cual presenta una interfaz en la cual se encuentra una lista histórica de pruebas realizadas:

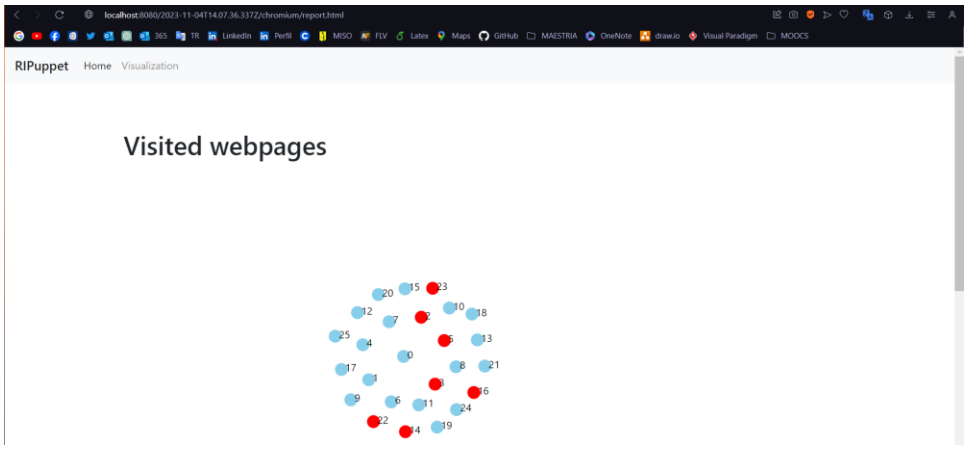


Se entrar dando click en la prueba a revisar donde se encuentra los navegadores donde se ejecuto la prueba, que para este caso se ejecuta correctamente en Chromium y en Webkit, en Firefox se queda la ejecucion y no es posible generar el reporte.

Dentro del reporte de cada navegador encontramos diferentes directorios donde podemos revisar individualmente los diferentes archivos generados por la prueba.



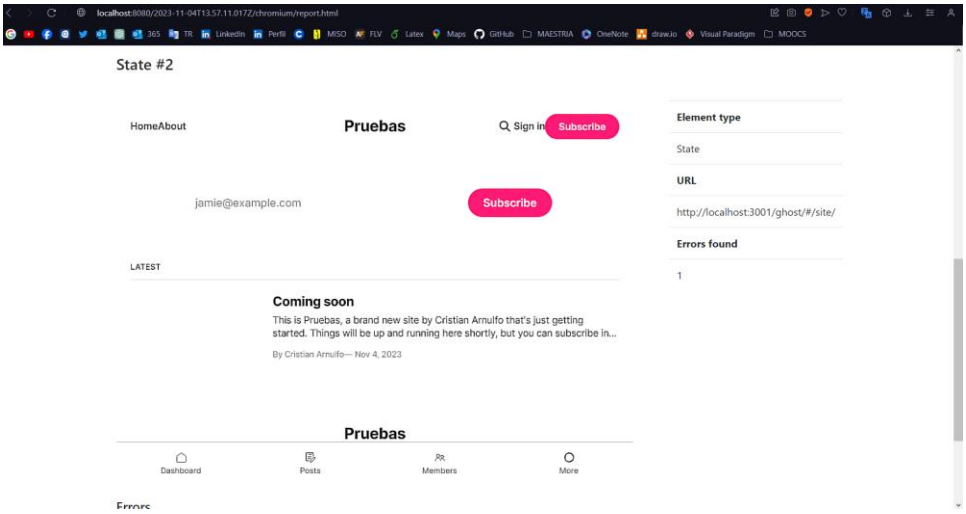
El reporte completo se encuentra en la sección report.html.



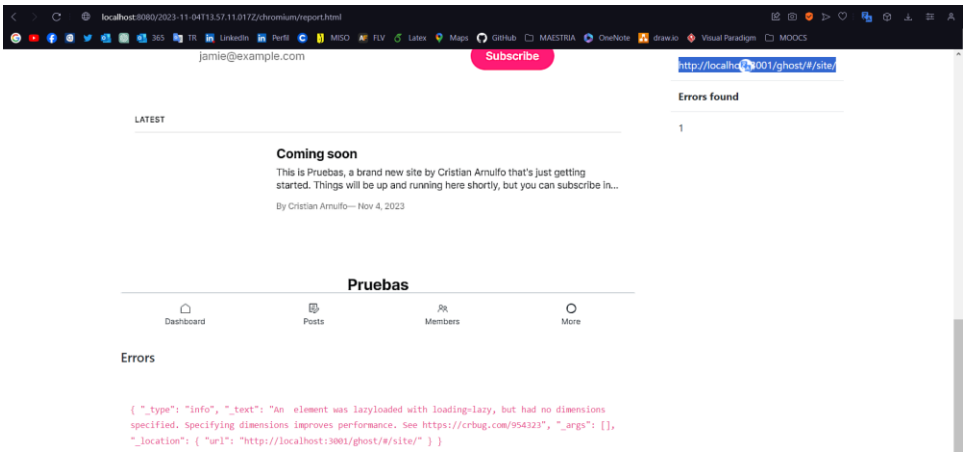
- **Chromium**

Dentro del reporte que encontramos en chromium se observa que tiene un alto numero de paginas a las que logra entrar, dependiendo del nivel de profundidad, estas paginas son representadas por un circulo ya sea de color azul o rojo, con un numero, iniciando en el login, con el numero 0 y un circulo de color azul si logra loguearse.

Después de lograr loguearse inicia a probar las diferentes interfaces de la aplicación entrando aleatoriamente a diferentes paginas en diferente orden. Dentro de esta al dar click en los diferentes círculos en la parte inferior de la pagina encontramos el reporte de la visita por la pagina.



En la que encontramos el numero del State seleccionado, un screenshot de la vista de la ventana, el tipo de elemento, la URL, y los errores encontrados, para el caso de ejemplo en el State #2 encontro 1 error.



El Erro encontrado fue:

```
{ "_type": "info", "_text": "An  element was lazyloaded with loading=lazy, but had no dimensions specified. Specifying dimensions improves performance. See https://crbug.com/954323", "_args": [], "_location": { "url": "http://localhost:3001/ghost/#/site/" } }
```

Este se encuentra dentro del “SITE”, el cual nos indica que las dimensiones no se encontraban especificadas y que Especificar las dimensiones mejora el rendimiento, adicional nos da la URL donde se encuenrra el error y el argumento.

Entre otros errores podemos encontrar los siguientes:

Ubicación: editor/post/

```
{ "_type": "info", "_text": "Autofocus processing was blocked because a document already has a focused element.", "_args": [], "_location": { "url": "http://localhost:3001/ghost/#/editor/post/" } }
```

En el que nos indica que el procesamiento de enfoque automático se bloqueó porque un documento ya tiene un elemento enfocado.

Ubicación: Servicio externo.

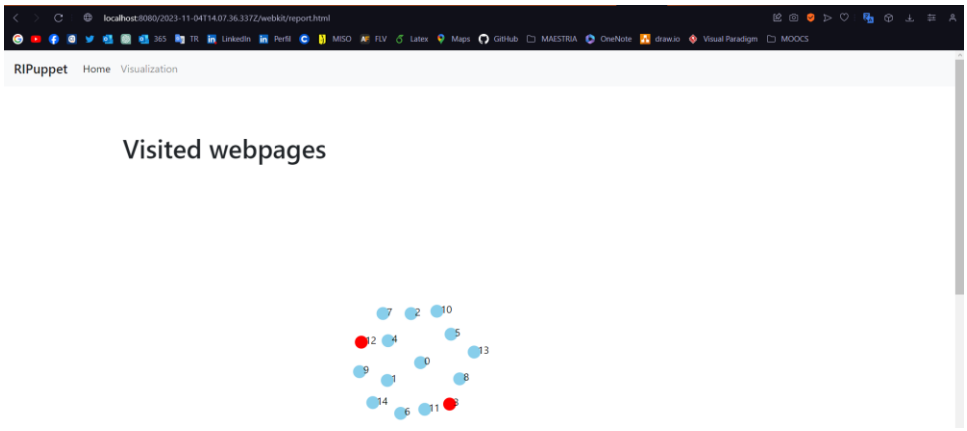
```
{ "_type": "error", "_text": "Failed to load resource: the server responded with a status of 403 ()", "_args": [], "_location": { "url": "https://resources.ghost.io/resources/ghost/api/admin/users/me/" } }
```

En el que nos indica que el servicio presenta: error al cargar el recurso: el servidor respondió con un estado de 403 ()

- **Webkit**

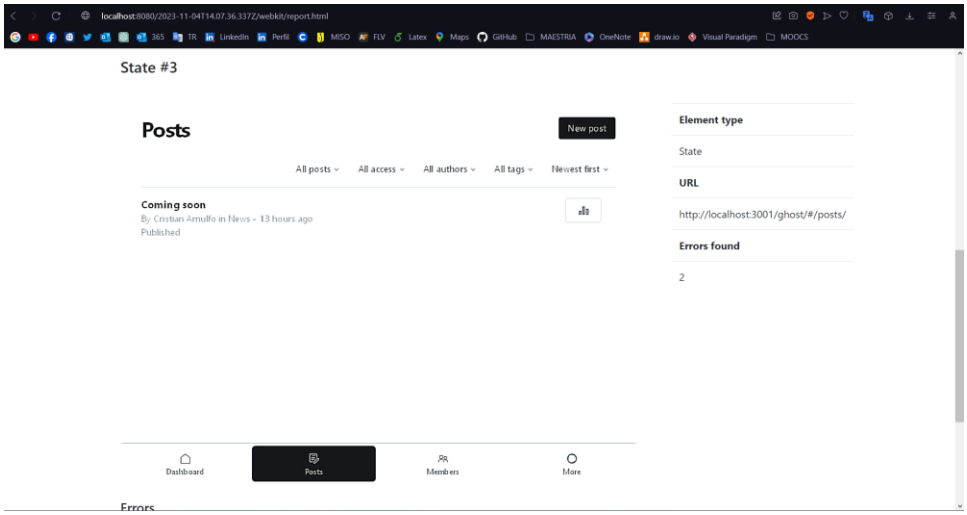
Dentro del reporte que encontramos en webkit se observa que tiene un alto numero de paginas a las que logra entrar aunque menor que en chromium, dependiendo del nivel de profundidad, estas paginas

son representadas por un circulo ya sea de color azul o rojo, con un numero, iniciando en el login, con el numero 0 y un circulo de color azul si logra loguearse.



Después de lograr loguearse inicia a probar las diferentes interfaces de la aplicación entrando aleatoriamente a diferentes paginas en diferente orden. Dentro de esta al dar click en los diferentes círculos en la parte inferior de la pagina encontramos el reporte de la visita por la pagina.

En la que encontramos el numero del State seleccionado, un screenshot de la vista de la ventana, el tipo de elemento, la URL, y los errores encontrados, para el caso de ejemplo en el State #3 encontro 2 errores.



Los 2 errores encontrados son:

{ "_type": "error", "_text": "Viewport argument key \"minimal-ui\" not recognized and ignored.", "_args": [], "_location": { "url": "http://localhost:3001/ghost/#/posts/", "lineNumber": 14, "columnNumber": 141 } }

En el que nos indica que la clave de argumento de la ventana gráfica \"minimal-ui\" no se reconoce y se ignora, la cual se encuentra sobre la ubicación de POSTS.

{ "_type": "error", "_text": "Failed to load resource: SSL peer certificate or SSH remote key was not OK", "_args": [], "_location": { "url":

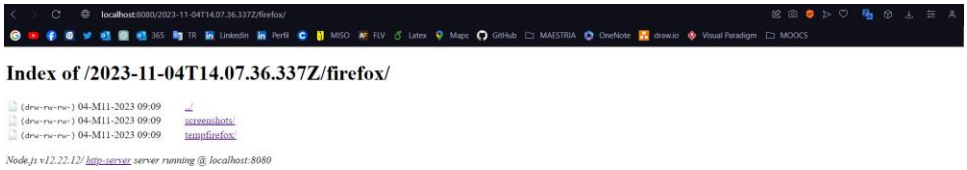
"https://ghost.org/changelog.json", "lineNumber": 0, "columnNumber": 0 } }

En el que nos indica que presenta error al cargar el recurso: el certificado SSL de igual o la clave remota SSH no estaba bien, que se encuentra en la url: https://ghost.org/changelog.json.

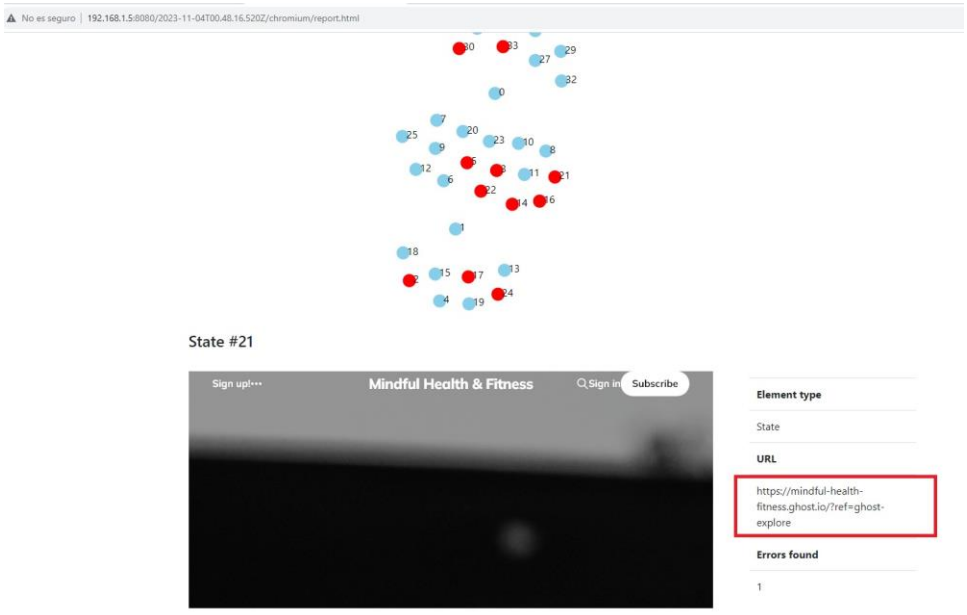
- **Firefox**

Para el caso de firefox inicia la ejecucion de la prueba pero llegado a un punto esta se queda pegada y se debe forzar la finizacion de la ejecucion de la prueba.

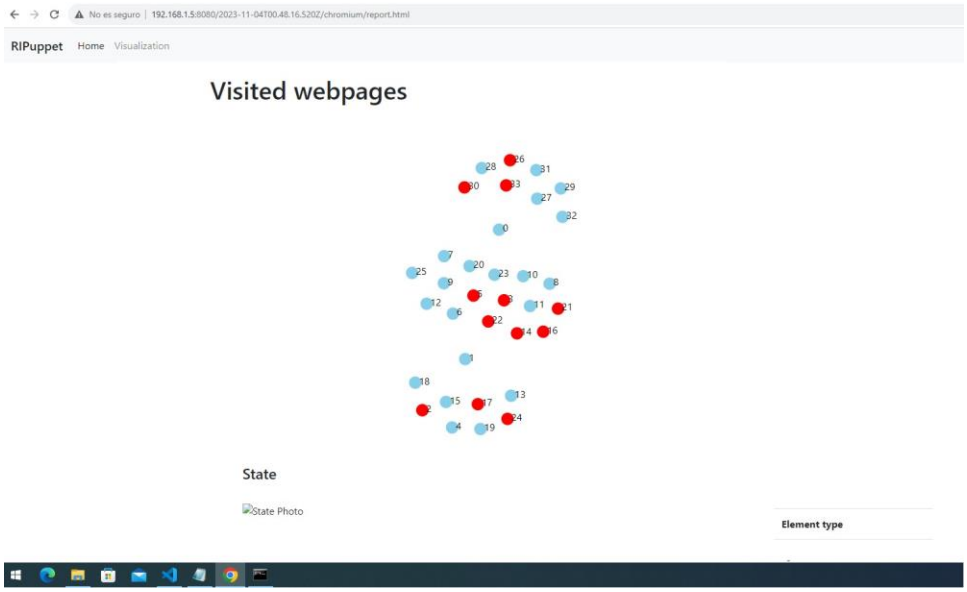
Esta ejecucion alcanza a generar los directorios pero al no poder ejecutar pruebas, no genera archivos ni reportes.



4. **Analisis:**
    - **Proceso de instalacion:** Tuvimos bastantes dificultades para instalar las dependencias de RIPuppet, probamos varias versiones de NodeJs (21, 20 y 18) hasta utilizar la correcta (12.22.12) utilizando NVM (Node version manager), como las pruebas se realizaron en el mismo computador, se opto por instalar GHOST en su version para docker.
    - **Ejecucion:** Se pudo apreciar que el recorrido de ejecucion varia, no siempre se visitan las paginas en el mismo orden, y tampoco el tiempo de ejecucion en cada ejecucion es el mismo.
    - **Navegación Externa:** RIPuppet al empezar a visitar la estructura de la aplicación empieza a visitar links externos, incluso reporta errores encontrados en estos links:



    - **Navegadores:** Se realizaron las pruebas en los navegadores Chromium y WebKit, al ejecutar las pruebas exitosamente, en Firefox la ejecucion se quedaba congelada, no llegamos a determinar la causa del error.
    - **Profundidad:** La variable profundidad se modificó, obteniendo como resultado una exploración del grafo mas amplia, por ende las pruebas toman mayor tiempo.
    - **Errores:** Al visualizar los reportes de resultados, se pudo apreciar que las pruebas exitosas se marcan con puntos azules y los errores con puntos rojos, sin embargo el contenido del error no nos pareció muy claro debido a que utiliza un lenguaje muy técnico.


No es seguro | 192.168.1.5:8080/2023-11-04T00.48.16.520Z/chromium/report.html
RIPuppet
Home
Visualization
Visited webpages
State
State Photo
Element type