

**UNIVERSIDAD SANTO TOMÁS**

**FACULTAD DE INGENIERÍA EN SISTEMAS**

**INFORME DE PRUEBAS DE SOFTWARE**

**AUTORES:**

**Juan Manuel Poveda Rodríguez**

**Richard Eduardo Molano Cubillos**

**Juan David Mozo Joya**

**Tunja 2025**

<b>Tabla de contenido:</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>3</b>
<b>3. Metodología</b>	<b>3</b>
<b>4. Pruebas con Qualys ssL labs</b>	<b>4</b>
<b>5. Pruebas con Selenium</b>	<b>7</b>
Elementos usados	8
Funcionamiento	8
<b>6. Pruebas de funcionalidad en lighthouse de Google</b>	<b>14</b>
<b>7. Pruebas de validación con W3C</b>	<b>16</b>
Resultados:	16
<b>8. Prueba de esfuerzo(k6)</b>	<b>19</b>
<b>9. Conclusiones</b>	<b>21</b>
<b>10. Referencias</b>	<b>21</b>

## I. Introducción

Se desarrollan las pruebas de calidad sobre el software Votum con el propósito de evaluar su calidad, rendimiento y conformidad con estándares establecidos por la ISO 9001:2015. Se utilizaron herramientas especializadas para analizar distintos aspectos del software y con ello identificar los problemas y encontrar las posibles soluciones para dar solución y mejorar la calidad del software en cuestión.

## II. Objetivos

### Objetivos generales

Desarrollar la evaluación de manera íntegra al software mediante la ejecución de pruebas de calidad sistemáticas con el fin de identificar defectos y verificar el cumplimiento de los requerimientos del programa.

### Objetivos específicos

- Verificar la correcta estructura del código HTML del sistema mediante el uso del validador W3C, con el fin de garantizar la conformidad con los estándares web.
- Evaluar el rendimiento de la aplicación bajo condiciones de carga mediante la herramienta k6, con el propósito de identificar posibles cuellos de botella.
- Analizar la funcionalidad general del sistema utilizando Google Lighthouse, enfocándose en aspectos como el rendimiento, accesibilidad, buenas prácticas y posicionamiento SEO.
- Automatizar escenarios de prueba funcional mediante Selenium IDE para validar el comportamiento de la aplicación desde la perspectiva del usuario final.
- Comprobar la configuración de

seguridad del protocolo HTTPS en el servidor utilizando Qualys SSL Labs, asegurando el uso de cifrados seguros y protocolos actualizados.

## III. Metodología

Para la ejecución de las pruebas, se adoptó un enfoque sistemático y estructurado compuesto por las siguientes etapas:

- **Definición de criterios de evaluación:**

Se establecen los parámetros específicos a validar en función de los objetivos propuestos y las capacidades de cada herramienta.

- **Selección e implementación de herramientas:**

Se elegirán herramientas de uso común en la industria para la ejecución de pruebas automatizadas, de rendimiento, validación de código y análisis de seguridad.

- **Desarrollo y ejecución de pruebas:**

Se procederá con la configuración y ejecución de las pruebas conforme a los criterios definidos, garantizando un entorno controlado y repetible.

- **Documentación de resultados:**

Los hallazgos obtenidos durante las pruebas son registrados mediante capturas de pantalla, informes automáticos y descripciones técnicas.

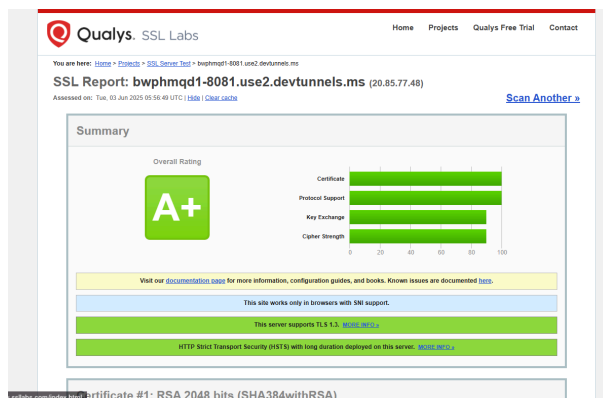
- **Análisis e interpretación de datos:**

Finalmente, los resultados son analizados sin realizar ajustes en el código fuente, permitiendo identificar áreas críticas de mejora y el comportamiento actual del sistema bajo evaluación.

## IV. Pruebas con Qualys ssl labs

Con **Qualys SSL Labs** se analizará la seguridad del dominio HTTPS con los cifrados ofrecido debido a que el servidor utilizado es un servidor local en este caso usamos un protocolo HTTPS donde la ruta es <https://bwphmqd1-8081.use2.devtnnls.ms/> donde podemos que utilizamos un dev tunnels ofrecido por VS Code para el despliegue simple y un manejo de SPA para así mantener una seguridad también en los enrutados donde podemos ver que por medio de esto la navegación no nos da endpoints.

Se muestra el resultado de una prueba de seguridad SSL/TLS realizada con la herramienta **Qualys SSL Labs** sobre el dominio:



**bwphmqd1-8081.use2.devtnnls.ms**  
**Fecha del análisis:** 3 de junio de 2025,  
05:56:49 UTC.

del resultado

**Calificación general: A+**

Esta es la puntuación más alta posible, lo que indica que el servidor está muy bien configurado en términos de seguridad TLS/SSL.

## Componentes evaluados:

**Certificate:** Cerca del 100%  
(certificado válido y bien configurado).

**Protocol Support:** 100% (soporta protocolos modernos y seguros).

**Key Exchange:** Alto puntaje  
(intercambio de claves seguro).

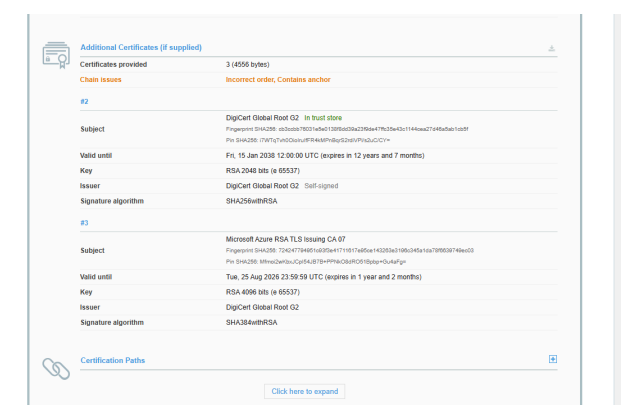
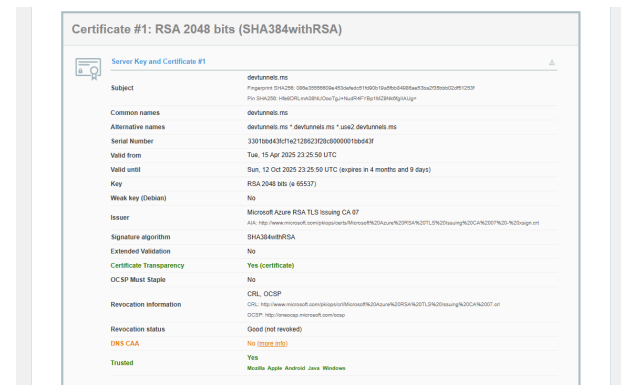
**Cipher Strength:** Alto puntaje (uso de cifrados fuertes).



El servidor **soporta exclusivamente TLS 1.3**, qué es la versión más segura del protocolo actualmente.

El sitio **funciona solo en navegadores con soporte para SNI** (Server Name Indication).


Tiene implementado **HSTS (HTTP Strict Transport Security)** con una duración prolongada, lo que previene ataques de tipo downgrade y MITM.

## Evidencias:





Configuration																			
	<h3>Protocols</h3> <table><tbody><tr><td>TLS 1.3</td><td>Yes</td></tr><tr><td>TLS 1.2</td><td>Yes</td></tr><tr><td>TLS 1.1</td><td>No</td></tr><tr><td>TLS 1.0</td><td>No</td></tr><tr><td>SSL 3</td><td>No</td></tr><tr><td>SSL 2</td><td>No</td></tr></tbody></table> <p>(*) Experimental: Server negotiated using No-DHE</p>	TLS 1.3	Yes	TLS 1.2	Yes	TLS 1.1	No	TLS 1.0	No	SSL 3	No	SSL 2	No						
TLS 1.3	Yes																		
TLS 1.2	Yes																		
TLS 1.1	No																		
TLS 1.0	No																		
SSL 3	No																		
SSL 2	No																		
	<h3>Cipher Suites</h3> <table><tbody><tr><td colspan="2"><b># TLS 1.3 (suites in server preferred order)</b></td></tr><tr><td>TLS_AES_256_GCM_SHA384 (h1302)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS 256</td></tr><tr><td>TLS_CHACHA20_POLY1305_SHA256 (h1303)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS 256</td></tr><tr><td>TLS_AES_128_GCM_SHA256 (h1301)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS 128</td></tr><tr><td colspan="2"><b># TLS 1.2 (suites in server preferred order)</b></td></tr><tr><td>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (h1009)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS 256</td></tr><tr><td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (h1008)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS 256</td></tr><tr><td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (h1005)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 256</td></tr><tr><td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (h1007)</td><td>ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 128</td></tr></tbody></table>	<b># TLS 1.3 (suites in server preferred order)</b>		TLS_AES_256_GCM_SHA384 (h1302)	ECDH:ECDH (suw, 3072 bit RSA) PS 256	TLS_CHACHA20_POLY1305_SHA256 (h1303)	ECDH:ECDH (suw, 3072 bit RSA) PS 256	TLS_AES_128_GCM_SHA256 (h1301)	ECDH:ECDH (suw, 3072 bit RSA) PS 128	<b># TLS 1.2 (suites in server preferred order)</b>		TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (h1009)	ECDH:ECDH (suw, 3072 bit RSA) PS 256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (h1008)	ECDH:ECDH (suw, 3072 bit RSA) PS 256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (h1005)	ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (h1007)	ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 128
<b># TLS 1.3 (suites in server preferred order)</b>																			
TLS_AES_256_GCM_SHA384 (h1302)	ECDH:ECDH (suw, 3072 bit RSA) PS 256																		
TLS_CHACHA20_POLY1305_SHA256 (h1303)	ECDH:ECDH (suw, 3072 bit RSA) PS 256																		
TLS_AES_128_GCM_SHA256 (h1301)	ECDH:ECDH (suw, 3072 bit RSA) PS 128																		
<b># TLS 1.2 (suites in server preferred order)</b>																			
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (h1009)	ECDH:ECDH (suw, 3072 bit RSA) PS 256																		
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (h1008)	ECDH:ECDH (suw, 3072 bit RSA) PS 256																		
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (h1005)	ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 256																		
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (h1007)	ECDH:ECDH (suw, 3072 bit RSA) PS <b>WEAK</b> 128																		

Protocol Details	Supported
<b>Secure Renegotiation</b>	<b>Supported</b>
Secure Client Initiated Renegotiation	No
Insecure Client Initiated Renegotiation	No
BEAST attack	Mitigated server-side ( <a href="#">more info</a> )
POODLE (SSLv3)	No, SSLv3 not supported ( <a href="#">more info</a> )
POODLE (TLS)	No ( <a href="#">more info</a> )
Zombie POODLE	No ( <a href="#">more info</a> ) TLS 1.2: fixed
GOLDENDOODLE	No ( <a href="#">more info</a> ) TLS 1.2: fixed
OpenSSL 0 Length	No ( <a href="#">more info</a> ) TLS 1.2: fixed
Sleeping POODLE	No ( <a href="#">more info</a> ) TLS 1.2: fixed
Downgrade attack prevention	<b>Yes, TLS_FALLBACK_SVC supported (<a href="#">more info</a>)</b>
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No ( <a href="#">more info</a> )
Ticketbleed (vulnerability)	No ( <a href="#">more info</a> )
OpenSSL C.C.S vuln. (CVE-2014-0224)	No ( <a href="#">more info</a> )
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No ( <a href="#">more info</a> )
ROBOT (vulnerability)	No ( <a href="#">more info</a> )
<b>Forward Secrecy</b>	<b>Yes (with most browsers) ROBUST (<a href="#">more info</a>)</b>
ALPN	Yes: <code>h2,h3;q=1</code>
NPN	No
Session resumption (caching)	No (Dns assigned but not accepted)
Session resumption (tickets)	No
OCSP stapling	No



## HTTP Requests


<https://baphemot1.208f1.unw2.dev/annals.me/> (HTTP/1.1 200 OK)



## Miscellaneous

Test date	Tue, 03 Jun 2025 05:55:47 UTC
Test duration	61.641 seconds
HTTP status code	200
HTTP server signature	-
Server hostname	-

SSL Report v2.4.0

## Método de uso

Name your new project

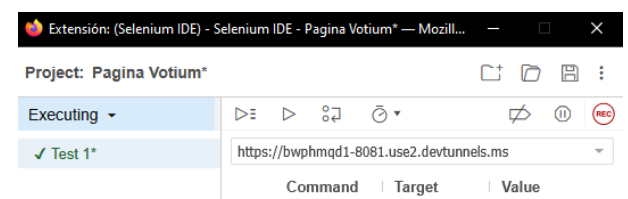
Please provide a name for your new project.

PROJECT NAME

Pagina Vptium

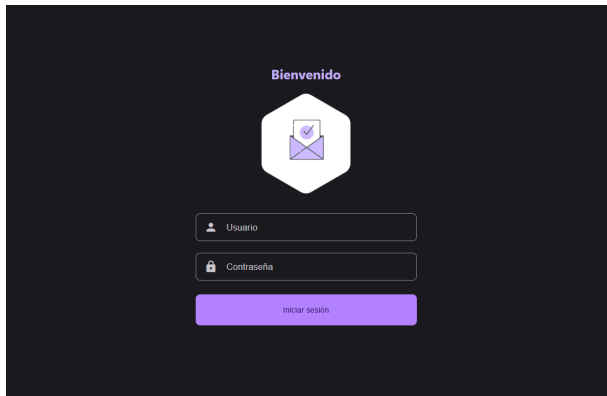
You can change the name of your project at any time by clicking it and entering a new name.

OKCANCEL

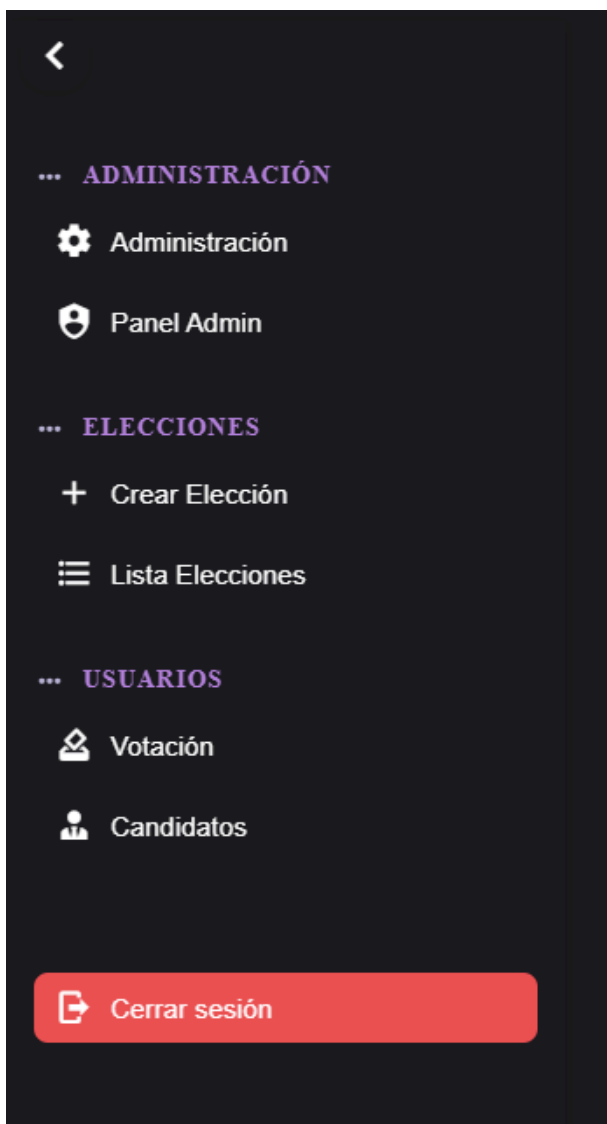


Se realizó el “**Test 1**” el cual fue usado para tomar paso a paso la funcionalidad del software

en web en Selenium, tomando en cuenta todos los casos tales como:

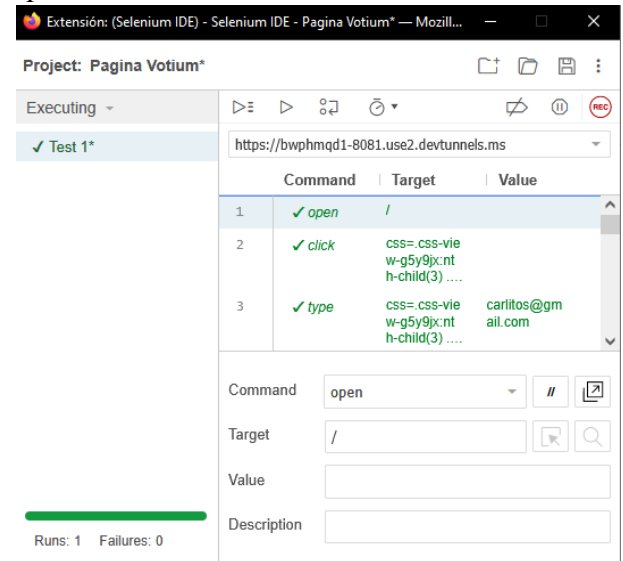


## Inicio sesión



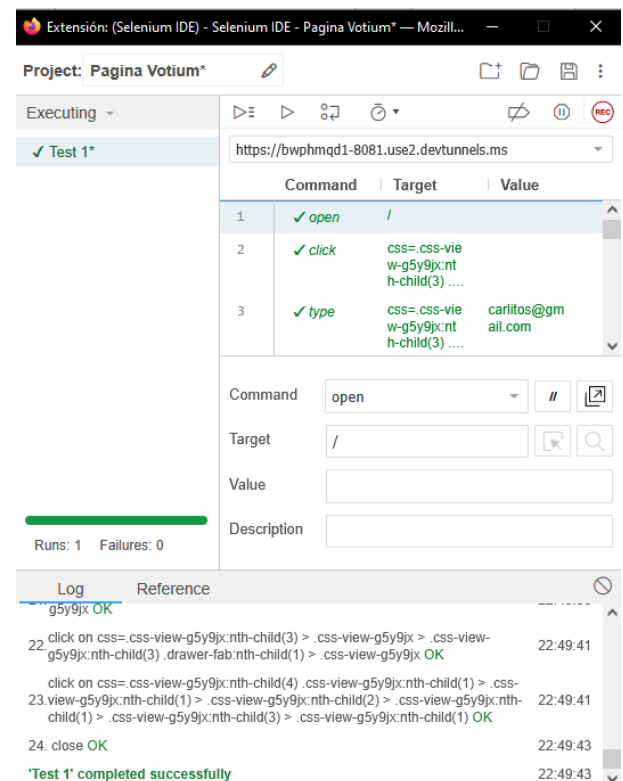
Funcionalidades en su totalidad de la

aplicación.



Se le dio fin a la grabación y se ejecutó el análisis automático para que la máquina revisará el programa por sí mismo.

## Resultados

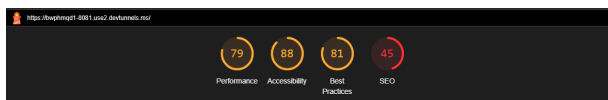


El análisis realizado mediante “Selenium IDE” se ejecutó correctamente y permitió verificar el comportamiento esperado de la aplicación. Durante la prueba, todas las acciones configuradas —incluyendo la apertura de la página, la interacción con los

elementos y la introducción de datos en los campos correspondientes— se completaron sin errores. Como resultado, se obtuvo una ejecución satisfactoria, cumpliendo con los objetivos establecidos para este caso de prueba, lo que confirma que el sistema respondió de manera adecuada ante los escenarios evaluados.

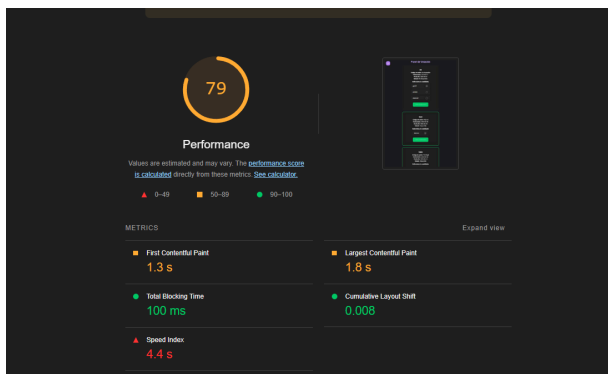
## Pruebas de funcionalidad en Lighthouse de Google

Esta vista muestra el global del análisis realizado sobre la URL <https://bwphmqd1-8081.use2.devtnnls.ms/>.



Categoría	Puntuación	Evaluación
Performance	79	Aceptable
Accesibilidad	88	Buena
Buenas Prácticas	81	Aceptable
SEO	45	Deficiente

La puntuación más baja fue en SEO, lo cual indica deficiencias en etiquetas meta, títulos, accesibilidad para buscadores o estructura HTML mal definida para indexación.



Puntuación general: 79

Métricas clave:

First Contentful Paint: 1.3s (bueno)

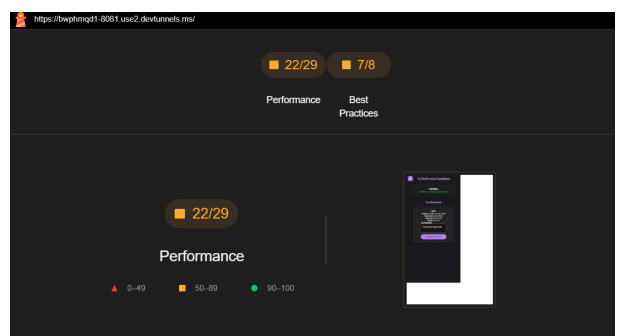
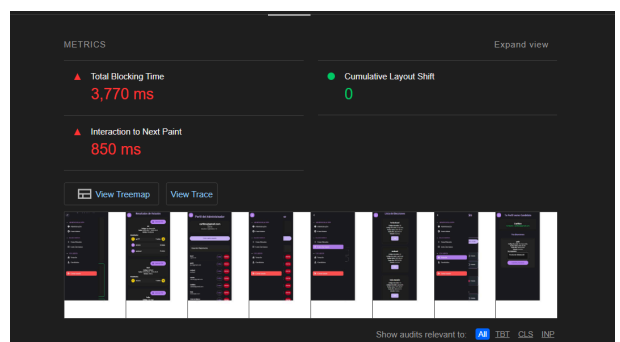
Largest Contentful Paint: 1.8s (bueno)

Total Blocking Time: 100ms

Speed Index: 4.4s (podría mejorar)

Cumulative Layout Shift: 0.008 (excelente, no hay movimientos inesperados en la interfaz)

La mayoría de las métricas son positivas, pero el *Speed Index* sugiere que la página tarda en mostrar visualmente todos los elementos clave, lo que puede afectar la percepción de rapidez por parte del usuario.



## Pruebas de validación con W3C

### Errores encontrados:

#### Uso de barras inclinadas / en etiquetas vacías

Se observaron barras inclinadas al final de etiquetas vacías como `<meta>` y `<link>`.

Aunque no afectan directamente la estructura del documento en HTML5, su uso es innecesario y puede generar problemas de interpretación cuando los atributos no están

entre comillas. En HTML5 se recomienda omitir la barra final en estos elementos.

## Uso de atributos no válidos (b-ip6mpux4kw)

Se identificó el uso del atributo b-ip6mpux4kw en etiquetas como <header>, <nav>, <div>, <button>, <span>, <ul>, <li>, y <footer>. Este tipo de atributos personalizados no están definidos por la especificación HTML y no son reconocidos por los navegadores, lo que podría provocar problemas de accesibilidad, validación o compatibilidad. Si este atributo es generado por un framework o librería, se recomienda validar su necesidad o reemplazarlo por atributos estándar o data-\*

## Uso incorrecto de la etiqueta <center>

La etiqueta <center> está obsoleta en HTML5 y no debe utilizarse. Su funcionalidad debe lograrse mediante estilos CSS, usando reglas como text-align: center; para lograr el mismo efecto de forma válida y accesible.

## Uso innecesario del atributo role="main" en el elemento <main>

El elemento <main> ya tiene implícitamente el rol de "main" en HTML5, por lo tanto, incluir role="main" es redundante. Eliminarlo mejora la limpieza del código y evita repeticiones innecesarias.

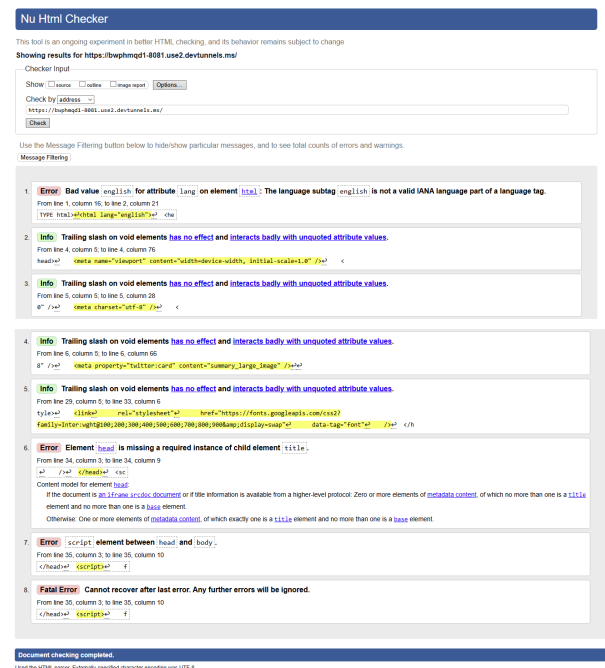
## Advertencias:

El uso de elementos obsoletos, atributos personalizados no válidos y malas prácticas en la escritura del HTML puede afectar la **compatibilidad, accesibilidad y mantenibilidad** del sitio web.

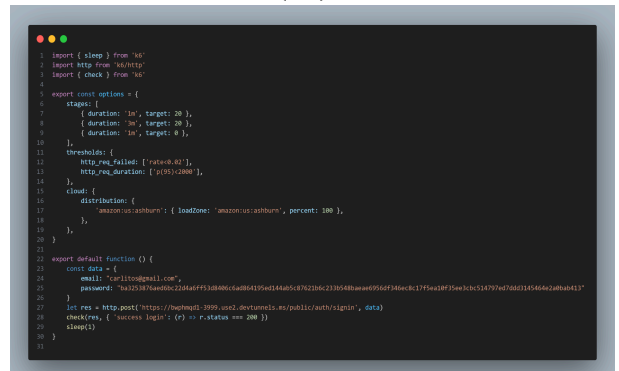
Evidencia: Se adjunta captura de pantalla con los errores detectados.

Siendo así que estos elementos se generan de esta manera debido a la construcción de React Native, así que no los podemos cambiar directamente, así que asumimos su valor y no lo tendremos en cuenta, ya que no genera errores en pantalla y tampoco a la hora de la

## experiencia del usuario



## Prueba de esfuerzo (k6)



## EXPLICACIÓN:

Este script simula usuarios concurrentes intentando iniciar sesión en una aplicación web mediante solicitudes a los endpoint de autenticación. Evalúa tanto la capacidad del servidor para manejar carga como el tiempo de respuesta.



## Importación de módulos

```
import { sleep } from 'k6'
import http from 'k6/http'
import { check } from 'k6'
```

**sleep:** Pausa entre iteraciones para simular el comportamiento real de un usuario.

**http:** Módulo para enviar solicitudes **HTTP**.

**Check:** Permite validar si la respuesta del servidor cumple ciertas condiciones (por ejemplo, si el código de estado es **200**).

## B. Configuración de la prueba (options)

```
export const options = {
  stages: [
    { duration: '1m', target:
20 },
    { duration: '3m', target:
20 },
    { duration: '1m', target:
0 },
  ],
  thresholds: {
    http_req_failed:
['rate<0.02'],
    http_req_duration:
['p(95)<2000'],
  },
  cloud: {
    distribution: {
      'amazon:us:ashburn':
{ loadZone:
'amazon:us:ashburn', percent:
100 },
    },
  },
}
```

```
}
```

Esto define:

**Stages (etapas de carga):**

Subida de usuarios: 1 minuto hasta alcanzar 20 usuarios.

Carga sostenida: 3 minutos manteniendo 20 usuarios.

Descenso: 1 minuto para volver a 0 usuarios.

**Thresholds (umbrales):**

**http\_req failed:** Se espera que menos del 2% de las solicitudes fallen.

**http\_req duration:** El 95% de las solicitudes deben completarse en menos de 2 segundos.

**Cloud distribution:** Los usuarios simulados se ejecutan desde la región us:ashburn de AWS (esto solo aplica si usas k6 Cloud).

## C. Función principal

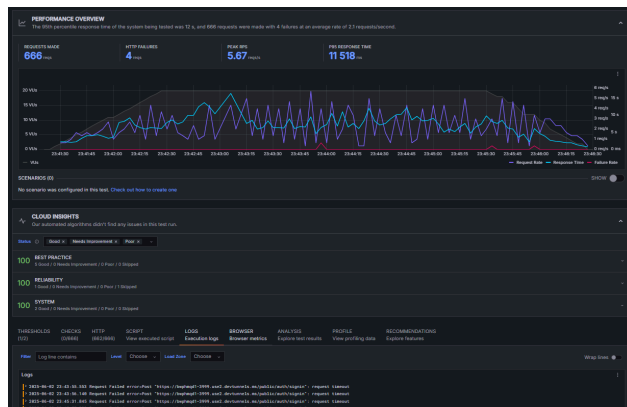
```
export default function () {
  const data = {
    email:
"carlitos@gmail.com",
    password: "..."
  }
  let res =
http.post('https://bwphmqd1-
3999.use2.devtunnels.ms/public/auth/signin', data)
  check(res, { 'success
login': (r) => r.status ===
200 })
  sleep(1)
}
```

Lo que hace esta función:

Crea un cuerpo de datos con

un correo y una contraseña encriptada.  
Envía una solicitud POST al endpoint /public/auth/signin.  
Valida la respuesta usando check: la prueba es exitosa si el código HTTP es 200 (inicio de sesión correcto).  
Espera 1 segundo antes de continuar a la siguiente iteración, para no saturar instantáneamente el servidor.

Este código permite:  
Simular 20 usuarios accediendo simultáneamente.  
Verificar que el servidor responde adecuadamente.  
Medir tiempo de respuesta y tasa de errores.  
Automatizar pruebas de login bajo carga en un entorno de prueba.



## Resultados

**Requests Made:** 666  
Total de solicitudes HTTP enviadas durante la prueba.  
**HTTP Failures:** 4  
Cuatro solicitudes fallaron (todas por **timeout**, como se

ve en los logs).  
**Peak RPS:** 5.67 req/s  
La tasa máxima de solicitudes por segundo alcanzada fue de 5.67.  
**P95 Response Time:** 11,518 ms  
El 95% de las respuestas tardaron **menos de 11.5 segundos**. Esta cifra es alta y puede indicar saturación del servidor o cuellos de botella.  
**Gráfica de líneas:**  
**Request Rate (azul claro):** frecuencia de solicitudes a lo largo del tiempo.  
**Response Time (morado):** tiempo de respuesta de cada solicitud.  
**Failure Rate (gris claro):** solicitudes fallidas en cada instante.

## A. Cloud Insights

Evaluación automática del rendimiento general por parte de k6 Cloud:

Categoría	Resultado
<b>Buenas Prácticas</b>	100 (óptimo)
<b>Fiabilidad</b>	100 (óptimo)
<b>Sistema</b>	100 (óptimo)

Esto indica que, a pesar de algunas fallas por tiempo de espera, no hubo problemas críticos de infraestructura o configuración.

## B. Logs de errores

En la parte inferior aparecen registros de las 4 fallas: nginx

Copiar Editar

Request Failed error=Post  
"https://.../signin": request  
timeout

**Causa:** El servidor no  
respondió en el tiempo  
esperado. Puede deberse a:  
Alta carga simultánea.  
Problemas en la red.  
Cuellos de botella en el  
backend.

Esta prueba muestra que el  
sistema fue capaz de manejar  
hasta **666 solicitudes** con  
solo **4 fallas (0.6%)**, lo cual  
es aceptable si se considera  
el umbral de error definido  
en el script (< 2%).  
Sin embargo, el **tiempo de  
respuesta promedio es alto  
(11.5 s para el 95%)**, lo que  
indica que la aplicación **no  
es suficientemente rápida  
bajo carga constante** y  
necesita optimizaciones de  
rendimiento.

## V. Conclusiones

A partir del análisis y resultados obtenidos  
durante la ejecución de las distintas pruebas,  
se concluye lo siguiente:

- El sistema evaluado presenta una  
configuración de seguridad robusta,  
obteniendo la máxima calificación  
(A+) en los análisis de SSL Labs, lo  
cual demuestra el cumplimiento de  
estándares modernos de seguridad  
como TLS 1.3 y HSTS.
- Las funcionalidades clave de la  
aplicación fueron validadas con éxito  
mediante pruebas automatizadas con

Selenium IDE, lo que evidencia un  
adecuado comportamiento desde el  
punto de vista del usuario final.

- El análisis con Google Lighthouse  
indica un rendimiento general  
aceptable y una buena accesibilidad;  
sin embargo, aspectos como el  
posicionamiento SEO requieren  
optimización, dado su bajo puntaje.
- La validación con W3C evidenció  
errores relacionados con el uso de  
etiquetas obsoletas y atributos no  
válidos, lo cual podría afectar la  
compatibilidad y la accesibilidad del  
sitio web.
- La prueba de carga realizada con k6  
permitió identificar una capacidad  
razonable del sistema bajo condiciones  
simuladas de uso concurrente. No  
obstante, los tiempos de respuesta  
elevados sugieren la necesidad de  
optimizar ciertos procesos para  
mejorar la experiencia del usuario bajo  
carga.

En conjunto, los resultados permiten tener una  
visión clara del estado actual del sistema, sus  
fortalezas en términos de seguridad y  
funcionalidad

Donde es clave ver y analizar los datos y  
notamos que el producto aunque tiene buenas  
aptitudes no cumple correctamente con los  
estándares de calidad debido a su velocidad y  
su formato al estar subido en una plataforma  
local para hacer hosting tenemos que no es lo  
suficientemente potente para realizar pruebas  
con gran esfuerzo y también notamos que react  
native el desarrollo principal para este  
programa es para dispositivos móviles donde  
vemos que no podemos realizar de igual

manera esta pruebas

Recuperado de

<https://www.sonarqube.org/>

- Selenium. (s.f.). Selenium. Recuperado de <https://www.selenium.dev/>

## VI. Referencias

[1] Google Developers, "Lighthouse: Open-source, automated tool for improving the quality of web pages". [En línea]. Disponible en: <https://developer.chrome.com/docs/lighthouse/>

[2] Grafana Labs, "k6 - Performance testing for developers". [En línea]. Disponible en: <https://k6.io/>

[3] Qualys, Inc., "SSL Server Test - Qualys SSL Labs". [En línea]. Disponible en: <https://www.ssllabs.com/ssltest/>

[4] Selenium, "Selenium IDE". [En línea]. Disponible en: <https://www.selenium.dev/selenium-ide/>

[5] W3C, "Markup Validation Service". [En línea]. Disponible en: <https://validator.w3.org/>

[6] Apache Software Foundation, "Apache JMeter Documentation". [En línea]. Disponible en: <https://jmeter.apache.org/>

[7] SonarSource, "SonarQube". [En línea]. Disponible en: <https://www.sonarqube.org/>

World Wide Web Consortium (W3C). (s.f.). W3C Validator. Recuperado de <https://validator.w3.org/>

- Apache JMeter Documentation . (s.f.). Apache JMeter. Recuperado de <https://jmeter.apache.org/>
- SonarSource. (s.f.). SonarQube.