Autenticacion con Redes Sociales en Node.js

Jose Daniel Mora Solano
ULACIT

San Jose, Costa Rica
jmoras612@ulacit.ed.cr

Rolando León Gamboa
ULACIT

San Jose, Costa Rica
rleong719@ulacit.ed.cr

***Resumen* — Este documento presenta puntos clave sobre los fundamentos teóricos y fases de diseño e implementación de tecnologías web para la autenticación de usuarios, a manera de desarrollar un componente para un aplicativo en el entorno node.js que permita al usuario final usar de manera intuitiva y segura sus datos en alguna de las plataformas de red social estudiadas.**

Palabras Clave—accesibilidad, endpoints, autenticación, usuario final, redes sociales.

I. INTRODUCCION

A. *Descripción*

El presente documento consiste en una investigación realizada como parte del desarrollo de un componente para la autenticación de usuarios en línea con plataformas de red social como *Facebook*, *Google*, *Instagram* y *Twitter*, basado en el entorno de ejecución *Node.js* para el lenguaje de programación *JavaScript*.

B. *Justificación*

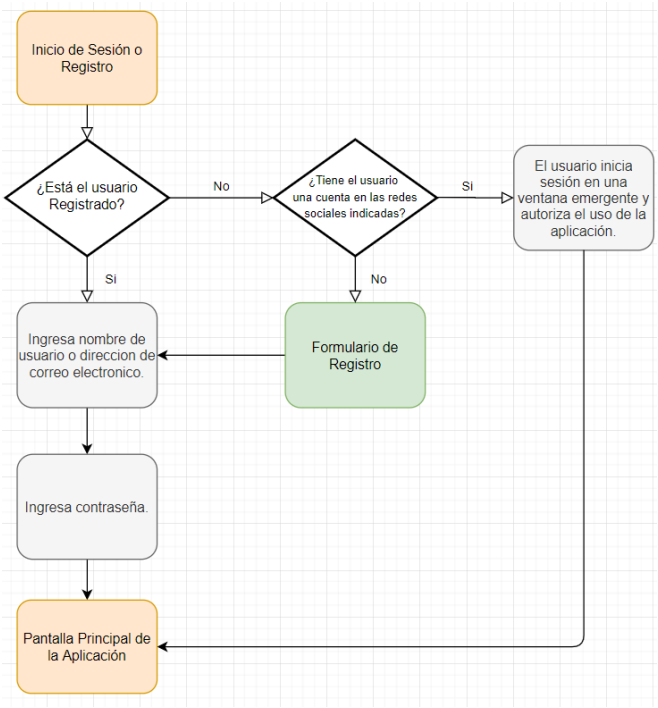
Se requiere un componente que permita la autenticación ágil y segura al usuario final por medio de datos personales que se encuentren previamente registrados en diversas redes sociales, facilitando el uso de la aplicación que se complementaría al integrarla con algunas de las plataformas más relevantes y evitando hacer pasar al usuario por formularios de registro innecesarios.

II. EXPERIENCIA DE USUARIO

A. *Historias de Usuario*

Como usuario final de la aplicación quiero hacer uso de mis datos personales existentes en la red social Facebook, autenticándome así como usuario único para acceder a la herramienta fácilmente.

B. *Diagrama de Flujo*



III. SERVIDOR EXPRESS NODE.JS

Node.js es un programa para servidores construido a partir del motor JavaScript V8 de *Google Chrome*, un compilador desarrollado para exploradores de internet basados en *Chromium* que procesa código Javascript durante su ejecución (en tiempo real o "real-time"). Su primera versión fue creada en el 2009 y su definición según la documentación oficial es la de un programa para la implementación rápida y escalable de aplicaciones de red (Node.js Org, 2020). Es descrito como una plataforma "asíncrona y orientada a objetos, de paradigma I/0 no bloqueante" (Code Academy, s.f).

A. *Justificación de uso y ventajas.*

De manera resumida esto significa que Node.js evita asignar recursos por separado a cada petición que procese una aplicación web, y en su lugar asigna todas las peticiones a un solo subproceso, en el cual por medio de la tecnología de Eventos de Node, se atienden llamadas a bases de datos o APIs en colas secundarias (asíncronamente) para que el código JavaScript nunca deje de ejecutarse (no se bloquean tareas al no esperar que cada una termine para iniciar la siguiente). Es por esto que un

servidor de aplicación basado en Node.js no espera a que el API de un servicio (ej. Inicio de Sesión con Facebook) retorne datos. Este simplemente continúa hacia la siguiente petición (ej. otro usuario obteniendo sus datos de Instagram) luego de llamar al primero y recibe una notificación gracias a los mecanismos de eventos en Node.js que le permite al servidor obtener la respuesta de las llamadas a APIs anteriores. Al nivel de hardware, esto maximiza el uso de un único CPU así como la memoria del sistema, permitiéndole manejar decenas de miles de conexiones simultáneamente y con alto rendimiento (Capan T, 2018).

B. NPM Express

La herramienta NPM del acrónimo "Node Package Manager" facilita la gestión de módulos de Node, siendo estos conjuntos de código reutilizable a disposición pública y gratuita que agiliza el despliegue de un nuevo servidor. Por su lado express es un framework o paquete de desarrollo rápido y minimalista para construir un servidor de node.js, y es reportado que compañías como IBM, UBER y PayPal lo usan para sus aplicaciones (OpenJS Foundation, s.f).

```
npm install express --save
```

Una vez creado el proyecto de node.js, ingresando al directorio principal se puede instalar el framework express con este comando, --save añade el paquete express al archivo de package.json.

IV. Creacion de Certificados y Activación HTTPS

A. SSL, TLS y HTTPS

Ejemplo de sitio web no seguro. (Cloudfare, s.f.)

El protocolo Secure Sockets Layer (oficialmente Transport Layer Security a partir de 1999) se utiliza para establecer enlaces autenticados y encriptados entre redes, y funciona por medio de documentos digitales que almacenan pares de llaves criptográficas. Una es la llave privada, un archivo de texto (código) confidencial almacenado en el servidor que protege y verifica las conexiones al mismo. Por su parte la llave pública se incorpora en el certificado SSL que procesan los navegadores web, y permite decodificar la llave privada y así establecer una conexión segura. Esta es la diferencia entre el protocolo HTTP (Hyper Text Transfer Protocol) y HTTP Secure (Digicert, s.f).

El desarrollar con APIs de servicios como la red social Facebook requiere un servidor con los certificados adecuados para evitar que los tokens de acceso de los usuarios sean vulnerables al robo de datos, por lo que es necesario comprar un certificado para el sitio web en desarrollo o generar un ambiente de pruebas local que tenga un certificado instalado.

"El protocolo HTTPS ofrece mayor seguridad para los sitios web como el Inicio de sesión con Facebook. Al cifrar las comunicaciones, se protege la privacidad y la integridad de la información que se intercambia." (Facebook, s.f).

B. Certificado Autofirmado

Existen varios métodos para obtener el protocolo seguro HTTPS en una aplicación, la documentación para desarrolladores de Facebook recomienda servicios como letsencrypt.org, que ofrece certificados gratuitos a sitios web que cumplan con ciertos parámetros. El siguiente es un certificado auto firmado que se genera e instala localmente, funcionando como una autoridad de certificación limitada al equipo en cuestión. En primer lugar se debe instalar la librería criptográfica OpenSSL desde https://slproweb.com/products/Win32OpenSSL.html, o se verifica si se está ya instalada con el comando openssl en la terminal. Una vez iniciado OpenSSL en el directorio del proyecto, se ejecuta lo siguiente para generar las llaves:.

```
openssl req -nodes -new -x509 -keyout server.key -out server.cert
```

La terminal generará un pequeño formulario. Los campos de país, provincia, ciudad y organización pueden ser ignorados. Es importante ingresar "localhost" en Common Name y un correo electrónico seguro.

State or Province Name (full name) [Some-State]:

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []: localhost

Email Address []: micorreo@correo.com

El resultado son los siguientes dos archivos:

server.cert = El certificado autofirmado.

server.key = La llave privada del certificado.

A continuación un ejemplo de una llave privada y pública, generadas a partir de cadenas aleatorias:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAm+036O2PlUQbKbSSs2ik
6O6TYy6+Zsas5oAk3GioGLl1RW9Ni8kagqdnD69Et29m1vl5OIPsBoW3OWb1aBW5
e3J0x9prXI1W/fpvuP9NmrHBUN4ES17VliRpfVH3aHfPC8rKpv3GvHYOcfOmMN+H
fBZlUeKJKs6c5WmSVdnZB0R4UAWuQ30aHEBVqtrhgHqYDBokVe0/H4wmwZEIQTIN
WniCOFR5UphJf5nP8ljGbmPxNTnfb/iHS/chjcjF7TGMG36e7EBoQijZEUQs5IBC
eVefOnFLK5jLx+BC//X+FNzByDilTt+l28I/3ZN1ujhak73YFbWjjLR2tjtp+LQg
NQIDAQAB
-----END PUBLIC KEY-----
```

C. Configuracion Final en Node Express Server y Explorador de Internet

- Una vez se tienen los documentos de llave privada y pública, estos se almacenan en un directorio dentro del proyecto.
- En el caso de Google Chrome, se dirige a la configuración avanzada y en la sección de Seguridad se accede a las opciones para certificados de seguridad.

- Una vez cargado o instalado el certificado privado, que sirve como sustituto “local” de una autorización real por parte de una autoridad de certificación, será posible realizar pruebas en el explorador consiguiendo el protocolo HTTPS durante la ejecución del proyecto.

```
const https = require('https')
const app = express()

app.get('/', (req, res) => {
  res.send('Hello HTTPS!')
})

https.createServer({}, app).listen(3000, () => {
  console.log('Listening...')
})
```

En la clase index.js se ha configurado el proyecto para requerir el protocolo https y escuchar al Puerto 3000 en el servidor local, finalizando las preparaciones para que las solicitudes a los distintos apis de redes sociales sean aceptadas.

V. SOBRE NODE JS

A. *Ideal, diseño:*

Construido con el motor de JavaScript V8 de Chrome, Node.js es un entorno de ejecución de JavaScript idealizado en los eventos asíncronos, su diseño cae en la creación de aplicaciones Network escalables. De diferentes formas pueden atenderse varias conexiones al mismo tiempo, y al mismo tiempo por cada de estas conexiones, se activa una devolución. Sin embargo, Node.js se dormirá si no hay trabajo.

Node.js contraste el modelo más común de concurrencia, el que utiliza hilos del sistema operativo. Ya que las redes basadas en hilos en gran parte son ineficientes y difíciles de usar. Además, Node.js está libres de preocupación por el bloqueo del proceso, ya que no existe. Casi ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea.

B. *Similitudes, diferencias e Influencia:*

- Sistemas como Event Machine (Ruby) o Twisted (Python).
- Sin embargo, Node.js lleva el modelo un poco más allá. Incluye bucles de eventos como ejecución en lugar y no una biblioteca y siempre existe una llamada de bloqueo para empezar este bucle de eventos.
- Node.js, no cuenta como tal con la llamada de inicio del evento de bucle o start-the-event-loop.
- Node.js, entra en el bucle de eventos después de ejecutar el script de entrada y sale cuando no hay más devoluciones de llamada para realizar. Se comporta de una forma similar a JavaScript en el navegador - el bucle de eventos está oculto al usuario.
- El elemento HTTP destaca en Node.js, ya que está diseñado tomando en cuenta la transmisión de operaciones con énfasis en streaming y la baja latencia. Lo cual hace que Node.js sea muy adecuado, por ejemplo, en una librería web.

“Que Node.js esté diseñado para trabajar sin hilos no significa que no pueda aprovechar múltiples núcleos en su entorno. Se pueden generar subprocesos o procesos hijos utilizando nuestra API, la cual está diseñada para que la comunicación entre ellos sea fácil mediante su proceso principal. Desarrollada sobre esa misma interfaz está el módulo clúster, que le permite compartir sockets entre procesos para permitir el balanceo de carga entre sus múltiples núcleos.” (OpenJS, 2020).

C. *Comunidad:*

El Comité de la Comunidad es un comité en la Fundación Node.js. El CommComm tiene autoridad sobre los esfuerzos hacia el exterior que se realicen frente a la comunidad, que incluyen:

- Comunidad de Evangelistas
- Iniciativas Educativas
- Dirección Cultural de la Fundación Node.js
- Alcance de la Organización de la Comunidad
- Traducción e Internacionalización
- Proyecto de Moderación/Mediación
- Alcance Público y Publicaciones

D. Express.js:

Express es la aplicación de Node.js más popular, se utiliza para más de 3000 redes en diferentes servidores, esta aplicación responde al comando Hello World! y provee mecanismos para:

- Escribir controladores para solicitudes con diferentes verbos Usar HTTP en diferentes rutas de URL (rutas).
- Integrar motores de renderizado de "vista" para generar Responder insertando datos en plantillas.
- Establecer la configuración común de la aplicación web, como el puerto que se utilizará para conectarse y la ubicación de las plantillas que se utilizan para representar la respuesta.
- Agregar "middleware" de procesamiento de solicitudes adicional en cualquier punto dentro de la canalización de manejo de solicitudes.

VI. EL API DE TWITTER

Twitter es lo que está pasando y es de lo que las personas están hablando en la actualidad. Twitter se puede acceder por medio de la web y dispositivos móviles. Twitter por medio de su API logra compartir información lo más ampliamente posible a empresas, desarrolladores y usuarios.

En un nivel muy alto, Twitter da el acceso a múltiples servicios por medio del API para permitir que las personas

puedan desarrollar software que se integre con Twitter, para brindar soluciones a diferentes empresas para responder a sus clientes.

A. Acceso a la data de Twitter

Cuando algún usuario desea acceder a la API de Twitter, necesita registrarse a la página de desarrolladores que es Twitter Developers para crear una aplicación y tener acceso a las funcionalidades. Dichas aplicaciones solo tienen acceso a la información pública de Twitter. Algunos endpoints, como los responsables de recibir y enviar mensajes directos, necesitan permisos adicionales de los usuarios antes de que pueda obtener la información necesaria.

Dichos permisos no se otorgan de forma predeterminada; el usuario elige por aplicación si desea proporcionar acceso y poder controlar las aplicaciones que están autorizadas en su cuenta.

El API de Twitter incluye una larga variedad de endpoints, que caen en cinco grupos:

a) Cuentas y usuarios: Permiten a los desarrolladores administrar mediante programación el perfil y la configuración de una cuenta, silenciar o bloquear usuarios, administrar usuarios y seguidores, solicitar datos sobre la actividad de una cuenta que este autorizada y muchos más.

b) Tweets y respuestas: Se pone a disposicion de los desarroladores los tweets y respuestas publicos, para permitir a los desarrolladores public tweets a traves de la API de Twitter. Los desarrolladorespueden acceder a los Tweets por medio de palabras claves especificas o solicitando una muestra de cuentas especificas. ONG como la ONU utilizan estos puntos finales para identificar, comprender y contrarrestar la información errónea en torno a las iniciativas de salud pública

c) Mensajes directos: Las terminales de Twitter de mensajes directos proporcionan el acceso a las conversacion privadas de aquellos usuarios que otorgan permiso explicitamente a una aplicación en especifico. Los mensajes directros no se venden. La API de mensajes directos brinda acceso limitado a los desarrolladores para crear experiencias personalizadas en Twitter. Para cuentas que poseen o administran, las empresas tienen la posibilidad de producir experiencias de conversación que son impulsadas por humanos o chatbot para comunicarse directamente con los clientes para la experiencia que se brinda a la atencion al cliente, marketing o participacion de marca.

d) Anuncios: Tambien proporcionan un conjunto de APIspara que lo desarolladores ayuden a las empresas a crear y administrar automaticamente campañas de publicidad en Twitter. Los desarroladores pueden utilizar Tweets publicos para encontrar temas e interes y brindar a las empresas herramientas para producir campañas publicitarias para llegar a diversas audiencias en Twitter.

e) Herramientas de editor y SDK: El API bi¡rinda herramientas para que los desarrolladores y editores de software incorporen lineas de tiempod e Twitter, botones para compartir y otro tipo de contenido de Twitter en paginas web. Dichas herramientas permiten a las marcas llevar conversaciones públicas en vivo desde Twitter a la experiencia wewb y facilitar a los clientes compartir información y articulos de sus sitios.

VII. PASSPORT.JS

Passport es un middleware de autenticación para Node.js. Es extremadamente flexible y modular, Passport puede ser incorporado discretamente en cualquier aplicación web basada en Express. Es un conjunto de estrategias que admite la autenticación mediante un nombre de usuario y una contraseña o por medio de acceso de redes sociales como Facebook, Google o Twitter.

La autenticación de pasaportes se basa en un servicio centralizado proporcionado por Microsoft. La autenticación de Passport identifica a un usuario con el uso de su dirección de correo electrónico y una contraseña, y se puede usar una sola cuenta de Passport con muchos sitios web diferentes. La autenticación de pasaporte se usa principalmente para sitios web públicos con miles de usuarios.

Algunas de las características de Passport son:

- Más de 300 estrategias de autenticación.
- Inicio de sesión con OpenID y OAuth.
- Maneja fácilmente el éxito y el fracaso.
- Admite sesiones persistentes. Permisos y alcance dinámico.
- Implementar estrategias personalizadas. Y elegir las estrategias necesarias.

VIII. CONCLUSIONES

La versatilidad del entorno node.js junto a la facilidad de uso de los métodos API investigados nos han permitido apreciar la importancia de proveer soluciones seguras y convenientes al usuario final, aprovechando las mejores prácticas y tendencias en el uso de internet para ofrecer un mejor producto. La autenticación de usuarios así como otras funcionalidades ofrecidas por los servicios web de las redes sociales anteriormente discutidas representan una poderosa herramienta en el diseño de la experiencia de usuario y la legitimización de los aplicativos que la empleen, así como una importante alternativa de accesibilidad que es hoy un estándar en el manejo de cuentas en línea.

IX. REFERENCIAS

[1] Code Academy. (s.f). What is Node? Sitio web: https://www.codecademy.com/articles/what-is-node

[2] Capan, T. (2018). ¿Por qué demonios usaría Node.js? Un tutorial caso por caso. Sitio web: https://www.toptal.com/nodejs/por-que-demonios-usaria-node-js-un-tutorial-caso-por-caso

[3] DigiCert. (s.f). What is SSL, TLS and HTTPS?. julio 8, 2020, de DigiCert Sitio web: https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https

[4] ExpressJS. (s.f). Fast, unopinionated, minimalist web framework for Node.js. Sitio web: http://expressjs.com/

[5] Facebook. (s.f). Activar HTTPS. julio 8, 2020, de FACEBOOK for Developers Sitio web: https://developers.facebook.com/docs/facebook-login/web/enabling-https

[6] NodeJS Org. (s.f). About Node.js. Sitio Web: https://nodejs.org/en/about/

[7] OpenJS Foundation. (2020). About Node.js®. 20/08/20, de Node.js Sitio web: https://nodejs.org/en/about/

[8] OpenSSL. (s.f). Welcome to OpenSSL!. julio 8, 2020, de OpenSSL Sitio web: https://www.openssl.org/

[9] Passport.js. (s.f). Passport. agosto 21, 2020, de Passport.js Sitio web: http://www.passportjs.org/

[10] Passport.js. (s.f). Features. agosto 21, 2020, de Passport.js Sitio web: http://www.passportjs.org/features/

[11] Twitter. (s.f). About Twitter's APIs. agosto 21, 2020, de Twitter Sitio web: https://help.twitter.com/en/rules-and-policies/twitter-api