

# Comprensión de protocolos de red desde una perspectiva de programación

Wendy Quirós Ramírez y Elías José Salazar Zeledón

Escuela de Ingeniería Informática,  
Universidad Latinoamericana de Ciencia y Tecnología,  
ULACIT, Urbanización Tournón, 10235-1000  
San José, Costa Rica  
[wquirosr086, esalazarz859]@ulacit.ed.cr  
<http://www.ulacit.ac.cr>

**Resumen** En la actualidad, la enseñanza en la mayoría de los cursos de redes de computadoras se basa en conceptos teóricos, explicaciones magistrales y lecturas de material bibliográfico; esto tiene como consecuencia que no se tomen en cuenta los aspectos prácticos de la programación de protocolos como parte de los contenidos de dichos cursos. De acuerdo con lo anterior, esta investigación busca apoyar el desarrollo de un enfoque que sirva para aplicar los conceptos y protocolos vistos en las clases, los cuales faciliten su comprensión por medio de la descripción detallada de su programación.

**Keywords:** Protocolo de red, redes de computadoras, enseñanza de redes de computadoras, programación e Internet

## 1. Introducción

En el contexto actual, las personas y las organizaciones llevan a cabo actividades que giran en torno a la tecnología y las telecomunicaciones, con el uso de una gran variedad de dispositivos, en su mayoría móviles. Dichos dispositivos utilizan protocolos, que originalmente fueron creados para conectar computadoras (i.e. en su acepción original), con el fin de formar redes de dispositivos.

En línea con lo anterior, el desarrollo de sistemas para comunicaciones requiere comprender el funcionamiento de los principales protocolos que son comunes en las redes de dispositivos. Sin embargo, algunos de los libros más conocidos en el área de redes de computadoras, aunque explican muy bien los conceptos teóricos sobre los protocolos de comunicación más comunes, no ofrecen detalles sobre su programación. Como consecuencia, el presente artículo busca apoyar a los estudiantes de carreras universitarias en el proceso de comprensión de los protocolos de comunicación más conocidos y utilizados, tanto en la academia como en la industria.

Con ese fin, el presente trabajo realiza la descripción detallada de un grupo de demostraciones de programación de aplicaciones y servidores con HTML5,

JavaScript, Node.js y Express para realizar conversaciones escritas, videollamadas, envío de correo electrónico y transferencia de archivos al usar los protocolos HTTP<sup>1</sup>, HTTPS<sup>2</sup> y SMTP<sup>3</sup>. De forma adicional a las demostraciones que se incluyen en este documento, el trabajo incluye anexos que detallan la guía de instalación y configuración del ambiente de desarrollo y las demostraciones.

En términos generales, el proceso para mejorar la enseñanza, requiere comprender las formas de aprendizaje de los alumnos; a partir de lo que aprenden y cómo lo aprenden. En consecuencia, este trabajo se relaciona con el proceso de enseñanza y la fenomenografía (Ramsden, 1992).

## 2. Trabajos relacionados

El aprendizaje de protocolos de red es un proceso que cada individuo realiza de forma distinta mediante la comprensión de conceptos generales, definiciones y términos. Sobre este particular, Berglund et al. identificaron cuatro formas de cómo las personas comprenden los conceptos relacionados con los protocolos de red (Berglund y Lister, 2007).

La identificación de esas cuatro formas de comprensión es el resultado de una investigación basada en la fenomenografía<sup>4</sup> sobre la manera en que los estudiantes entienden los protocolos de red.

El estudio realizado por Berglund et al. consiste en una serie de entrevistas sobre protocolos de red a 15 estudiantes universitarios de la carrera de Informática de tercer y cuarto año, procedentes de Estados Unidos y Suecia. Dichos estudiantes estaban matriculados en un curso, cuya evaluación estaba basada en un proyecto para programar, con equipos de trabajo, el control de un juguete motorizado. Cabe mencionar que como requisito indispensable, este tipo de proyecto requiere tener buenas habilidades para el desarrollo de software y protocolos de comunicación.

El análisis de los resultados de las entrevistas permitió observar las siguientes definiciones de protocolo de red entre los alumnos:

1. Un protocolo de red es una forma de establecer comunicación entre dos computadoras.
2. El protocolo de red es un método de comunicación mediante Internet.
3. El protocolo de red como un conjunto de reglas.
4. Un protocolo de red es un estándar.

En general, las definiciones identificadas se relacionaban con el aprendizaje de terminología por medio de la lectura de libros o explicaciones de profesores, y no a partir de la experimentación o trabajo práctico.

<sup>1</sup> Siglas en inglés para Hypertext Transfer Protocol.

<sup>2</sup> Siglas en inglés para Hypertext Transfer Protocol Secure.

<sup>3</sup> Siglas en inglés para Simple Mail Transfer Protocol.

<sup>4</sup> La fenomenografía se basa en las diferentes formas de percibir la información cuando los individuos leen un texto, escuchan una exposición o tratan de resolver problemas, porque describe los fenómenos del mundo como otros los ven y muestra las diferentes maneras que son vistos, por lo general en un contexto educativo (Santos, 2006).

De acuerdo con la fenomenografía del aprendizaje, las definiciones obtenidas de las entrevistas se encuentran relacionadas entre sí, al ser mencionadas y limitadas a dos equipos en la primera definición, lo cual indica que una red es una forma de comunicar dos computadoras. Sin embargo, la segunda y tercera definición contemplan un conjunto de equipos, lo cual abarca una red de computadoras. La cuarta definición considera a los equipos, las redes y a los seres humanos, que son los encargados de establecer los estándares. La tabla 1 muestra un resumen de dichas definiciones y su relación con la percepción y elementos (Berglund y Lister, 2007).

Concepto de protocolo de red	Percepción de los alumnos	Elemento
Un protocolo de red es una forma de comunicar dos computadoras	Paquetes y dos computadoras	Dos computadoras
El protocolo de red es un método de comunicación mediante Internet	Comunicación extremo a extremo	Red de computadoras
Protocolo de red como un conjunto de reglas	Reglas que regulan la comunicación	Red de computadoras
Un protocolo de red es un estándar	Conceptos y definiciones	Computadoras, redes de computadoras y seres humanos

**Cuadro 1.** Concepto general de protocolo de red.

El trabajo de investigación realizado por Salleh et al. (Salleh, Winslett, Bruce, y Tickle, 2005) clasificó la comprensión que tienen los estudiantes del concepto de protocolo de red en dos categorías:

1. Dos o más entidades de comunicación.
2. Un sistema de comunicación de dispositivos interconectados con protocolos asociados (i.e. engloba protocolos, estándares, reglas y redes de computadoras).

Las categorías de esta investigación se encuentran relacionadas con las categorías encontradas en la investigación de Berglund et al.:

- La primera categoría de ambos trabajos indican que los protocolos de red son un medio de comunicación entre equipos.
- La segunda categoría de Salleh, et al. se relaciona con la segunda, tercera y cuarta categoría de la investigación de Berglund, et al.

La importancia sobre la comprensión de los protocolos de red, abarca distintas áreas, que dependen del contexto en el que esté el estudiante; así va a tener que aplicar los conceptos enfocados en el campo en que se encuentra. A continuación se detallan las diferentes maneras en que un protocolo de red puede ser relevante y hacia quién se encuentra dirigido, según el estudio realizado por Berglund, et al.:

1. Relevancia para programadores: Cuando un programador escribe código para una aplicación que por ejemplo ofrece servicios TCP<sup>5</sup>, como creación de conexiones y envío de datos; debe conocer cómo es el funcionamiento del protocolo y su concepto, para aplicarlo en sus desarrollos, por lo que sí es relevante cuando se programa.
2. Relevancia para un diseñador de programas: Al realizar el diseño y análisis de un programa, se deben considerar las propiedades y características del protocolo que se va a utilizar y bajo qué circunstancias y cómo se va a utilizar en la aplicación; por tanto esto es importante ya que si se tiene el concepto claro, se facilita la elección del protocolo durante la fase de diseño del programa.
3. Relevancia para el desarrollo técnico: Este apartado se asemeja al anterior, solo que la diferencia clave se encuentra en que si se aplica un protocolo ya existente o si es el desarrollo de un protocolo a la medida, para lo cual se pueden emplear uno o más protocolos ya existentes al combinar sus funcionalidades y características.
4. Relevancia para las políticas: En la categoría 4 del estudio de Berglund, et al., se menciona el protocolo de red como un estándar, que se refiere a cómo se lleva a cabo el desarrollo de un protocolo y a sus propiedades, para lo cual se efectúa un análisis detallado y discusiones entre las partes involucradas; esto es relevante cuando se requiera conocer acerca de las normas y reglas que rigen la creación de un protocolo.

En relación con los resultados obtenidos sobre la importancia de la comprensión de los protocolos de red descritos en la sección anterior; existe un estudio efectuado por Booth, que identifica tres categorías relevantes en la programación en un grupo de estudiantes principiantes de ingeniería, las cuales son en cierta parte similares a lo expuesto por Berglund et, al. Las categorías son las siguientes: orientación del producto, orientación del problema, y orientación del ordenador; todas las categorías son importantes durante las fases de diseño de software, creación de prototipos y programación respectivamente (Booth, 1992). Se podrían desarrollar las categorías de los dos estudios y comparar sus características, para futuros trabajos.

El currículum para las carreras de informática de la IEEE<sup>6</sup> y ACM<sup>7</sup> señala que los fundamentos de la informática consisten en un conjunto diverso de disciplinas (The Joint Task Force on Computing Curricula, 2001). Asimismo señala que los estudiantes deben integrar la teoría y la práctica, reconocer la importancia de la abstracción y apreciar el valor de un buen diseño de ingeniería.

De forma particular, dicho documento argumenta para el área de redes de computadoras, que el dominio de esta, implica llevar la teoría a la práctica. Por tanto, un profesional en informática debe estar familiarizado con los diferentes

<sup>5</sup> Siglas para Transmission Control Protocol, es uno de los protocolos fundamentales de Internet.

<sup>6</sup> Siglas en inglés para Institute of Electrical and Electronics Engineers.

<sup>7</sup> Siglas en inglés para Association for Computing Machinery

aspectos relacionados con las redes de computadoras (The Joint Task Force on Computing Curricula, 2001).

El informe hace hincapié en la necesidad de comprender un sistema en una forma más avanzada que simplemente verlo como un programa, señala que los graduados de una carrera en informática deben desarrollar una comprensión de alto nivel, la cual trascienda los sistemas y los detalles de la programación de los diversos componentes que abarcan la estructura de los sistemas informáticos y los procesos implicados en su análisis y desarrollo (The Joint Task Force on Computing Curricula, 2001).

### 3. Resultados

En esta sección se presenta una selección de los ejemplos realizados para este trabajo de investigación; el resto de ejemplos se incluyen en un documento anexo a este artículo. Los ejemplos, en general, fueron desarrollados utilizando tecnología HTML5, Bootstrap, JQuery y Node.js, y se ejecutan en los navegadores Mozilla Firefox y Google Chrome, debido a que brindan soporte para HTML5. La lista de todos los ejemplos que fueron programados se detallan a continuación:

1. Servidor HTTP utilizando Node.js.
2. Servidor HTTP utilizando Node.js y Express.
3. Servidor HTTPS utilizando Node.js y Express.
4. Subir archivos al servidor utilizando Node.js y Express.
5. Streaming de video utilizando Node.js
6. Chat utilizando Node.js, Express y Socket.io.
7. Envío de correo electrónico utilizando Node.js
8. Chat con videollamada utilizando Node.js y Socket.io

De la lista anterior fueron seleccionados los ejemplos correspondientes a HTTPS (ejemplo 3) y envío de correo electrónico (ejemplo 7), debido a las características de sus funciones y su uso masivo en la actualidad.

#### 3.1. Servidor HTTPS

¿Qué es el protocolo HTTPS y qué elementos o componentes se requieren para su funcionamiento?

El protocolo HTTPS es el protocolo para transferencia segura de hipertexto que hace uso del protocolo TCP y el puerto 443 (puerto común<sup>8</sup>). Este protocolo utiliza SSL<sup>9</sup> y TLS<sup>10</sup>, y requiere del uso de un certificado digital, el cual es un documento que sirve para verificar la identidad del usuario. Los certificados digitales se encuentran firmados por una entidad emisora de certificados, y utilizan

<sup>8</sup> La acepción en inglés para este tipo de puertos es “well-known ports”.

<sup>9</sup> SSL se corresponden con las siglas Secure Socket Layer.

<sup>10</sup> TLS son las siglas que se corresponden con Transport Layer Security.

una infraestructura PKI <sup>11</sup>. El certificado digital cuenta, para su funcionamiento con una llave pública y una llave privada; la primera descripta la firma asociada y la segunda encripta la información respectivamente. El estándar más utilizado para certificados digitales es el X509 (Ince, 2001). El certificado digital empleado para el ejemplo en esta sección, se generó al utilizar la herramienta de software libre OpenSSL, la cual contiene bibliotecas para el manejo criptográfico y provee implementaciones de los algoritmos de llave pública más utilizados en la industria como AES <sup>12</sup> y RSA <sup>13</sup> (John Viega, 2002).

En el ejemplo que se muestra en la figura 1, se detalla la programación del servidor HTTPS, que inicia con la declaración de las variables y componentes Node.js (líneas 7-11); para ello se declara la variable con el número de puerto que utilizará el servidor (línea 7). En la sección de la declaración de los componentes Node.js se realiza la definición de la variable que contiene la biblioteca<sup>14</sup> fs; la cual es la encargada de interactuar con el sistema de archivos del equipo (línea 8). Adicionalmente, se define el uso de la biblioteca https, que tendrá a cargo el manejo del servidor (línea 9) y la biblioteca Express; dicha biblioteca establece las rutas de los directorios del servidor (línea 10), y que se complementa con la instancia de su variable (línea 11). Para configurar el certificado digital (líneas 14-20), se define la variable que contiene la llave privada del certificado; se utiliza la biblioteca fs para leer de la ruta especificada el archivo en formato .pem <sup>15</sup> (línea 14). De seguido, se declara la variable que contiene el archivo del certificado digital; para esto se utiliza la biblioteca fs igual que en el paso anterior, para leer de la ruta especificada el archivo en formato .pem con el certificado (línea 17). En la línea 20 se define un arreglo que contiene las variables de la llave privada y del certificado, las cuales se utilizarán para crear el servidor HTTPS.

Para la configuración de Express (líneas 23-28), se definen las funciones propias de la biblioteca, que sirven para configurar los directorios del servidor. En la línea 23 se declara la función set, la cual establece la ruta llamada views, que redirecciona hacia el directorio indicado; cabe destacar que la cláusula dirname indica el directorio donde se ejecuta la aplicación. La otra función de Express que se utiliza en el ejemplo es la función get; la misma se ejecuta cuando se accede al servidor mediante una petición HTTP GET (línea 26), al llevar a cabo la petición, se redirecciona hacia el archivo index.html que se encuentra dentro del directorio de views (línea 28). La creación del servidor HTTPS (líneas 32-38) se realiza con la función *createServer*, que recibe como parámetro las credenciales de llave privada y del certificado digital, junto con el objeto Express (línea 32). Cuando el servidor es creado, se muestra un mensaje en la consola del servidor que indica su creación (línea 35) y se establece el puerto mediante el cual el servidor escuchará peticiones (línea 38).

<sup>11</sup> PKI corresponde a las siglas Public Key Infrastructure.

<sup>12</sup> AES corresponde a las siglas de Advanced Encryption Standard.

<sup>13</sup> RSA son las siglas del algoritmo Rivest, Shamir y Adleman.

<sup>14</sup> Corresponde a un conjunto de funcionalidades de programación.

<sup>15</sup> PEM son las siglas de Privacy-enhanced Electronic Mail, es un formato de archivo de certificado codificado en Base64.

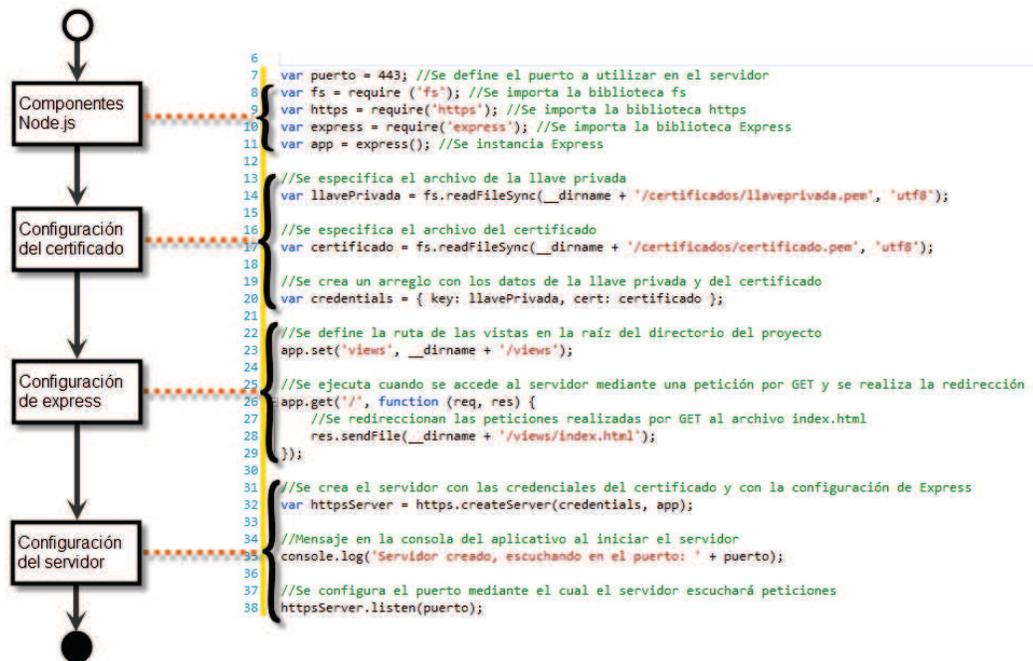


Figura 1. Código fuente de servidor HTTPS desarrollado en Node.js con Express

### 3.2. Envío de correo electrónico

#### ¿Qué se necesita para utilizar el correo electrónico?

Para usar el correo electrónico se requiere tener conexión a Internet en el equipo desde donde se va a enviar el correo, para ello se debe tener una suscripción con un ente ISP<sup>16</sup>. También, se necesita un programa conocido como cliente de correo electrónico<sup>17</sup> o un servicio de envío de correo electrónico basado en la Web como Gmail, Yahoo! o Hotmail; para llevar a cabo la configuración de un cliente de correo electrónico se debe tener información sobre la cuenta de correo electrónico, la contraseña, los servidores de correo electrónico entrante y saliente, y otros detalles. (Microsoft, s.f.)

En el ejemplo que se detalla en las figuras 2, 3 y 4 se explica la programación de un cliente de correo electrónico al que se le configura una cuenta de Gmail, para que pueda enviar mensajes de correo electrónico utilizando Node.js. Se inicia con la declaración de los componentes de Node.js (líneas 8-11), para ello se define la variable del puerto que utilizará el servidor (línea 8), de seguido se realiza la definición de la variable que contiene la biblioteca nodemailer (línea 9), que contiene funciones para configurar la cuenta de correo electrónico con los datos del tipo de servicio, dirección y contraseña; además realiza el envío del correo. Adicionalmente, se define la variable que contiene la biblioteca Express, la cual establece las rutas de los directorios del servidor (línea 10); a continuación se realiza la instancia de la misma (línea 11).

En la sección de configuración de correo electrónico se crea un objeto del tipo `Transporter` mediante la función `createTransport` (línea 14), la cual recibe como parámetros el tipo de servicio, en este caso se especifica Gmail (línea 16), datos para la autenticación (línea 19), que contendrá el usuario de la cuenta de correo electrónico (línea 21) y la contraseña de la cuenta (línea 24).

Para la configuración de Express (líneas 28-31), se realiza la definición de funciones propias de la biblioteca, que sirven para configurar los directorios del servidor. En la línea 29 se declara la función `get` que se ejecuta cuando se accede al servidor mediante una petición HTTP GET; cuando se realiza la petición se redirecciona hacia el archivo `index.html` que se encuentra dentro en la raíz del directorio del proyecto (línea 31).

En la sección de envío del correo electrónico (líneas 36-77) se define la función Express `get`; la cual se ejecuta cuando se accede al servidor mediante una petición GET con el valor `send` (línea 36). Se declara la variable `mailOptions`, la cual contiene los atributos del mensaje que se va a enviar por correo electrónico (línea 38).

<sup>16</sup> Siglas en inglés de Internet Service Provider, traducido como Proveedor de Servicios de Internet.

<sup>17</sup> Es un programa usado en una computadora o dispositivo portátil, para acceder y administrar una cuenta de correo.

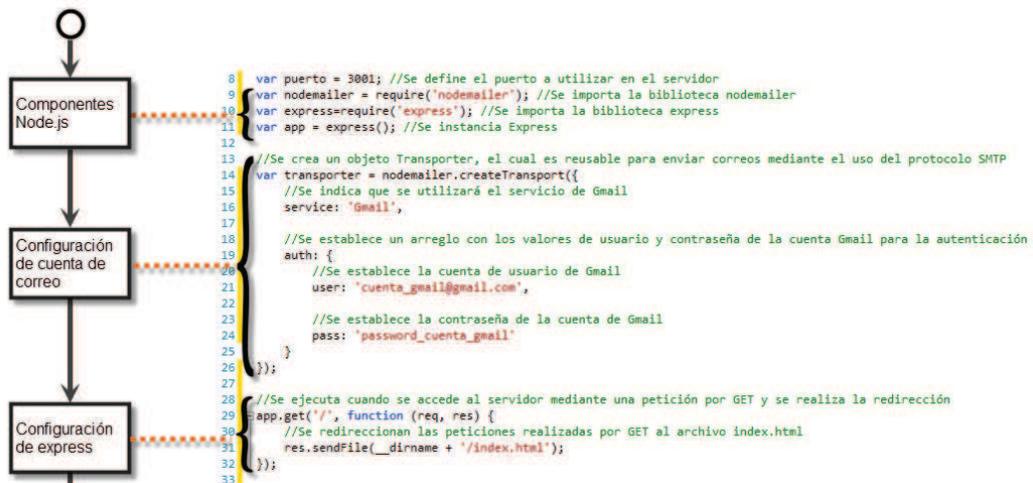


Figura 2. Código fuente para envío de correo electrónico utilizando Node.js con Express Parte 1

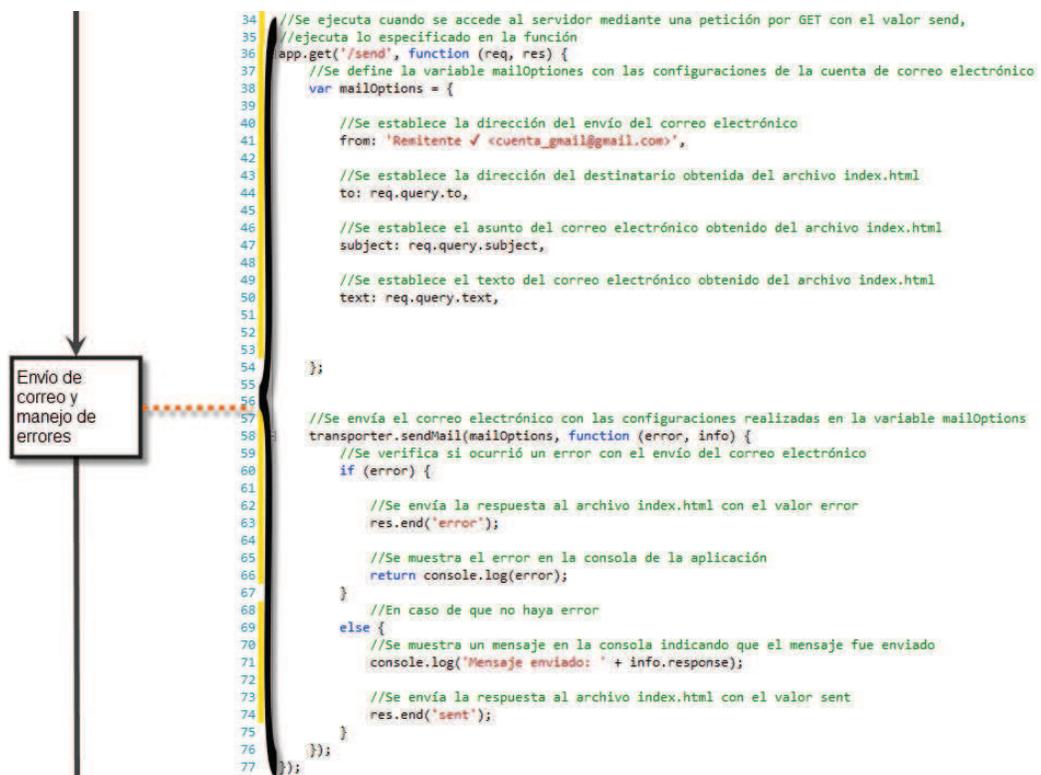


Figura 3. Código fuente para envío de correo electrónico utilizando Node.js con Express Parte 2

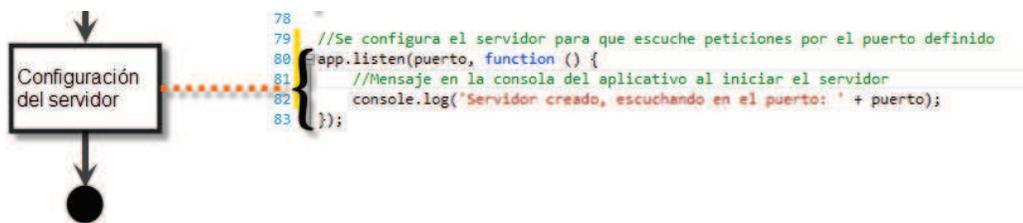


Figura 4. Código fuente para envío de correo electrónico utilizando Node.js con Express Parte 3

Existe un archivo html, que contiene un formulario para efectuar el envío del mensaje de correo electrónico; de dicho archivo se obtienen los siguientes atributos: la dirección de correo electrónico del remitente (línea 41), la dirección de correo electrónico del destinatario (línea 44), el asunto del correo electrónico (línea 47), el contenido del mensaje (línea 50) y se establece un contenido html en la línea 53, el cual se incluye dentro del mensaje. El detalle del código fuente del archivo html se especifica en la guía que se encuentra anexa al presente artículo.

Para el envío del correo electrónico se define la función *sendMail*, la cual recibe como parámetros la variable *mailOptions* que posee la configuración de la cuenta de correo electrónico y una función que tiene parámetros de indicadores para saber si hubo un error durante el envío (línea 58). Se verifica si ocurrió un error con el envío (línea 60), en caso afirmativo se envía el parámetro de error para que el archivo html muestre un mensaje indicando que hubo un error (línea 63), se muestra un mensaje en la consola del aplicativo con el detalle del mismo (línea 66). Si no hay error con el envío (línea 69), se muestra un mensaje en la consola del aplicativo informando que el mensaje fue enviado de forma exitosa (línea 71); posteriormente se envía el parámetro al archivo html indicando que el mensaje se envió correctamente (línea 74), para que se muestre en el html un aviso de que el mensaje se ha enviado correctamente.

Se realiza la configuración del servidor; para esto se define el puerto mediante el cual el servidor escuchará peticiones (línea 80) y se muestra un mensaje en la consola del aplicativo que indica que el servidor ha sido iniciado y que se encuentra escuchando peticiones (línea 82).

#### 4. Conclusiones y recomendaciones

Este trabajo tiene en cuenta que existen diferentes formas en que los estudiantes entienden los conceptos sobre protocolos de red, y que algunos libros que son referencia del área (Tanenbaum, 2002) (Stallings, 2003) aunque explican con profundidad el funcionamiento de estos, no ofrecen ejemplos sobre cómo programarlos para facilitar su comprensión e integración en el desarrollo de soluciones novedosas.

1. La principal contribución de esta investigación es ofrecer material para apoyar a los profesores y alumnos en el proceso de comprensión de los protocolos, mediante ejemplos de programación de herramientas que abarcan los principales protocolos de red.
2. Dentro de los conceptos y formas de aprendizaje identificados en los trabajos relacionados con la presente investigación, se determina que la manera de enseñanza de los protocolos de red se lleva a cabo mediante la lectura de libros, explicaciones de profesores y terminología; no se efectúa a partir de un trabajo práctico como la programación.
3. El currículo para las carreras de informática de la IEEE y ACM señala que en materia de enseñanza de redes de computadoras, se debe llevar la teoría a la práctica (The Joint Task Force on Computing Curricula, 2001) y

un buen insumo para lograr este objetivo es incluir dentro de los programas académicos ejercicios prácticos basados en programación para facilitar la comprensión de los conceptos de redes.

4. Como trabajo futuro se recomienda utilizar el presente artículo; además de la guía con la instalación y configuración de los ejemplos de programación, para que sean utilizados en un curso que abarque los conceptos de protocolos de redes.
5. Como complemento de este trabajo de investigación se recomienda abarcar más protocolos de red y desarrollar más herramientas de programación que demuestren su funcionamiento.
6. Como parte de este proyecto de investigación, se recomienda analizar y estudiar el proyecto WebRTC, el cual es una iniciativa de software libre apoyada por Google, Mozilla y Opera; que busca proveer a los navegadores y dispositivos móviles, capacidades para establecer comunicaciones en tiempo real mediante un conjunto de protocolos; los cuales interactúan con los dispositivos de cámara, audio y video mediante el uso de APIs<sup>18</sup>. Al tomar en consideración la proyección de crecimiento que han tenido los dispositivos móviles en la actualidad, y la gran demanda de uso de los navegadores web, se establece como un trabajo futuro la investigación de la tecnología para aplicarla en el desarrollo de herramientas. El sitio web del proyecto Web RTC es <http://www.webrtc.org>.

## Referencias

- Berglund, A., y Lister, R. (2007). How do students understand computer network protocols? En *Proceedings of the 20th annual conference of the national advisory committee on computing qualifications (naccq)* : (pp. 15–29). National Advisory Committee on Computing Qualifications. pages 2, 3
- Booth, S. (1992). *Learning to program: A phenomenographic perspective*. pages 4
- Ince, D. (2001). *Dictionary of internet*. Oxford University Press. pages 6
- John Viega, P. C., Matt Messier. (2002). *Network security with openssl*. O'Reilly, Media Inc. pages 6
- Microsoft. (s.f.). *Introducción al correo electrónico*. Descargado 2015-08-08, de <http://windows.microsoft.com/es-419/windows/getting-started-email#getting-started-email=windows-7> pages 8
- Ramsden, P. (1992). Learning to teach in higher education. *Studies in Higher Education*, 18(1), 105-111. pages 2
- Salleh, E., Winslett, G., Bruce, C., y Tickle, A. (2005). Diagnostic learning: Using web-based self-diagnostic tools for learning abstract concepts in data network education. , 241-250. pages 3
- Santos, T. O. (2006). La fenomenografía, una perspectiva para la investigación del aprendizaje y la enseñanza.

<sup>18</sup> Siglas del inglés Application Programming Interface.

pages 2

Stallings, W. (2003). *Computer networking with internet protocols*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. pages 12

Tanenbaum, A. (2002). *Computer networks* (4th ed.). Prentice Hall Professional Technical Reference. pages 12

The Joint Task Force on Computing Curricula, C. (Ed.). (2001, septiembre). Computing curricula 2001. *Journal on Educational Resources in Computing (JERIC)*, 1(3es). Descargado de <http://doi.acm.org/10.1145/384274.384275> doi: 10.1145/384274.384275 pages 4, 5, 12