

Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles

Function Activation on Intelligent Buildings Using Mobile Devices through Voice Commands

Moumtadi Fatima

*Departamento de Ingeniería en Electrónica
Facultad de Ingeniería
Universidad Nacional Autónoma de México
Correo: fatima@fi-b.unam.mx*

Granados-Lovera Fabián

*Departamento de Ingeniería en Electrónica
Facultad de Ingeniería
Universidad Nacional Autónoma de México
Correo: fabianlovera@yahoo.com.mx*

Delgado-Hernández Julio Carlos

*Centro Universitario Valle de México de la UAEM
Universidad Autónoma del Estado de México
Correo: jcdelgadoh@uaemex.mx*

Información del artículo: recibido: mayo de 2012, reevaluado: marzo de 2013, aceptado: abril de 2013

Resumen

El desarrollo de las tecnologías de información y comunicación ha posibilitado la incorporación, a diferentes áreas de la actividad humana, de aplicaciones que permiten controlar dispositivos eléctricos y electrónicos mediante comandos de voz. Con este tipo de aplicaciones, la telemedicina ha logrado que personas cuyas capacidades físicas fueron disminuidas temporalmente incrementen su nivel de autonomía; a la educación se han incorporado herramientas que facilitan el uso de aplicaciones de tipo general a los usuarios con discapacidad física; por último, la domótica ha posibilitado a personas con movilidad reducida permanente controlar el funcionamiento de los dispositivos de uso corriente en un hogar, utilizando comandos de voz. En este artículo se describe una solución de este último tipo, desarrollada con un principio cliente-servidor. Como dispositivo cliente fue habilitado un teléfono celular con un perfil MIDP 2.0 al que se le cargó una aplicación propietaria desarrollada en Java MicroEdition; como servidor, una PC habilitada como servidor Web, equipada con un motor de reconocimiento de palabras y una interfaz de integración hacia una red Konnex. El prototipo funcional del sistema desarrollado permite tener control sobre tres cargas y su confiabilidad, de 87% de acierto con el reconocedor de comandos, fue incrementada al incorporar una interfaz gráfica con menú de comandos desplegable sobre monitor.

Descriptores:

- control
- comandos de voz
- aplicaciones cliente-servidor
- redes Konnex

Abstract

Development of information and communication technologies has allowed the incorporation into different areas of human activity of apps that control electrical and electronic devices through voice commands. With these apps, in telemedicine people affected by some temporary decrease in their physical capacities have improved their level of autonomy; utilities have been added to educational environments to facilitate the use of IT applications to users with physical disability; finally, home automated solutions have made possible to any person with permanent limited mobility to take control over home devices using voice commands. In this article a home automated solution, developed over a client-server principle is presented. As the client device a MIDP 2.0 cell phone with a Java MicroEdition application loaded was used; as server a web server PC was used serving also as gateway towards a Konnex network, added with a speech recognizer engine. Fully functional prototype developed allowed take control over 3 devices with 87% success of the speech recognizer reliability, this percentage improved after the use of a drop-down menu of commands displayed over the monitor.

Keywords:

- control
- voice commands
- client-server applications
- Konnex network

Introducción

La marcada tendencia de desarrollo en las tecnologías de información y comunicación de los últimos años ha posibilitado la incorporación de aplicaciones que permiten controlar dispositivos eléctricos y electrónicos mediante comandos de voz a diferentes entornos de la actividad humana cotidiana. Esta incorporación no ha carecido de problemas en la implementación, pues la imitación de habilidades naturales del ser humano constituye uno de los mayores retos a los que científicos e ingenieros se enfrentan, debido a que la sustitución de actividades motrices o de discernimiento requiere complejos sistemas de modelado, discriminación y toma de decisiones, como será demostrado más adelante.

En telemedicina, por ejemplo, se han documentado adelantos en el diseño y realización de diversos dispositivos médicos. Pueden citarse como ejemplo los desfibriladores con módulos de comando de voz presentados en 2007 por el Instituto Central de Investigación Digital cubano, con capacidad de “reducir la ocurrencia de

errores de operación y facilitar su uso, de vital importancia, teniendo en cuenta las características de este tipo de equipo médico y las circunstancias en que comúnmente se utiliza, se ha logrado que personas afectadas por algún tipo de disminución temporal de sus capacidades (visuales, auditivas o táctiles) mejoren el nivel de autonomía de su vida personal” (Forneiro y Piñeiro, 2007). También se citan las sillas eléctricas con capacidad de guiado autónomo controlado por voz, presentadas en el Primer Congreso Internacional de Domótica, Robótica y Teleasistencia para todos (Alcubierre *et al.*, 2005). Particular mención debe hacerse de la interfaz hombre- máquina empleado en este dispositivo, que incorporaba un sistema de reconocimiento de voz que involucraba un parametrizador y un reconocedor de comandos de voz, compuesto a su vez por un logaritmo de modelado de lenguaje y uno de modelado acústico (figura 1). Este modelo, aunque simple en su concepción, incorpora complejos mecanismos de discriminación y toma de decisiones basadas en el comando de voz, y sirvió como precedente conceptual directo a la realización del prototipo desarrollado por nuestro equipo de investigación.

En el área de la educación también se han alcanzado logros importantes, a través de la incorporación de los denominados “programas de autoayuda”, aplicaciones o utilidades, cuyo propósito fundamental es facilitar el uso de aplicaciones informáticas de tipo general a usuarios con discapaci-

Figura 1. Diagrama de bloques funcional de un reconocedor de voz (Alcubierre *et al.*, 2005)

dad física. Si bien es cierto que se ha desarrollado un gran número de programas dirigidos a aumentar en los alumnos el enfoque de la atención, aumentar los estímulos visuales, disminuir la sensación de aislamiento y controlar variables del entorno en general (Castellano *et al.*, 2003), estos no siempre han sido inclusivos en su diseño, descuidando aspectos relacionados con las capacidades físicas de algunos de los usuarios de estas soluciones. Un ejemplo de integración de soluciones para usuarios con discapacidades físicas puede consultarse en Moralejo *et al.* (2010), en donde se explica cómo el entorno JClic (un entorno de cuyo desarrollo fue visado por el Departamento de Universidades, Investigación y Sociedad de la Información -DURSI- de la Generalitat de Cataluña), usado para la creación y realización de actividades educativas multimedia, permitió la inclusión del sistema Sphinx, (un servidor de búsqueda de textos de distribución libre desarrollado por Sphinx Technologies Inc.) para controlar sus funciones por comandos de voz (Moralejo *et al.*, 2010). Sphinx se basa en el uso de modelos ocultos de Markov (HMM, *Hidden Markov Models*), requiriendo para su correcto funcionamiento una etapa previa de aprendizaje de las características (o parámetros) de un conjunto de unidades de sonido, para después encontrar la secuencia de unidades de sonido más probable dada una señal de voz particular (Lamere *et al.*, 2003).

Son las soluciones domóticas, sin embargo, las que más se han beneficiado de la incorporación de estos mecanismos de reconocimiento de voz, permitiendo a cualquier persona con movilidad reducida permanente, controlar el funcionamiento de los dispositivos de uso corriente en un hogar, utilizando comandos que generalmente incluyen el objeto que se pretende controlar, la acción que se pretende que el objeto realice y el lugar donde se encuentra el mismo. Las acciones que pueden ser ejecutadas con soluciones domóticas con reconocimiento de comandos de voz (basadas generalmente en redes Konnex-KNX- y un algoritmo de reconocimiento de comandos) pueden dividirse en tres grandes grupos:

1. La ejecución de comandos de voz dentro de casas y edificios inteligentes (cargas simples, como prender o apagar luces).
2. Disparar procesos en un servidor de aplicaciones (accionar reproductores multimedia o archivos que ejecuten funciones dentro del servidor).
3. Direccionamiento y visualización de páginas Web: noticias, clima, etcétera.

La solución propuesta en el presente artículo es una aplicación domótica cliente-servidor, compuesta por

tres bloques funcionales: un cliente (teléfono celular modelo W810i), habilitado con una aplicación propietaria desarrollada en Java Micro Edition (J2ME), que permite generar y enviar señales de voz a través de Internet (usando una red WiFi 802.11g); un servidor (una PC con procesador intel CORE i5 a 2.4GHz, con 4GB en RAM y un sistema operativo Windows 7 ultimate de 64 bits), habilitado como servidor Web y servidor domótico también, con un motor de reconocimiento de palabras aisladas, y una instalación EIB/KNX. Con este diseño se controlaron tres cargas, dos interruptores y un ventilador (figura 2).

El cliente

Como cliente fue habilitado un teléfono celular modelo W810i de la marca Sony Ericsson con el perfil MIDP 2.0 integrado a su plataforma física. Este perfil (Mobile Information device Profile 2.0-MIDP 2.0 en inglés) fue aprobado en noviembre del 2002 en la solicitud de especificación Java 118 (Java Specification Request-JSR 118 en inglés). Si bien en diciembre de 2009 la versión MIDP 2.0 fue mejorada con la publicación del documento JSR 271 (2009) que regula la versión MIDP 3.0, la documentación establece la completa compatibilidad entre versiones (MIDP 3.0. 2012) y el funcionamiento de la aplicación sobre esta plataforma garantiza su funcionamiento prácticamente en cualquier teléfono celular que cuente con el perfil MIDP 2.0 y superior en el que se hayan cargado las interfaces de programación de aplicación (*Application Program Interface-API*) `javax.microedition.rms` (para realizar la gestión de la base de datos), `javax.microedition.media` y `javax.microedition.media.control` (para la captura y reproducción de audio), y `javax.microedition.io.HttpConnection` (para el envío de la información sobre Internet).

La aplicación fue creada utilizando la herramienta de desarrollo Java "Eclipse" versión 3.3.2, que además de asegurar la validación, compilación incremental, re-

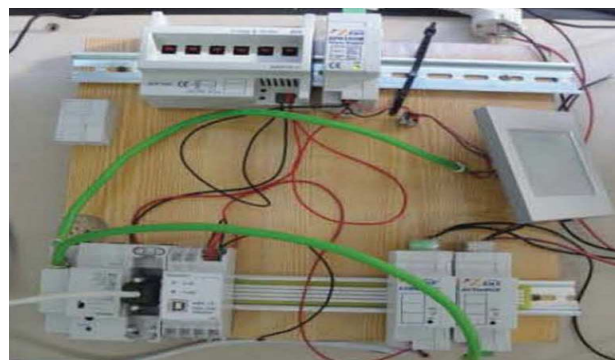


Figura 2. Maqueta-prototipo KHX del modelo desarrollado

ferenciamiento cruzado, un editor XML y ayuda para el desarrollo de código (figura 3), permite la interacción con el “Wireless toolkit” de Sun Microsystems (en este caso, la versión 2.5.1) (figura 4).

Las API `javax.microedition.media` y `javax.microedition.media.control` deben contener en su configuración el método “`setRecordStream`” y en la API `javax.microedition.io.HttpConnection` se configuran los métodos “`connect`” y “`write`”. Al final se crean los archivos que se instalarán en el dispositivo móvil (Galvez y Ortega, 2003) (con extensiones `.jar` y `.jad`) (figura 5).

El servidor

La arquitectura del servidor se muestra en la figura 6. El servidor se compone de tres bloques conceptuales: Un

servidor Web con una base de datos e intérpretes PHP y Pearl; un motor de reconocimiento de voz y un servidor domótico, todo esto con el fin de ejecutar funciones de procesamiento de la señal de voz y de servir además como pasarela hacia la red domótica. Asimismo, previendo que el sistema de reconocimiento de comandos de voz falle, se agregó un menú desplegable sobre el monitor, lo que garantiza el funcionamiento del sistema, incrementando la confiabilidad de la solución. Los comandos introducidos se denominan comandos “por activación de click”, haciendo referencia a su introducción a través del teclado.

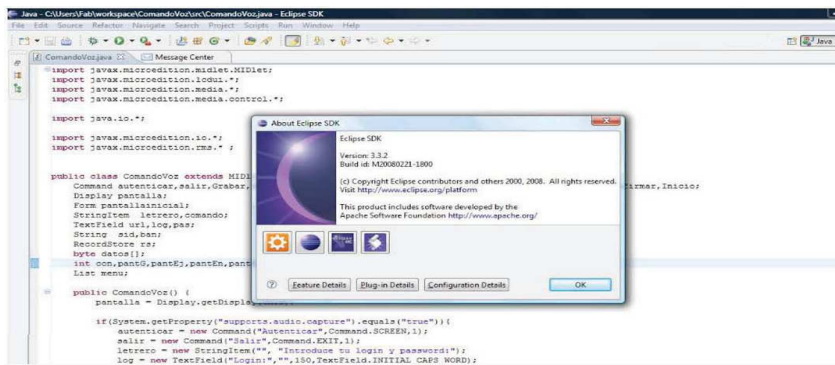


Figura 3. Editor Java “Eclipse” version 3.3.2

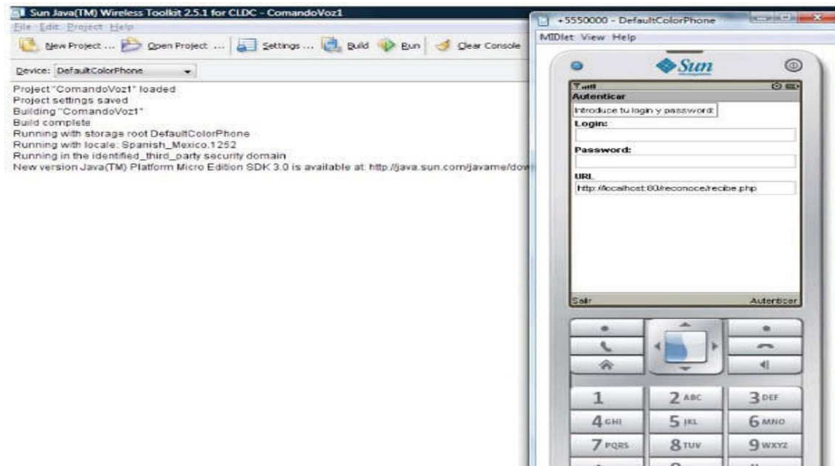


Figura 4. Wireless toolkit de Sun Microsystems version 2.5.1

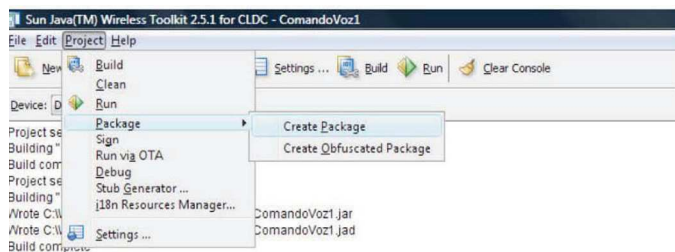


Figura 5. Creación de los archivos .jad y .jar

El servidor Web

El software utilizado para habilitar el servidor web fue el servidor de distribución libre XAMPP (que consta de un servidor MySQL, de un servidor Web Apache y de intérpretes para los lenguajes PHP y Pearl) (Dvorski, 2007). Este software ofrece un conjunto de funciones altamente eficientes para el manejo de archivos y de sesiones simultáneas generadas por clientes múltiples en línea a través de la gestión de sockets. Para el prototipo desarrollado se utilizó la version 1.5.4 Beta de XAMPP para Windows.

En la figura 7 podemos observar un fragmento de la página principal del sistema llamada *recibe.php*, diseñada para recibir los datos enviados por el cliente y generar una serie de instrucciones que permitan ejecutar la acción requerida: la autenticación del usuario; el reconocimiento de una señal acústica, el entrenamiento del módulo de comandos de voz o la ejecución del comando mismo. El comando, a su vez, puede ser dirigido

al sistema de reconocimiento de voz, a la base de datos o a la red domótica, según corresponda.

El motor de reconocimiento de voz: etapa de entrenamiento

El segundo bloque conceptual del servidor, el motor de reconocimiento de voz, divide sus funciones en dos partes para su correcto funcionamiento. En la primera etapa, denominada “etapa de entrenamiento”, se obtienen las características acústicas de un determinado número de señales de audio que el sistema usará como comandos después de su reconocimiento. En la segunda etapa, denominada “etapa de reconocimiento”, las señales se procesan para caracterizar el comando de voz mediante la extracción de características acústicas por medio del algoritmo de clasificación de patrones llamado “modelos ocultos de Markov” y de los coeficientes Mel-Cepstral. Para la realización de estas funciones se usó la versión 7.9.0.259 R2009b de 64 bits del software comercial MatLab (desarrollado por Mathworks), usando dos aplicaciones denominadas “*entrena_web.exe*” y “*reconoce_web.exe*”, ejecutados desde un código PHP.

Al ser éste un sistema multiusuario, el usuario que necesite utilizar el sistema deberá abrir una sesión. Desde la interfaz de administración de este módulo es posible gestionar a los usuarios del sistema, introducir la ruta hacia ciertos archivos en el servidor, direcciones URL o comandos domóticos por activación de click. La interfaz de ingreso al sistema de administración se muestra en la figura 8.

Como se muestra en la figura 9, la gestión de la dirección IP del servidor se realiza desde la ventana de la sesión de usuario.

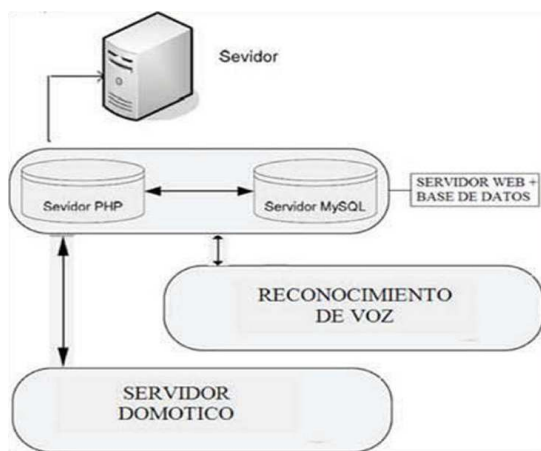


Figura 6. Arquitectura del servidor

```

<?php
session_start(); //inicia sesion
$time_ini=microtime(true); //inicia cronometro

$conf=$HTTP_POST_VARS['conf']; //variable de configuración del sistema

if(isset($_SESSION['user'])) { //verifico si tiene sesion abierta
    $dire = getcwd()."\\".$_SESSION['user'];
    chdir($dire); //cambia de directorio

    if($conf < 4){
        $dir=$HTTP_POST_VARS['dat']; //recupera la señal de voz
        $archivo=fopen("audio.txt", "w");
        if ($archivo) {
            fputs($archivo, $dir); //escribe la señal de voz txt
        }
        fclose ($archivo);
    }
}
  
```

Figura 7. Fragmento de la página “*recibe.php*”



Figura 8. Interfaz de ingreso al sistema de administración

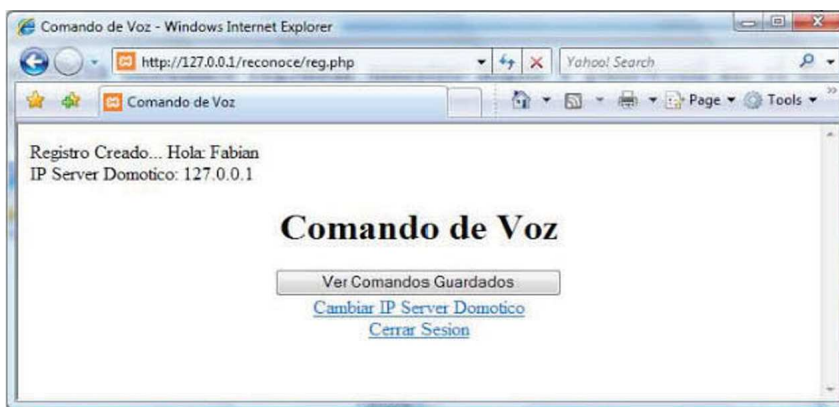


Figura 9. Administración del sistema de reconocimiento de voz

La figura 9 también muestra que desde esta misma ventana es posible ver los comandos guardados para el usuario, cuya sesión esté abierta, pero la agregación y configuración de estos comandos (tanto los de voz como los de click) se realiza desde la página de administración del servidor XAMPP, como se muestra en la figura 10.

Es posible ver, al analizar la figura 10, que la base de datos del servidor se usa para guardar los perfiles de cada uno de los usuarios, lo que permite realizar su autenticación y reconocer la lista de comandos registrados en su perfil a ejecutarse por click. La autenticación es necesaria debido a estos factores y al hecho de que el reconocimiento de los comandos de voz es dependiente del locutor; al momento de abrir su sesión personal, el locutor tiene acceso a configuraciones específicas de comandos (de voz y por click) registradas.

El módulo de entrenamiento de voz se muestra en la figura 11.

Para registrar las señales acústicas de cada locutor, en la etapa de entrenamiento se recolectan siete veces, de acuerdo con lo sugerido en (Lovera, 2010). El motor de reconocimiento de voz hace pasar a cada una de las señales a través de un bloque de almacenamiento en

memoria y preprocesado de señales, cuyo diagrama se muestra en la figura 12.

El bloque de preprocesado tiene como finalidad preparar la señal antes de entrar a la etapa de entrenamiento o reconocimiento de comandos de voz. Así, una vez que se obtienen las señales acústicas de parte del locutor, la señal pasa por un proceso de conversión analógico-digital de la voz, en el que cualquier ruido presente debe minimizarse o, mejor aún, anularse. Con este fin pueden utilizarse dos posibles algoritmos: la sustracción espectral y la cancelación adaptativa de ruido (Huang y Rabiner, 1991). Debido a que la aplicación del primer algoritmo requiere activar la detección de voz, en la solución propuesta se utilizó una cancelación adaptativa de ruido.

El preénfasis, que consiste en la eliminación de las frecuencias altas de la señal mediante un filtro pasa bajas, precede al módulo final del bloque, la detección de voz. La complejidad de la detección de voz se incrementa en entornos extremadamente ruidosos, en los que la anulación de señales espurias del módulo de cancelación de ruido no fue suficientemente fiel. En éste módulo se realzan las altas frecuencias presentes en la señal, calculando el nivel de los umbrales de ruido para

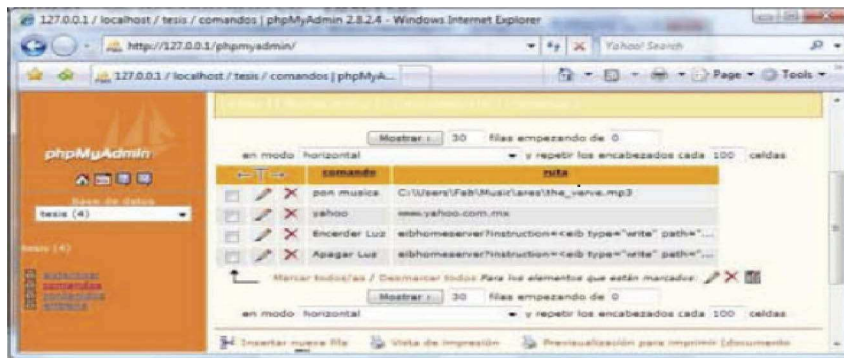


Figura 10. Administración del servidor XAMPP

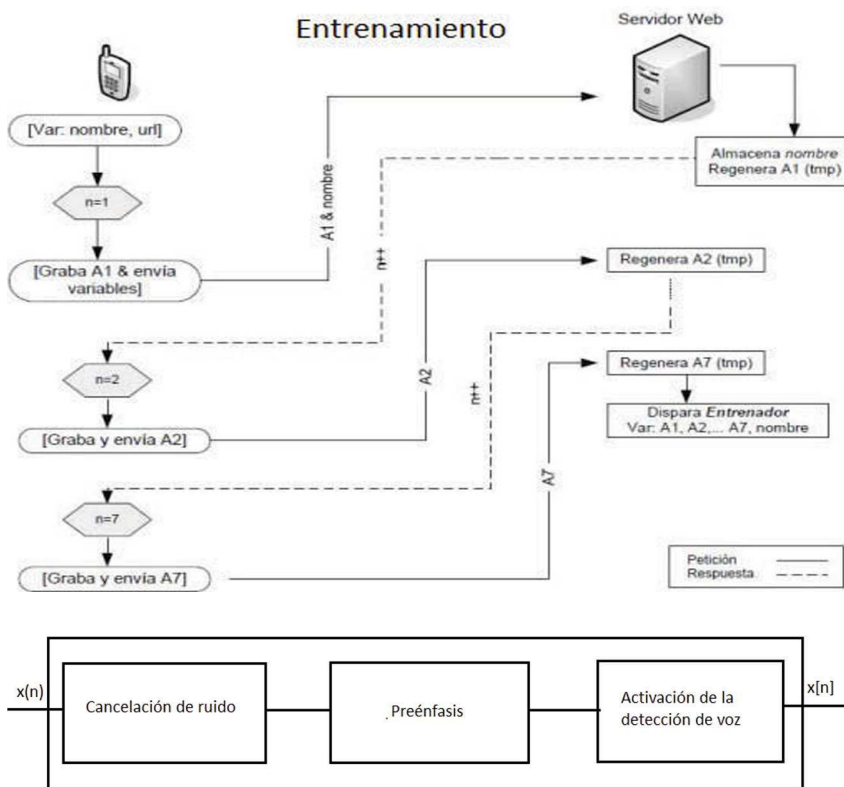


Figura 11. Diagrama esquemático del módulo de entrenamiento de voz

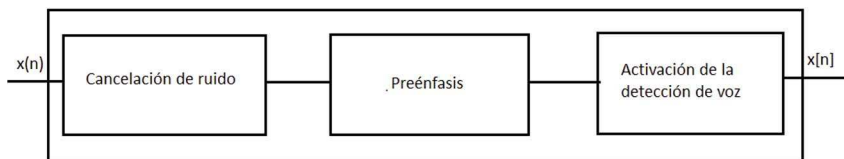


Figura 12. Módulo de pre-procesado de señales

después restarlos. Una vez detectadas las palabras que forman el comando de voz, se aplica una ventana de entramado de Hamming por cada 128 muestras, con la finalidad de hacer más eficiente el procesamiento en frecuencia, evitando modificar el contenido espectral de la señal tras su división en tramas. Una vez preparadas las señales de audio, el sistema genera un archivo denominado audio7.bat, que se forma a partir de una línea de comando presente en el antes mencionado script "recibe.php". Los contenidos de este archivo se envían al último bloque del pre-procesado, el módulo de activación de la detección de voz, para ser caracterizadas mediante la extracción de los contenidos acústicos de las señales, por medio de los coefi-

cientes Mel-Cepstral y los algoritmos de clasificación de patrones llamados modelos ocultos de Markov (HMM).

Los coeficientes Mel-Cepstral se componen de dos partes: el cálculo Cepstral y un método denominado escalado Mel. El cálculo Cepstral se utiliza para definir un filtro para el tracto vocal con un procesamiento homomórfico, un proceso de transformación al dominio lineal de señales combinadas de manera no lineal, mientras que el escalado Mel permite, de acuerdo con las características fisiológicas del oído humano, afectar las señales de voz de manera no lineal, en una escala denominada "escala Mel" (Deller *et al.*, 2000) que corresponde a la figura 13.

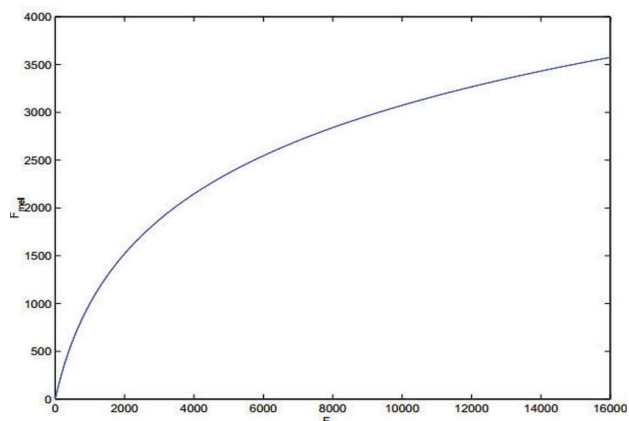
El algoritmo de clasificación de patrones llamado “modelos ocultos de Markov” se utiliza en el procesamiento de señales de voz debido a su capacidad para modelar procesos aleatorios. Para robustecer el funcionamiento del reconocedor se implementa una medida de confianza de la palabra reconocida, lo que permitirá evitar que palabras pronunciadas, pero fuera del vocabulario sean reconocidas como comandos registrados en sistema.

Al final del proceso el cliente recibe un mensaje de notificación de conclusión del proceso de aprendizaje, generado por el servidor.

Se ha documentado, además del algoritmo de modelos ocultos de Markov, el uso de otros procedimientos de reconocimiento de comandos de voz, basados en el uso de fonemas. Sin embargo, el uso de fonemas no está exento de dificultades, debido a que la identificación de las fronteras entre ellos por lo regular es difícil de encontrar en representaciones acústicas de voz (Oropeza y Suárez, 2006). Este procedimiento comienza con la producción de un mensaje hablado por el usuario, utilizando una forma o estilo de habla restringido, con un vocabulario reducido, con palabras pronunciadas de forma aislada, frases tipo, etcétera. A partir de la señal de voz, un proceso de clasificación, basado en reconocimiento de patrones asociados a diferentes unidades lingüísticas (palabras, fonemas, sílabas, etcétera), permite a la interfaz de comunicaciones extraer de la base de datos la información solicitada por el usuario (Fandiño, 2005). Debido a estas limitaciones el grupo de trabajo se decantó por la opción de los modelos ocultos de Markov, que se expuso.

El motor de reconocimiento de voz: etapa de reconocimiento

El diagrama del módulo de reconocimiento se ilustra en la figura 14.



Para reconocer un comando de voz, del mismo modo que en el entrenamiento, desde el archivo “recibe.exe” se manda a ejecutar un archivo con extensión .bat (en este caso, “start_audio.bat”). Por medio del código PHP se recupera el nombre del comando reconocido y se envía al cliente, esperando confirmación.

Cuando el sistema de reconocimiento de voz falla y es necesario usar el esquema de click para hacer ejecutar los comandos, el servidor envía al cliente el listado de los comandos a los que tiene acceso según la información del servidor MySQL de XAMPP. El cliente a su vez, escoge de la lista el comando necesario y lo envía al servidor.

El servidor doméstico

En cuanto al tercer elemento de la solución, el servidor doméstico, se trata de un Linux EIB Home Server (Werntgates *et al.*, 2012), quien funge como pasarela al bus KNX. Esencialmente consta de dos componentes: un **homedriver**, una interfaz entre homeserver y el Bus KNX, encargada de enviar los mensajes de gestión de dispositivos en el bus KNX mediante el puerto serie del ordenador utilizando el protocolo FT1.2. El segundo componente es el denominado **homeserver**, una interfaz de comunicación entre la capa de aplicación y el homedriver que se encarga de crear sockets de comunicación entre los clientes y el homedriver para el paso de información entre el cliente y la red doméstica. El número de clientes teóricamente es ilimitado.

La figura 15 muestra el diagrama por bloques del sistema Konnex (EIB / KNX) (Sistema KNX, 2012).

El estándar Konnex (KNX) es la iniciativa de tres asociaciones europeas: BCI (Francia, sistema Batibus), EIB (Bélgica, sistema EIB) y Europe Home Association System (Holanda, sistema EHS). Como resultado de su unión se crea la asociación KNX con sede en Bélgica, con el propósito de competir con los sistemas domésticos de nortea-

Figura 13. Correspondencia entre la escala en Hertz y la escala Mel de señales de voz (Deller *et al.*, 2000)

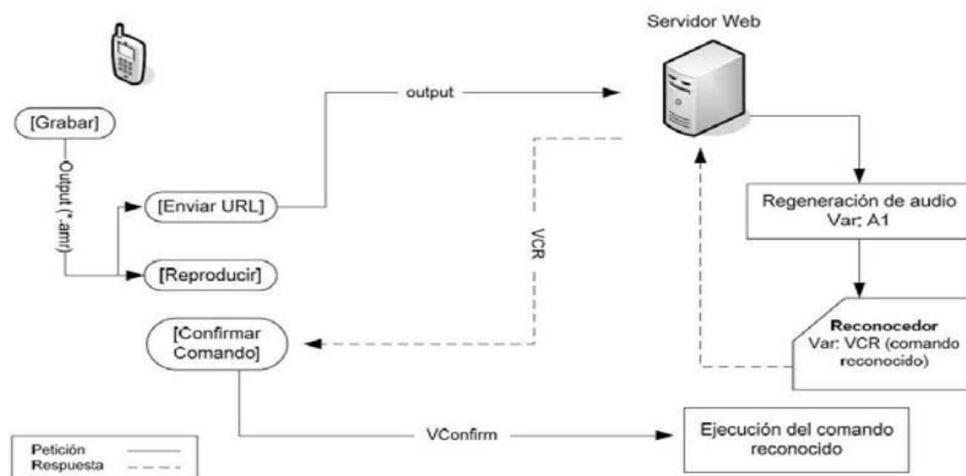


Figura 14. Diagrama esquemático del módulo de reconocimiento de voz

mérica Lonworks y CEBus. En el año 2006 el CENELEC (Comité Europeo de Normalización Electrotécnica) le otorga al estándar el estatus de norma internacional bajo el número ISO/IEC 14543-3 (Sistema KNX, 2012), convirtiéndose en el primer estándar a nivel mundial para la comunicación de control de viviendas y edificios totalmente independiente de la plataforma tecnológica (física o lógica) sobre la que se implemente.

Soporta conexiones sobre cobre y radiofrecuencias y puede ser configurado desde una PC (modo de sistema), desde fábrica (modo simple) o autoconfigurarse dinámicamente (modo A). Esto le otorga al sistema flexibilidad en la implementación, convirtiéndose en un sistema modular (con una capacidad de hasta 64 dispositivos controlados por línea), posible de ser implementado en cualquier nivel, con una capacidad máxima de 15 líneas o 960 dispositivos integrados en las denominadas áreas. Un acoplador de área permitiría integrar hasta 15 áreas distintas (14400 dispositivos), lo que representa la capacidad máxima de un sistema KNX. Cada línea debe disponer de su propia fuente de alimentación.

El direccionamiento que se utiliza en el sistema tiene dos niveles; uno físico (compuesto de tres campos: área -4 bits-, línea -4 bits- y dispositivo -8 bits) y uno de grupo. Este direccionamiento puede ser de dos (grupo principal / subgrupo) o de tres (grupo principal / grupo intermedio / subgrupo) niveles.

El direccionamiento que se utiliza en el sistema tiene dos niveles; uno físico (compuesto de tres campos: área -4 bits-, línea -4 bits- y dispositivo -8 bits) y uno de grupo. Este direccionamiento puede ser de dos (grupo principal / subgrupo) o de tres (grupo principal / grupo intermedio / subgrupo) niveles.

Figura 15. Diagrama de bloques del servidor KNX (Sistema KNX, 2012)

En el caso de la solución propuesta, el medio de conexión fueron las radiofrecuencias y el esquema de direccionamiento de las cargas controladas pero el sistema fue de dos niveles.

El estándar reconoce cuatro tipos de comandos:

- *Read*: para conocer el estado de algún objeto por medio de su dirección de grupo.
- *Write*: para cambiar el estado de un dispositivo por medio de su dirección de grupo.
- *Subscribe*: para monitorear los cambios de algún dispositivo.
- *Unsubscribe*: para cancelar el comando subscribe.

En la solución propuesta a través de la instrucción Write se tomó control sobre los dispositivos (ventilador y luces).

Resultados

Para realizar las pruebas se montó una maqueta KNX para tomar el control de tres cargas (dos luces y un ventilador) con los siguientes componentes (figura 2):

- Siemens, RS232 Interface N 148-042.
- Siemens, 5WG1 191-5AB11 (fuente).
- ZENNiO, ZPS 160M ZN1PS-160M.
- ZENNiO, LUZEN ONE ZN1DI-4001.
- ZENNiO, InZennio Z38 ZN1VI-TP38.
- Weinzierl, KNX USB Interface 310.

Las cargas pueden accionarse utilizando comandos de voz y por una lista de comandos desplegada sobre la pantalla del dispositivo móvil. Las pruebas realizadas de acceso a través de la lista de comandos mostraron una confiabilidad del sistema de 99%; sin embargo, para el caso de los comandos de voz, se cargó en los perfiles de los 10 usuarios registrados una lista de comandos (como había sido comentado, con siete repeticiones de los comandos por parte de los usuarios a fin de alimentar la base de datos) y se realizaron pruebas de reconocimiento, repitiendo 10 veces cada comando para poder obtener un porcentaje de error que caracterizara al reconocedor de voz (el porcentaje que se muestra es el promedio de los resultados obtenidos). Los comandos utilizados fueron los que se marcan en la tabla 1. En la tabla, además de los comandos, se marcan los porcentajes de reconocimiento del comando por comando, por usuario y en total.

A partir de los datos de la tabla puede concluirse que, en términos generales, el desempeño del sistema fue estable y que no se registraron pérdidas de conexión

o fallos al momento de la interacción entre los componentes del mismo. Los comandos en la etapa de entrenamiento se registraron correctamente en el servidor, lo que es fácilmente comprobable pues los comandos fueron reconocidos, con mayor o menor éxito en la etapa de reconocimiento, pero todos sin excepción.

De la última línea de la tabla se desprende que el promedio de confiabilidad de funcionamiento del servidor (entendiendo que la confiabilidad se refiere al porcentaje de reconocimiento de los comandos) fue 87%. Estos resultados se obtuvieron en entornos de grabación y prueba con niveles de ruido ambiental bajo y controlado, hablando además fuerte y con una articulación correcta.

El usuario 7 fue el que obtuvo mayor porcentaje de reconocimiento de los comandos de voz (92.85) y el usuario que menor porcentaje obtuvo fue el 8 (81.42), atribuible a su dicción y tono de voz. Por otro lado, el comando que mejor reconocimiento del sistema tuvo fue “persianas”, mientras que el que peor porcentaje de reconocimiento presentó fue “televisión”. Menciona especial es que los últimos 4 comandos no accionaban ningún dispositivo, y la manera de comprobar el reconocimiento del comando por parte del sistema fue al detectar tramas de solicitud de confirmación por parte del servidor hacia el cliente. Si descartamos estos últimos cuatro comandos, que no accionaban dispositivo alguno, entonces vemos que el comando que mayor reconocimiento tuvo fue “ventilador” (90%), mientras que los menos reconocidos fueron los relacionados con apagar las luces (85.71% en ambos casos).

El tiempo de reconocimiento de los comandos fue muy alto, en promedio de 11 segundos entre la emisión del comando y el accionar del dispositivo requerido (los tres que fueron contemplados en la maqueta), atribuible al uso de un software de tipo general como lo es Matlab en la solución propuesta. La conexión entre cliente y servidor se realizó sobre redes WiFi 802.11g y el servidor estaba dedicado únicamente a la aplicación desarrollada, de manera tal que se excluye que el retardo tan alto fue debido al alto nivel de tráfico, quedando como causa altamente probable el hecho de involucrar soluciones comerciales de alto nivel.

Aun cuando la activación de un dispositivo requiere que el comando (si fue emitido de manera acústica) sea confirmado por parte del cliente, los tiempos de retardo son muy altos.

En estudios posteriores será posible comprobar si esta es realmente la causa, al escribir programas en software de bajo nivel (lenguajes Java o C++, por ejemplo), especialmente desarrollados para realizar las funciones requeridas por la solución y realizar nuevamente las

Tabla 1. Porcentaje de reconocimiento de comandos de voz

Comando	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	%
Luz uno	100.00	71.43	85.71	85.71	85.71	85.71	100.00	71.43	85.71	100.00	87.14
Luz dos	71.43	100.00	100.00	85.71	85.71	100.00	85.71	100.00	71.43	85.71	88.57
Apagar luz uno	85.71	71.43	85.71	85.71	100.00	100.00	85.71	71.43	71.43	100.00	85.71
Apagar luz dos	85.71	100.00	85.71	100.00	85.71	85.71	100.00	71.43	71.43	71.43	85.71
Ventilador	100.00	100.00	85.71	85.71	100.00	85.71	100.00	85.71	85.71	71.43	90.00
Apagar ventilador	100.00	71.43	100.00	85.71	100.00	85.71	100.00	71.43	71.43	85.71	87.14
Música	85.71	85.71	85.71	71.43	100.00	71.43	100.00	85.71	85.71	100.00	87.14
Persianas	100.00	71.43	100.00	100.00	85.71	100.00	85.71	85.71	100.00	100.00	92.85
Puerta	85.71	85.71	71.43	85.71	85.71	85.71	85.71	85.71	85.71	85.71	84.28
Televisor	85.71	85.71	71.43	71.43	85.71	71.43	85.71	85.71	100.00	71.43	81.42
Total	90.00	84.28	87.14	85.71	91.42	87.14	92.85	81.42	82.85	87.14	87.00

mediciones correspondientes, comparándolas con los resultados aquí obtenidos.

Conclusiones

En esta investigación se logró obtener un prototipo funcional del sistema domótico, el cual sirve de interfaz entre los usuarios y los dispositivos conectados a una red KNX, logrando ejecutar comandos de voz dentro de esta. La aplicación desarrollada en J2ME fue probada exitosamente en el teléfono celular modelo W810i de la marca Sony Ericsson y puede ser ejecutada en cualquier dispositivo que cuente con la capa de configuración CLDC 1.1 y perfil MIDP 2.0.

El reconocimiento de voz depende en gran medida del ambiente y el ruido que se pueda presentar en él, por lo que la grabación de los comandos está sujeta a condiciones estrictas de control sobre el entorno.

El reconocimiento de los comandos se realiza de manera independiente para cada uno de los locutores registrados en el sistema, tanto los comandos de voz como los de click. Sin embargo, la capacidad de gestión del manejador MySQL y la del disco duro de la PC donde el servidor XAMPP fue alojado (500 GB), permite aseverar que la capacidad del sistema no limita la cantidad de usuarios que pudieran tener acceso a él.

Al ser responsabilidad del servidor la mayor parte del funcionamiento del sistema, las características físicas y del software cargado en el cliente son mínimas, lo que convierte a la solución en ampliamente utilizable.

A pesar de que el sistema puede caracterizarse como altamente confiable (con un porcentaje de 87% en comandos de voz y hasta casi 100% en comandos de

click), se mostró claramente lento en los tiempos de respuesta, quedando como reto resolver el problema de minimizar los tiempos de respuesta.

Referencias

- Alcubierre J.M., Minguez J., Montesano L. *et al.* Silla de ruedas inteligente controlada por voz, en: Memorias del Primer Congreso Internacional de Domótica, Robótica y Teleasistencia para todos, Madrid, España, noviembre de 2005.
- Castellano R., Sacco A., Zurueta S. La utilización de software de uso general y aplicaciones específicas en el área de las discapacidades motrices, en: Memorias del IV Congreso Iberoamericano de Informática en la Educación Especial, Madrid, España, diciembre de 2003.
- Deller J. *et al.* *Discrete Time Processing of Speech Signals*, UK, McMillan Publishing Co, 2000.
- Dvorski D. Installing, Configuring and Developing with XAMPP [en línea], marzo de 2007 [fecha de consulta: junio de 2012]. Disponible en: <http://dalibor.dvorski.net/downloads/docs/InstallingConfiguringDevelopingWithXAMPP.pdf>
- Especificaciones del sistema KNX [en línea], [fecha de consulta: septiembre de 2012]. Disponible en: <http://www.knx.org/es/descargas-soporte/descargas/>
- Fandiño D. Estado del arte en el reconocimiento automático de voz Universidad Nacional de Colombia, Seminario de Investigación [en línea], abril de 2005 [fecha de consulta: abril de 2013]. Disponible en: http://www.geocities.ws/deibywolf/Estado_del_arte.pdf
- Forneiro Y., Piñero L. Programador para módulos de comandos de voz, en: Memorias del VII Congreso de la Sociedad Cubana de Bioingeniería, Ciudad de la Habana, Cuba, julio de 2007.
- Gálvez S., Ortega L. Java a tope: J2ME (Java 2 Micro Edition)–versión electrónica, Universidad de Málaga [en línea], 2003 [fe-

- cha de consulta: agosto de 2012]. Disponible en: <http://www.lcc.uma.es/~galvez/ftp/libros/J2ME.pdf>
- Huang B., Rabiner L. Hidden Markov Models for Speech Recognition. *Revista Technometrics*, volumen 33 (número 3), agosto 1991: 251-272.
- JSR 271: Mobile Information Device Profile 3 [en línea], noviembre de 2009, [fecha de consulta: junio de 2012]. Disponible en <http://www.jcp.org/en/jsr/detail?id=271>
- Lamere P., Kwok P., Walker W., Gouva E., Singh R., Raj B., Wolf P. Design of the CMU Sphinx 4 Decoder [en línea], agosto de 2003 [fecha de consulta: mayo de 2012]. Disponible en: <http://www.merl.com/papers/docs/TR2003-110.pdf>
- Lovera F. *Sistema de reconocimiento de voz con dispositivos móviles aplicado a casas inteligentes*, tesis, Facultad de Ingeniería, UNAM, México, 2010.
- MIDP 3.0. [en línea], julio de 2012, [fecha de consulta: septiembre de 2012]. Disponible en: http://www.developer.nokia.com/Community/Wiki/MIDP_3.0
- Moralejo L., Ostermann S., Sáncz C. Adaptación a JClíc para alumnos con deficiencia motriz, mediante comandos por voz [en línea], octubre de 2010 [fecha de consulta: marzo de 2012]. Disponible en: <http://repositorio.cuaed.unam.mx:8080/jspui/handle/123456789/1490>
- Nilsson M., Ejnarsson M. *Speech Recognition Using Hidden Markov Model-Performance Evaluation in Noisy Environment*, tesis, Departamento de telecomunicaciones y procesamiento de señal, Instituto Blekinge de Tecnología (Suecia), 2002.
- Oropeza J., Suárez S. Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. *Computación y Sistemas*, México, volumen 9, (número 3), 2006: 270-286.
- Werntgates H., Neumann J., Vinarski V. Controlling EIB/KNX Devices from Linux Using USB [en línea], 2012 [fecha de consulta: agosto de 2012]. Disponible en: http://www.cs.hs-rm.de/~werntges/proj/knxusb/KSC2005-Paper_final.pdf

Este artículo se cita:

Citación estilo Chicago

Moumtadi, Fatima, Fabián Granados-Lovera, Julio Carlos Delgado-Hernández. Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles. *Ingeniería Investigación y Tecnología*, XV, 02 (2014): 175-186.

Citación estilo ISO 690

Moumtadi F., Granados-Lovera F., Delgado-Hernández J.C. Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles. *Ingeniería Investigación y Tecnología*, volumen XV (número 2), abril-junio 2014: 175-186.

Semblanza de los autores

Fatima Moumtadi. Obtuvo la maestría en sistemas de radiodifusión satelital y el doctorado en televisión en la Facultad de Radiodifusión y Televisión de la Universidad Técnica de Comunicaciones e Informática de Moscú, Rusia (MTUCI). Se desarrolló profesionalmente en el área de radiofrecuencia. Ha publicado artículos en congresos y revistas nacionales e internacionales. Actualmente es profesora de carrera en el departamento de Electrónica en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

Fabián Granados-Lovera. Ingeniero en telecomunicaciones y maestro en el área de radio-comunicaciones, por el Departamento de Telecomunicaciones de la División Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

Julio Carlos Delgado-Hernández. Finalizó cursos de especialización en sistemas de conmutación y de transmisión digital (1996) y de doctorado en ciencias técnicas (2002) en la Facultad de Sistemas de Electrocomunicación de la Universidad Técnica de Comunicaciones e Informática de Moscú, Rusia (MTUCI). Durante 4 años, hasta junio de 2009, se desempeñó como director de Networking de la empresa Interstice net-@ SA de CV, actividad que combinó con labores de docencia en la Universidad del Valle de México, en la Universidad Anáhuac del Norte y en la Escuela Militar de Ingenieros. Actualmente se desempeña como profesor de tiempo completo en el Centro Universitario Valle de México de la UAEM.