CAPITULO IV

SISTEMA DE RECONOCIMIENTO DE VOZ PARA PERSONAS EN CONDICIÓN DE DISCAPACIDAD MOTRIZ

Luis Guillermo Molero Suárez

Doctor en Ciencias Gerenciales. Unipanamericana Fundación Universitaria Panamericana Avenida (Calle) 32 No. 17-30, Teusaquillo, Bogotá, Colombia. Correo electrónico: lmolero@unipamericana.edu.co. <https://orcid.org/0000-0002-4117-8273>

Luis Ernesto Velasco Waltero

Ingeniero en Telecomunicaciones. Unipanamericana Fundación Universitaria Panamericana Avenida (Calle) 32 No. 17-30, Teusaquillo, Bogotá, Colombia. Correo electrónico: levelasco@unipanamericana.edu.co.

Jorge Alejandro Gaitán Albarracín

Ingeniero en Telecomunicaciones. Unipanamericana Fundación Universitaria Panamericana Avenida (Calle) 32 No. 17-30, Teusaquillo, Bogotá, Colombia. Correo electrónico: jalejandrogaitan@unipanamericana.edu.co.

Juan Camilo Valencia Hurtado

Ingeniero en Telecomunicaciones. Unipanamericana Fundación Universitaria Panamericana Avenida (Calle) 32 No. 17-30, Teusaquillo, Bogotá, Colombia. Correo electrónico: jcamilovalencia@unipanamericana.edu.co.

Luis Alejandro Tocarruncho Ariza

Ingeniero en Telecomunicaciones. Unipanamericana Fundación Universitaria Panamericana Avenida (Calle) 32 No. 17-30, Teusaquillo, Bogotá, Colombia. Correo electrónico: latocarruncho@unipanamericana.edu.co.

Resumen

La presente investigación tuvo como propósito desarrollar un sistema de reconocimiento de voz a escala para personas con discapacidad motora en beneficio de su inclusión social y laboral. De acuerdo con el estudio, la investigación fue de tipo aplicada, transversal y su diseño no experimental. Para lograr lo expuesto, en el contenido se analizó a detalle los siguientes puntos: 1) Sistemas de reconocimiento de voz y 2) Discapacidad motriz. El procedimiento utilizado fue determinar a través del desarrollo de un sistema domótico a escala los beneficios de esta técnica no convencional como instrumento de inclusión y acondicionamiento de espacios los cuales contribuyen a mejorar las condiciones de personas con discapacidad motriz. Los resultados

obtenidos cumplen con el objetivo propuesto previamente, permitiendo desarrollar un sistema de reconocimiento de voz a escala. Finalmente, se concluye que la tecnología de reconocimiento de voz impacta positivamente en la calidad de vida de pacientes en condiciones de discapacidad motriz.

Palabras clave: discapacidad motriz, domótica, inteligencia artificial, reconocimiento de voz.

VOICE RECOGNITION SYSTEM FOR PEOPLE WITH MOTOR DISABILITIES

Abstract

The purpose of this research was to develop a scale voice recognition system for people with motor disabilities to benefit their social and work inclusion. According to the study, the research was applied, cross-sectional, and non-experimental in design. To achieve this, the following points were analyzed in detail in the content: 1) Voice recognition systems and 2) Motor disability. The procedure used was to determine through the development of a home automation system at scale the benefits of this unconventional technique as an instrument for inclusion and conditioning of spaces which contribute to improving the conditions of people with motor disabilities. The results obtained meet the previously proposed objective, allowing the development of a voice recognition system at scale. Finally, it is concluded that voice recognition technology positively impacts the quality of life of patients in conditions of motor disability.

Keywords: artificial intelligence, home automation, motor disability, voice recognition.

Este capítulo es resultado de un proyecto de investigación intitulado “Desarrollo de Suite Interactiva con fines de rehabilitación para personas con movilidad reducida” el cual se encuentra finalizado.

Institución financiadora: Unipanamericana - Fundación Universitaria Panamericana bajo el Código del proyecto SIGP/Código entidad financiadora número PI242019.

Introducción

En la actualidad y desde siempre, los avances tecnológicos han permeado toda clase de disciplinas en búsqueda de la optimización de procesos y métodos que permitan el desarrollo de actividades de manera más productiva, logrando así, el perfeccionamiento de técnicas novedosas que han alcanzado a la cuarta revolución industrial conocida como industria 4.0.

Las tecnologías derivadas de la industria 4.0, han desplegado un sinfín de herramientas y soluciones que evidencian en algunos casos particulares, meritorios procesos de reingeniería vinculando prácticas tradicionales con poderosos sistemas para la solución de eventos modulares.

En adición a lo anterior, desde la última década se ha diversificado el uso de tecnologías inclusivas como son los sistemas domóticos, que permiten a través de una mixtura de equipos acondicionar espacios como oficinas, hogares, entre otros, para ofrecer de esta manera mejores condiciones de vida a las personas en condición de discapacidad. Existen claras evidencias como lo referencia Talbot, *et al.*, (2018); Parsch y Masrur, (2018), Korkmaz *et al.*, (2015); Joumaa, *et al.*, (2011); Louis *et al.*, (2015), que reflejan las ventajas de estos sistemas en pro del bienestar de estos grupos vulnerables, permitiendo a través de la optimización de áreas brindar comodidad y disminuir la brecha de movilidad.

Sobre la base de las evidencias expuestas, la investigación tuvo como propósito desarrollar un sistema de reconocimiento de voz a escala para personas con discapacidad motora en beneficio de su inclusión social y laboral, con el fin de brindar avances tecnológicos que impacten positivamente a este segmento de la población en aras de mejorar su calidad de vida e inclusión social y laboral.

Fundamentación teórica

Reconocimiento de voz

Los avances tecnológicos, han permitido dar un salto gigante en la evolución humana creando sistemas que permitan, incluso, entender lo que se está pronunciando. Es por ello, que para generar el reconocimiento de voz no es únicamente importante partir del deseo de interpretar los fonemas o sílabas, si no, de extraer las ideas que se estructuran a partir de las palabras, tal como hacen mención Esparza y Avalos, (2019).

En consecuencia, para De Mora *et al.*, (2013) “Los sistemas de reconocimiento de voz proporcionan un método de comunicación directa entre el hombre y el computador, para ello convierten el lenguaje humano hablado en un lenguaje máquina que entienda el computador.”. (p.289).

Por otra parte, expresa Monsó, (1994) que:

Un sistema de captura de datos mediante el reconocimiento de la voz puede ser descrito, de forma simple, como aquel que permite que la información sea introducida de forma automática en un sistema informático mediante el reconocimiento de la voz humana. (p. 139)

Asimismo, de acuerdo con Armas, (2019), el funcionamiento de un sistema de reconocimiento de voz se compone principalmente de: el extractor de características y el clasificador; cuando se recibe la señal de voz, ésta es tomada por un analizador el cual da como resultado la palabra que reconoce. Luego se detona el procesamiento al lenguaje natural, dando así una representación conocida y finalmente se realiza una acción.

En adición a lo anterior, existen tres tipos de enfoques de reconocimiento automático de voz, a saber: el enfoque acústico-fonético, el enfoque de patrones y el enfoque de la inteligencia artificial.

El enfoque acústico-fonético según Puertas (2000), corresponde a una técnica que tiene como fin interpretar y decodificar los sonidos que contiene la voz para asignar características

determinadas. Sin embargo, las propiedades de este enfoque no son altamente variables, pues depende del tono del locutor y la velocidad en la que se generan las secuencias fonéticas, asimismo, la identificación por comandos de voz presenta falencias en ambientes ruidosos e incluso con personas que presenten problemas en su voz.

Por otra parte, el enfoque de patrones es un método que decodifica la señal ingresada a partir de un grupo de señales reconocidas por el sistema, que permite, el reconocimiento de las características principales que generan el modelamiento de la información, es decir, que a partir del entrenamiento del sistema se puede reconocer y tomar la decisión. (Puertas, 2000). Finalmente el enfoque de la inteligencia artificial, es un método que utiliza las redes neuronales artificiales (RNA o ANN). Estas RNA, son sistemas que agrupan un conjunto de unidades neuronales conectadas entre sí, que permiten entrenarse a partir de un grupo de datos de manera aleatoria, a este proceso se le conoce como aprendizaje y permite su usabilidad en paralelo para el afinamiento, sin embargo, para su correcto uso y entrenamiento es necesario usar dispositivos con grandes capacidades de procesamiento (Puertas, 2000).

Para los efectos de la investigación, se desarrolló un sistema de hardware y software a escala, también conocido como domótica, capaz de automatizar los espacios concernientes a un conjunto de habitaciones donde se implemente la apertura y cierre de puertas, ventanas y el encendido y apagado de luminarias con reconocimiento por instrucciones de voz a través del enfoque de inteligencia artificial.

Discapacidad motriz

La discapacidad motriz se define como la alteración de la capacidad del movimiento que se presenta en distintos grados disminuyendo la función de desplazamiento de la persona o de manipulación, que limita al individuo en su desarrollo personal y social, pudiendo ser congénita o adquirida, siendo esta última, como consecuencia de lesiones, accidentes, o enfermedades que afectan el sistema musculoesquelético (Saavedra *et al.*, citado por Molero *et al.*, 2019).

Es de especial interés definir los criterios que envuelven la discapacidad motriz a los fines de desarrollar un sistema domótico a través de una colección de dispositivos de hardware y software de control a manera de mejorar sustancialmente la calidad de vida y autonomía de personas con este tipo de deficiencias.

Metodología

La investigación se sustenta metodológicamente bajo el paradigma positivista con enfoque cuantitativo. Al respecto, González y Rodríguez, (2011), afirman que, desde el punto de vista metodológico, se suele denominar cuantitativa a la investigación que, predominantemente tiende a usar instrumentos de medición, desarrollo de sistemas y diseños de prototipos. Como es el caso específico del estudio, el cual trata sobre desarrollar un sistema de reconocimiento de voz a escala para personas con discapacidad motora en beneficio de su inclusión social y laboral.

Con respecto al conocimiento científico generado como resultado del estudio corresponde a la afirmación positiva de las teorías de Sistema de reconocimiento de Voz y Discapacidad Motriz, a través de la aplicación del método científico. Es por ello, que el conocimiento generado busca explicar causalmente los fenómenos indagados por medio de las teorías ya mencionadas, y por tanto se concibe a la razón como medio para otros fines; es decir, una razón instrumental. (Perez, 2010).

Por otra parte, analizando el objetivo de esta investigación, se describe como de tipo aplicada, por cuanto busca a través de los resultados derivados de la opinión de los expertos, según Tamayo y Tamayo, (2006) señalan que la investigación aplicada, es aquella donde se aprovechan realidades concretas en los resultados de investigación. En ese sentido, el presente trabajo se define como aplicado, ya que se propuso solucionar un problema específico con el empleo de herramientas tecnológicas que permitan desarrollar un sistema de reconocimiento de voz a escala para personas en condición de discapacidad motriz. Según el criterio de Chavez, (2007, p. 45) “la investigación aplicada tiene como fin resolver problemas en un tiempo corto”.

Si bien es cierto, los diseños de investigación vienen hacer el proceso mediante el cual, se planifica la investigación, es decir donde se exponen los aspectos sobre el cual se pretende crear conocimiento, a través de una situación dada, para Mendez, (2006), el diseño de investigación significa el criterio utilizado por el investigador para proyectar un futuro deseado.

Por otra parte, para Arias, (2006), el diseño de investigación es la estrategia adoptada por el investigador para responder al problema planteado, es decir, el procedimiento concebido para obtener la información deseada. Sin embargo, el diseño señala al investigador lo que debe hacer para alcanzar sus objetivos de estudio, a fin de dar respuesta a las interrogantes de conocimiento planteado. La investigación de acuerdo con su diseño es no experimental, en virtud de que involucra procesos sistemáticos de búsqueda e indagación a través de una descripción, una comparación, un análisis y una predicción para que finalmente se diseñe una propuesta; ya que pretende caracterizar las variables; coincidiendo con lo expuesto por Hernández *et al.*, (2010).

En la misma se simula un diseño, por consiguiente, las variables no se manipulan deliberadamente, es decir, no se promueve el cambio intencionado de las variables objeto de estudio, ni se manipulan variables ni hipótesis, no se controla nada.

De igual manera, dicha investigación se clasificó dentro del tipo transversal descriptiva, debido a que la recolección de los datos de interés para verificar la autenticidad de la información sobre las variables en estudio se establecerá según los criterios del investigador y en el lapso programado para ello. Según Hernández *et al.*, (2010), dichos estudios recolectan datos en un solo momento, en un tiempo único, teniendo como propósito describir variables y analizar su incidencia e interrelación en un momento dado.

Según el método de estudio, y según lo expresado por Chavez, (2007, p. 135), la investigación de tipo descriptiva, “es aquella que se orienta a recolectar información relacionada con el estado real de las personas, objetivos, situaciones o fenómenos, tal cual como se presentaron en el momento de su recolección, describe lo que se mide sin realizar

inferencias ni verificar hipótesis”. Según Tamayo y Tamayo, (2006, p. 46), este estudio puede tipificarse, en cuanto al nivel de conocimiento adquirido como descriptiva, ya que se trabaja sobre una realidad desde el hecho, y su característica fundamental es la de presentar una interpretación correcta, en este caso, se caracteriza y describen las variables de estudio para abordar el diseño que se percibe.

Atendiendo a estas consideraciones, el propósito de la presente investigación reúne el desarrollar un sistema de reconocimiento de voz a escala para personas con discapacidad motora en beneficio de su inclusión social y laboral, esto, a manera de estrategia no convencional, estableciendo una estrategia de transformación digital desde una nueva perspectiva de cultura tecnológica (Molero *et al.*, 2019).

Por otra parte, para la implementación del sistema de reconocimiento de voz a escala para personas con discapacidad motriz, se hace uso de la metodología tecnológica (Cegarra, 2004), la cual consta de 5 fases, a saber: Identificación, Exploración, Diseño, Desarrollo y Evaluación.

Finalmente, la investigación se llevó a cabo alrededor de un estudio de tipo aplicado, transversal y su diseño no experimental empleando las categorías Sistemas de reconocimiento de voz y Discapacidad Motriz y en consecuencia, poder enmarcar de forma práctica la justificación del uso de los sistemas de reconocimiento de voz como novedosa herramienta para establecer ambientes inclusivos en pacientes con discapacidad motriz.

Resultados

En este apartado se describe el proceso desarrollo del producto final de acuerdo a sus fases, que se expone a continuación:

Identificación

En esta fase se observó y reflexionó sistemáticamente sobre las realidades teórico/prácticas y empíricas usando diferentes tipos de documentos, datos y censos, con el fin de determinar algunas necesidades de la población con discapacidad motriz y de esta manera determinar una solución real. Permitiendo concluir, que la sistematización de espacios reduciendo la interacción física con los mismos, ofrecen mayor comodidad e inclusión a este grupo vulnerable permitiendo mejorar su calidad de vida.

Exploración

En este apartado, se realizó una búsqueda de información profunda, se analizaron alternativas tecnológicas apropiadas para personas con discapacidad motriz a fin de suplir las necesidades identificadas. En ese sentido, se concluyó que la opción eficaz para la automatización de una locación pasaba por el desarrollo de un sistema domótico a escala para a través de comando de voz tener la capacidad de operar puertas, ventanas y luminarias sin mayor esfuerzo.

Diseño

A continuación, se muestra la especificación de diseño del software y hardware que permitió crear la interfaz de usuario y el sistema domótico. Estos sustentos fueron la base principal para el desarrollo del producto.

Diseño de Software

En ese orden de ideas, la interfaz de programación utilizada fue la herramienta “App Inventor 2” (AI2), que es un entorno de desarrollo de software creado por Google y administrado por el Instituto de Massachussets (MIT) que permite crear aplicaciones por medio de bloques de manera intuitiva y gráfica. La Interfaz de usuario será diseñada para dispositivos con sistema operativo “Android” en versiones superiores a la versión 6.0, pues a partir de esta versión se cuenta con los servicios de “Google Assistant”, herramienta de inteligencia artificial que responde de forma óptima a diversas situaciones sin recurrir a grandes fuentes de procesamiento, asimismo, mejora significativamente la incidencia del ruido ambiental, el empleo de grandes dispositivos para el procesamiento de órdenes de voz y reduce los costos asociados a la adquisición de licencias de software de terceros.

Asimismo, para el desarrollo de la aplicación se hizo uso del lenguaje “Kawa” que está integrado en el editor de bloques de la herramienta AI2, esta herramienta es distribuida igualmente por el MIT bajo licencia libre y puede ser accedida desde su portal web <https://appinventor.mit.edu/>. Asimismo, esta herramienta es soportada por la librería de acceso gratuito “Blockly” de “JavaScript”, que permite desarrollar a partir de códigos de bloques visual y está integrada a la herramienta AI2.

Para efectos del sistema domótico, se pretendió que las personas con discapacidad motriz tengan la posibilidad de controlar objetos básicos del hogar a través de instrucciones de voz mediante el uso de un smartphone “Android” conectado a la red local WIFI. Para el objetivo propuesto, se diseñó una aplicación que se conectará al sistema, compuesta de las siguientes pantallas:

1. Pantalla de inicio transitoria
2. Bienvenida
3. Menú de opciones
4. Menú de sistema de reconocimiento de voz
5. Sistema de respaldo y,
6. Serie de funciones preestablecidas.

En la tabla 1, se define la funcionalidad de cada una de las pantallas:

Tabla 2.
Pantallas del aplicativo.

Pantalla	Funcionalidad
Pantalla de Inicio Transitoria	Llevará el nombre de la aplicación y el signo distintivo que identifica al proyecto de investigación, donde luego de transcurrir de dos segundos, abrirá la pantalla de “Bienvenida”
Bienvenida	Llevará el signo distintivo “R3DVirtual” y tres opciones: “Iniciar”: permite al usuario ingresar al menú de opciones. “Información acerca de la aplicación”: permite al usuario conocer acerca de la aplicación, usabilidad y beneficios. “Cerrar aplicación”: permite al usuario cerrar la aplicación de forma permanente.
Menú de opciones	Corresponde al menú de opciones principal, que permite elegir tres opciones: “Reconocimiento de voz”: permite al usuario ingresar al menú de reconocimiento de voz. “Sistema de respaldo”: permite al usuario ingresar a la pantalla de encender luminarias. “Regresar”: permite al usuario regresar a la pantalla de bienvenida.
Menú sistema de reconocimiento de voz	Corresponde al menú de opciones, que tiene cinco alternativas, a saber: “Encender luminarias” “Apagar luminarias” “Abrir accesos” “Cerrar accesos” “Regresar” Al oprimir cada una de estas opciones dirigirá al usuario a cada pantalla respectivamente.
Serie de funciones preestablecidas	La serie de funciones preestablecidas corresponde a las frases que la persona con discapacidad motriz deberá pronunciar, estas opciones están desarrolladas bajo comandos de voz, se activaran al escuchar del usuario el comando hablado, este, convertirá sonido hablado en texto usando para ello la función de reconocimiento de voz de “Android”, junto a un componente web que proporcione funciones para solicitudes del Protocolo de Transferencia de Hipertexto (HTTP), GET (recuperar cualquier información), POST (enviar información al servidor), PUT (almacenar entidad en una ubicación específica) y DELETE (eliminar información de una ubicación específica), adicionalmente cada una de estas opciones permitirá la elección de regresar a la pantalla anterior.
Sistema de respaldo	Esta pantalla es tal vez uno de los aspectos más importantes del aplicativo, en virtud, de que mantiene activa la aplicación aun si se presentan fallos de conectividad. Esta opción, se conectará al sistema domótico sin necesidad de estar conectado a Internet y en caso de que falle la conexión o los servidores de “Google assistant” para el reconocimiento de voz, esta pueda ser empleada con solo imprimir una opción como “Encender luz baño”.

Fuente: elaboración propia (2019).

Diseño del hardware

Concerniente a este punto, se diseñaron las especificaciones del sistema embebido que se conectará a la red del hogar para el control de luminarias y accesos. En ese sentido, se planteó el uso de conexiones inalámbricas WIFI debido al amplio espectro de red que ofrece, las facilidades de instalación y su fácil interacción con el entorno para proporcionar la conectividad entre los dispositivos que permitan la comunicación del sistema domótico, evitando el uso de medios guiados (cables) que pudiesen aumentar el costo de la solución. De la misma forma, se hará uso de un Microcontrolador ESP32 que cuenta con dos módulos integrados: bluetooth y recepción inalámbrica, esto con la finalidad de optimizar espacios dentro del sistema domótico e integración con los lenguajes de programación conocidos. A continuación, se muestra el Diagrama de bloques y el esquemático de la red.

Diagrama de bloques

Este diagrama, aborda la definición funcional del sistema domótico haciendo uso de comandos de voz para realizar las peticiones, estas, se envían a la aplicación desarrollada haciendo uso de un smartphone con sistema operativo “Android” versión 6 o superiores, donde se utilizan peticiones GET que permiten recuperar el estado de una información, esto, con la finalidad de obtener en “texto” el mensaje transmitido en la señal analógica ingresada por el usuario y finalmente, la información obtenida es conducida a través de la red WIFI del hogar.

Luego de haber obtenido el mensaje en “texto”, se procede a realizar el envío de la orden al Microcontrolador ESP32, usando el módulo WIFI propio de este controlador. De acuerdo con la programación previamente establecida sobre el controlador, este sabrá reconocer a qué salida debe ubicar la acción solicitada.

A continuación, en la figura 1, se presenta el diagrama de bloques de funcionamiento de los dispositivos:

Esquemático de la red

En este esquemático de red para comunicaciones M2P, (por siglas en ingles de machine to Person) se utilizó una topología de red tipo estrella, en esta topología existe un nodo principal donde convergen todas las conexiones de red. Para el desarrollo del sistema domótico, el nodo principal es el Microcontrolador ESP32 el cual es el encargado de ejecutar las acciones solicitadas o las acciones preestablecidas, las cuales son: “Encender iluminarias”, “Apagar iluminarias”, “Abrir accesos” y “Cerrar accesos”, dejándolas a disposición del usuario.

Por otra parte, el “Servidor Web” publicado en el Microcontrolador opera mediante el protocolo HTTP, al cual se le asigna el puerto TCP 80 donde las peticiones al servidor para este sistema se realizan mediante HTTP utilizando el método de petición GET (recuperación de datos), en donde la función a realizar se solicita a través de la URL al “Servidor Web”. En la figura 2, se muestra el modelo cliente-servidor a utilizar:

Figura 1

Diagrama de bloques de funcionamiento de los dispositivos

Fuente: elaboración propia (2019).

Figura 2

Cliente-servidor HTTP

Fuente: elaboración propia (2019).

Durante el desarrollo del sistema, se estableció que el Microcontrolador ESP32 será el dispositivo que hará las veces de “Servidor Web” y recibirá las peticiones enviadas a través del reconocimiento de órdenes de voz integrado en la aplicación, así mismo, se encargará de procesar dicha petición, dar respuesta, y enviar las señales digitales a los actuadores.

De acuerdo con lo anterior, se presenta así el Microcontrolador ESP32 actuando como nodo central, encargado de enviar las solicitudes requeridas por el usuario, comunicándose

de esta forma con cada actuador (Servomotores, Diodo leds) del sistema, por lo tanto, si algún terminal presenta un fallo o pierde comunicación con el Microcontrolador, esto no afectará el resto de los componentes del sistema domótico, asimismo, el Router tendrá la función de servir como conexión entre el Smartphone y el Microcontrolador y lograr la comunicación final con cada elemento de la red. Seguidamente, en la figura 3 se presenta el diagrama de la topología estrella del sistema domótico:

Figura 3

Topología estrella sistema domótico

Fuente: elaboración propia (2019).

El Microcontrolador ESP32 cuenta con módulo WIFI, por lo cual, se realizaron pruebas con la librería “WiFi.h” del IDE de Arduino. Para el uso de esta librería, se planteó que el controlador tomara su IP estática o a través de DHCP (Protocolo de Configuración IP Dinámicas), obteniendo sus capacidades de “Servidor Web”, es decir, el sistema debe reconocer hacia qué red se está conectando y debe identificar la IP del servidor que fue asignado para que el software detecte y sincronice con el sistema. Una vez conectado el módulo a la red WIFI, se verificaron los puertos CDA del Microcontrolador a fin de crear el código que permita realizar las funciones preestablecidas para el sistema embebido.

Seguidamente, se verificó el Pinout del Microcontrolador ESP32 SoC para determinar cuáles pueden ser usados por los actuadores. A continuación, en la figura 4 se muestra el diagrama de Pinout del microprocesador ESP32:

Figura 4
Pinout ESP32

Fuente: Espressif System (2019).

Análogamente, se mencionan los pines requeridos del Microcontrolador ESP32 para el sistema, cabe resaltar, que este Microcontrolador cuenta con una fuente integrada de 3,3 v, sin embargo, no se tiene en cuenta para el planteamiento de esta propuesta debido a que un servomotor requiere como mínimo ese voltaje y esto no permite un movimiento adecuado, además, las funciones establecidas en el pseudocódigo que son usados para los actuadores ya están definidos, Servomotores y Leds. Posteriormente, en la tabla 2 se muestran los Pin requeridos por el prototipo:

Tabla 3
Pinout ESP32.

PIN ESP32 REQUERIDOS PARA EL PROTOTIPO		
PIN	FUNCIÓN	TERMINAL
Pinout 2	Al recibir la petición del puerto serial se escribe 1 o 0 según el dato recibido, el cual enciende o apaga la luz del baño.	LED
Pinout 4	Al recibir la petición del puerto serial se escribe 1 o 0 según el dato recibido, el cual enciende o apaga la luz de la habitación.	LED
Pinout 12	Al recibir la petición del puerto serial se escribe una posición o grado desde 0 a 90 según el dato recibido, este permite abrir o cerrar la puerta de la habitación girando en dos direcciones.	SERVOMOTOR
Pinout 13	Al recibir la petición del puerto serial se escribe una posición o grado desde 0 a 90 según el dato recibido, este permite abrir o cerrar la puerta del baño girando en dos direcciones.	SERVOMOTOR
Pinout 15	Al recibir la petición del puerto serial se escribe 1 o 0 según el dato recibido, el cual enciende o apaga la luz de la cocina	LED
Pinout 32	Al recibir la petición del puerto serial se escribe una posición o grado desde 0 a 90 según el dato recibido, este permite abrir o cerrar la puerta de la entrada girando en dos direcciones.	SERVOMOTOR
Pinout 33	Al recibir la petición del puerto serial se escribe una posición o grado desde 0 a 90 según el dato recibido, este permite abrir o cerrar la ventada girando en dos direcciones.	SERVOMOTOR

Fuente: elaboración propia (2019).

Es importante destacar, que para la construcción del pseudocódigo se plantea que por cada frase preprogramada el controlador reciba un código a través del reconocimiento de órdenes de voz de Android, dicho código será traducido por la programación realizada desde la creación de la aplicación. A continuación, se presentan los componentes de hardware del modelo:

Inicialmente, en la figura 5, se muestra el Microcontrolador ESP32 SoC (System on Chip) que integra tecnologías de bluetooth y WIFI para proporcionar conectividad a Internet y otros dispositivos. Es un módulo de bajo costo, ideal para el desarrollo de aplicaciones enfocadas a Internet de las Cosas (IoT).

Del mismo modo, en la figura 6, se muestra el Servo motor SG90. Este motor eléctrico, permite controlar la velocidad del giro, programando el tiempo para efectuar el mismo que está en el rango de trabajo de 1 mili segundo y 2 milisegundo. Por otra parte, el IDE de Arduino cuenta con una librería (servo.h) que permite programar un giro de -90° y 90° lo que viene siendo un ángulo de giro de 180°, es por esto, que el dispositivo se implementó en el modelo para simular la apertura y cierre de puertas y ventanas.

Figura 5
ESP32 SoC (System on Chip)

Fuente: https://www.banggood.com/ESP32-Development-Board-WiFibluetooth-Ultra-Low-Power-Consumption-Dual-Cores-ESP-32-ESP-32S-Board-p-1109512.html?cur_warehouse=CN

Figura 6
Servo motor SG90

Fuente: <https://electronilab.co/tienda/micro-servo-9g-towerpro/>

Posteriormente, en la figura 7, se muestra el Diodos LED quien emite una luz incandescente perfecto para simular el encendido de las luces en el modelo, de igual manera, puede ser utilizado en el hogar para la persona con discapacidad motriz ya que es un dispositivo de bajo consumo con una vida útil aproximada de 100.000 horas.

Figura 7
Diodos LED

Fuente: <https://electronilab.co/tienda/led-blanco-5mm-chorro/>

De forma semejante, en la figura 8, se muestra un modelo de smartphone quien cuenta con el sistema operativo Android versión 6 o superiores. Esta versión del sistema operativo permite que las aplicaciones que quieran acceder al micrófono, GPS o cámara del dispositivo puedan solicitarlo para hacer uso de estas funcionalidades. Asimismo, el teléfono inteligente debe contar con conexión WIFI.

Figura 8
Smartphone

Fuente: <https://laopinion.com/guia-de-compras/huawei-p30-vs-iphone-x-cual-es-el-mejor-celular-para-ti/>

Posteriormente, en la figura 9, se muestra un cargador de uso comercial que brindará voltaje regulado al sistema. Este adaptador trabaja con 3.3 a 5 V-DC ideal para que el sistema trabaje con normalidad, cabe mencionar, que si trabaja a más voltaje de lo que se está indicando puede quemar los dispositivos internos del modelo.

Figura 9
Adaptador Samsung EP-TA200

Fuente: <https://www.samsung.com/es/mobile-accessories/travel-adapter-micro-usb-ta20/>

Desarrollo

Una vez diseñado el sistema, se procedió a desarrollar el software y el hardware del sistema domótico. A continuación, se presentan las actividades que amalgamaron todo el desarrollo:

Desarrollo del Software

De acuerdo con los elementos mencionados en el diseño del software de control, se precisa la síntesis del desarrollo del mismo que servirá de interfaz entre el usuario y el sistema domótico, para ello, se explicará a detalle el desarrollo del FrontEnd, el desarrollo del BackEnd y las consideraciones de seguridad en las comunicaciones entre dispositivos.

Desarrollo del FrontEnd

En este apartado, se define cómo es la presentación del aplicativo con respecto al usuario, para ello, se hará uso de la programación visual (Scratch).

Pantalla de inicio transitoria

Descripción

A continuación, en la figura 10 denominada “Pantalla de inicio transitoria” de la aplicación, se muestra la imagen “PreVoice” correspondiente al nombre del aplicativo y en la parte inferior el signo distintivo “R3D Virtual” registrado del proyecto de investigación. Esta pantalla permanece activa por dos segundos una vez ejecutado el aplicativo y luego cambia a la siguiente pantalla.

Figura 10

Pantalla de inicio transitoria

Fuente: elaboración propia (2019).

Código de bloque

Descripción

El código de bloque que se muestra en la figura 11, corresponde a la “pantalla de inicio de transitoria” (Screen 1), que hace uso de una condicional simple, dicha condicional: Al iniciar la pantalla de inicio transitoria (Screen 1) llama al sonido (Sound1) y envía un valor verdadero (1) al temporizador habilitado (Clock1), entonces cuando el temporizador (Clock1) cumpla el tiempo establecido (2 Segundos) desde la interfaz de usuario (FrontEnd), abrirá la pantalla de “Bienvenida”.

Figura 11
Blockly de Inicio transitoria

Fuente: elaboración propia (2019)

Bienvenida

Descripción

Seguidamente, en la figura 12, denominada pantalla de “Bienvenida”, viene contentiva del signo distintivo “R3D Virtual” y tres opciones, a saber: “Información”, “Iniciar” y “Cerrar la aplicación”.

Figura 12
Screen de Bienvenida

Fuente: elaboración propia (2019).

Código de bloque

Descripción

El código de bloque que se muestra en la figura 13, correspondiente a “Bienvenida”, hace uso de condicionales simples, dichas condicionales corresponden a: cuando se oprima el botón “Iniciar” abrirá la pantalla “Menú de opciones”, cuando se oprima el botón “Información” abrirá la pantalla “Acerca de prevoice” y cuando se oprima el botón “Cerrar aplicación” cerrará la aplicación.

Figura 13

Blockly de Bienvenida

Fuente: elaboración propia (2019).

Acerca de prevoice

Descripción

La figura 14, muestra la pantalla “Acerca de prevoice”, esta permite al usuario conocer de qué trata el sistema, para qué sirve y qué puede hacer con ella.

Esta pantalla no contiene código de bloque, puesto que su intención es solo brindar información acerca de la aplicación, por lo que todo se realiza desde la parte de diseño. Para ir al menú anterior (bienvenida) se realiza a través del botón back de cada smartphone “Android”.

Menú de opciones

Descripción

La pantalla “Menú de opciones” que se muestra en la figura 15, es la pantalla luego de oprimir la opción “Iniciar” que muestra tres opciones para la interacción con el sistema: “Reconocimiento de voz”, “Sistema de respaldo” y “Regresar”.

Figura 14

Screen Acerca de prevoice

Fuente: elaboración propia (2019).

Figura 15

Screen Menú de opciones

Fuente: elaboración propia (2019).

Código de bloque

Descripción

El código de bloque que se muestra en la figura 16, correspondiente a “Menú de opciones”, que hace uso de condicionales simples, dichas condicionales corresponden a: cuando se oprima el botón “Reconocimiento de voz”, abrirá la pantalla “Menú sistema de reconocimiento de voz”, cuando se oprima el botón “Sistema de respaldo” abrirá la pantalla “Sistema de respaldo” y cuando se oprima el botón “Regresar”, regresará a la pantalla de “Bienvenida”.

Figura 16

Blockly Menú de opciones

Fuente: elaboración propia (2019).

Menú sistema de reconocimiento de voz

Descripción

Seguidamente, la figura 17, muestra el “Menú sistema de reconocimiento de voz” con los botones para ejecutar el “Encendido” o “Apagado” general de luminarias y la “Apertura” o “Cierre” de todos los accesos. Al oprimir alguno de estos botones, dirigirá al menú “Serie de funciones preestablecidas”, donde el usuario inicia la interacción con comandos de voz.

Código de bloque

Descripción

El código de bloque que se muestra en la figura 18 correspondiente a “Menú sistema de reconocimiento de voz”, hace uso de condicionales simples, dichas condicionales corresponden a: cuando se oprima los botones para hacer un “Encendido” o “Apagado” general de luminarias, o la “Apertura” y “Cierre” de todos los accesos. Por cada opción seleccionada, abrirá una pantalla independiente de acuerdo a la serie de funciones preestablecidas.

Figura 17

Menú Sistema de reconocimiento de voz

Fuente: elaboración propia (2019).

Figura 18

Blockly Menú de Reconocimiento de voz

Fuente: elaboración propia (2019).

Serie de funciones preestablecidas

Descripción

Esta interfaz permite al usuario enviar las órdenes por medio de comandos de voz al sistema domótico para la opción que desea realizar. Inicialmente, la interfaz muestra el

mensaje “¿Qué desea encender?”, es cuando, se oprime el logo de “Google Assistant” y se pronuncia cualquier texto expuesto en la pantalla (“luz cocina”, “luz baño”, “luz habitación”) y el sistema domótico se encargará de responder al comando de voz en tiempo real. En la figura 19, se muestra la pantalla de “funciones preestablecidas”:

Figura 19

Funciones preestablecidas encendido de iluminarias (reconocimiento de voz)

Fuente: elaboración propia (2019).

Código de bloque

Descripción

Los códigos de bloque que se muestran en las figuras 20, 21, 22 y 23 corresponden a “Serie de funciones preestablecidas”. Este código hace uso de condicionales simples, dichas condicionales corresponden a: cuando se oprime el botón “Reconocimiento de voz” (Logo de “Google assistant”) llamará al reconocimiento de voz de “Android” y obtendrá el texto, si el texto corresponde a una de las opciones o funciones preestablecidas (“luz cocina”, “luz baño”, “luz habitación”), entonces enviará a través de la web o URL una petición GET al servidor HTTP o ESP32.

IMPORTANTE: Cada texto, frase o función preestablecida tiene un código asociado a la URL para que a través de la petición GET el Microcontrolador ESP32 pueda reconocer qué función deba realizar.

Figura 20

Blockly Funciones preestablecidas encendido de iluminarias (reconocimiento de voz)

Fuente: elaboración propia (2019).

Figura 21

Blockly Funciones preestablecidas apagado de iluminarias (reconocimiento de voz)

Fuente: elaboración propia (2019).

Figura 22

Blockly Funciones preestablecidas apertura de accesos (reconocimiento de voz)

Fuente: elaboración propia (2019).

Figura 23

Blockly Funciones preestablecidas cierre de accesos (reconocimiento de voz)

Fuente: elaboración propia (2019).

Sistema de respaldo

Descripción

La interfaz que se muestra en la figura 24, permite al usuario enviar las ordenes con solo oprimir alguna opción del sistema domótico sin necesidad de estar conectado a Internet y, en caso de fallas en la conexión a Internet o los servidores de “Google assistant” para el reconocimiento de voz, este pueda ser empleado con solo imprimir una opción, por ejemplo: “Encender luz baño”, en donde se envía la petición HTTP GET al ESP32 y este envía la señal al actuador correspondiente.

Figura 24

Screen Sistema de respaldo

Fuente: elaboración propia (2019).

Código de bloque

Descripción

Los códigos de bloque que se muestran en las figuras 25 y 26 correspondiente a “Sistema de respaldo”, hacen uso de condicionales simples, dichas condicionales corresponden a: cuando se oprima las opciones “Encender luz” (amarillo), “Apagar luz” (Blanco), “Abrir puerta” (Gris), “Cerrar puerta” (Rosado), “Abrir/Cerrar Ventana” (Azul) o “Regresar” (Amarillo más oscuro), se enviará a través de la web o URL una petición GET al servidor http o ESP32.

IMPORTANTE: Cada texto, frase o función preestablecida tiene un código asociado a la URL para que a través de la petición GET el Microcontrolador ESP32 pueda reconocer qué función deba realizar.

Figura 25

Blockly Control de remoto cierre y apertura de accesos

Fuente: elaboración propia (2019).

Figura 26

Blockly Control remoto encendido y apagado de iluminarias

Fuente: elaboración propia (2019).

Consideraciones de seguridad en las comunicaciones entre dispositivos.

Con el fin de garantizar la seguridad del usuario que opera el sistema domótico, las aplicaciones que interactúan entre los dispositivos se construyeron a partir del principio de permisos mínimos, con la finalidad de ir creciendo en requerimientos de acuerdo con la necesidad presentada.

Los principios de diseño aplicados para la aplicación de servidor son:

- Permitir únicamente la conexión al servidor expuesto por el Microcontrolador, desde una red específica. En esta red previamente se configuró la reservación de la dirección IP asignada al controlador, adicional a ello, en la lógica de la configuración del Microcontrolador, se inhabilitó las conexiones diferentes al puerto 80, y, por último, se configuró la denegación de todas las peticiones no reconocidas.
- Desde el Router encargado de proveer la red WIFI donde convergen los dispositivos, se realizó un filtrado de direcciones MAC para asegurar a los dispositivos que están incluidos en la red, adicional, se habilitó la opción de redireccionamiento de puertos con la finalidad de asegurar que por la red únicamente viajaran peticiones conocidas.
- Por último, la aplicación que se instalará en el smartphone se entregará personalizada de acuerdo con los requerimientos anteriores y únicamente se instalará al usuario que manipule el sistema domótico.

Desarrollo del BackEnd

A continuación, se presenta el código de BackEnd que corresponde al Microcontrolador ESP32.

Inicialmente, se definieron las librerías compatibles con la tarjeta de desarrollo del Microcontrolador ESP32 con el fin de tener control de los diferentes actuadores (Servomotores y diodos LED), teniendo en cuenta lo anterior, se analizó el Pinout del Microcontrolador ESP32 y posteriormente se procedió a programarlo con el IDE de Arduino, considerando, su función de “Servidor Web”. Seguidamente, en la figura 27, se describe la programación implementada en el Microcontrolador.

Se definió la librería “WiFi.h”, disponible desde la página de (GitHub, 2019) y se instalaron en la aplicación de Arduino para trabajar con este entorno de desarrollo integrado. Esta librería, permite la conexión del módulo WIFI a la red WIFI del usuario para convertirse en un “Servidor Web”, para lo cual, se realizó la programación del Microcontrolador ESP32 SoC (figura 28). Seguidamente, se verificó a través del monitor serial (permite enviar y recibir datos) que el Microcontrolador ESP32 haya tomado la dirección IP (Internet Protocol) y como último paso, se verificó desde la web, la conexión con el Microcontrolador y la capacidad de poder administrar desde otro equipo haciendo la petición HTTP GET.

Figura 27

Configuración de conexión con el módulo WIFI

```
config conexion modulo wifi esp32.txt
1
2 // library is downloaded from IDE Arduino
3 #include <WiFi.h>
4
5 const char* ssid    = "TERABW";
6 const char* password = "qwerTyuiop95..";
7
8
9 IPAddress local_IP(192, 168, 0, 2);
10 IPAddress gateway(192, 168, 0, 1);
11 IPAddress subnet(255, 255, 255, 0);
12 IPAddress primaryDNS(8, 8, 8, 8); //optional
13 IPAddress secondaryDNS(8, 8, 4, 4); //optional
14
15 //config tarjeta esp32
16
17
18
19 WiFiServer server(80);
20 void setup()
21 {
22   if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
23     Serial.println("STA Failed to configure");
24   }
25
26   Serial.print("Connecting to ");
27   Serial.println(ssid);
28
29   WiFi.begin(ssid, password);
30
31   while (WiFi.status() != WL_CONNECTED) {
32     delay(500);
33     Serial.print(".");
34   }
35
36   Serial.println("");
37   Serial.println("WiFi connected!");
38   Serial.print("IP address: ");
39   Serial.println(WiFi.localIP());
40   Serial.print("ESP Mac Address: ");
41   Serial.println(WiFi.macAddress());
42   Serial.print("Subnet Mask: ");
43   Serial.println(WiFi.subnetMask());
44   Serial.print("Gateway IP: ");
45   Serial.println(WiFi.gatewayIP());
46   Serial.print("DNS: ");
47   Serial.println(WiFi.dnsIP());
48
49   server.begin();
50 } //sample connection and addressing
```

Fuente: elaboración propia (2019).

Figura 28

Líneas de conexión WIFI

```
config conexion modulo wifi esp32.txt  Líneas de con
1  Líneas de conexión wi fi
2
3 #include <WiFi.h>
4 #include <Servo.h>
5 const char* ssid    = "Nombre de red";
6 const char* password = "Contraseña de red";
7
8
9 IPAddress local_IP(192, 168, 0, 2);
10 IPAddress gateway(192, 168, 0, 1);
11 IPAddress subnet(255, 255, 255, 0);
12 IPAddress primaryDNS(8, 8, 8, 8);
13 IPAddress secondaryDNS(8, 8, 4, 4);
14
15 WiFiServer server(80);
16
```

Fuente: elaboración propia (2019).

Consecutivamente se definió la librería “Servo.h”, disponible desde la página de (GitHub, 2019), para controlar los servomotores junto con los pines de salida y estados actuales en el “void setup” (función que se ejecuta en el momento que el programa se inicia) para los actuadores. Posteriormente, se inició las sentencias de salida para la toma de datos a través del “void loop” (función que ejecuta el programa en bucle).

Finalmente, se realizaron las pruebas funcionales de Apertura/Cierre de puertas y ventanas y Encendido/Apagado de luminarias por medio de la conexión WIFI, observando que las salidas definidas actuaron de forma satisfactoria.

En la figura 29 y 30, se definieron las acciones del Microprocesador, los servomotores y la forma como estos van a usar sus pines para las entradas y salidas; la velocidad de conexión es de 115200 baudios.

En la figura 31, se definió la conexión entre hardware, el software y la red WIFI para que los elementos interactúen sin que se vaya a liberar la conexión. Este, es el centro de unión del dispositivo Android, el Router WIFI, el software y la conexión de red.

Figura 29

Líneas de activación de entradas al microprocesador

The image is a screenshot of a code editor window. The title bar shows 'config conexion modulo wifi esp32.txt' and 'Líneas de conexión wi fi'. The code is in C++ and defines four servos, sets up pins, and initializes a serial connection. The code is as follows:

```
1  Líneas de activación de entradas al microprocesador
2
3  static const int servo1Pin = 12;
4  static const int servo2Pin = 13;
5  static const int servo3Pin = 32;
6  static const int servo4Pin = 33;
7
8  Servo servo1;
9  Servo servo2;
10 Servo servo3;
11 Servo servo4;
12
13 void setup()
14 {
15     Serial.begin(115200);
16     pinMode(2, OUTPUT);
17     pinMode(15, OUTPUT);
18     pinMode(4, OUTPUT);
19     servo1.attach(servo1Pin);
20     servo2.attach(servo2Pin);
21     servo3.attach(servo3Pin);
22     servo4.attach(servo4Pin);
23
24     delay(10);
25 }
```

Fuente: elaboración propia (2019).

Figura 30

Líneas de conexión de dispositivo y red WIFI. Capa III entre dispositivos

```

1  Líneas de conexión dispositivo y red wi-fi capa uno entre dispositivos
2
3  Serial.begin(115200);
4  if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
5      Serial.println("STA Failed to configure");
6  }
7  Serial.print("Connecting to ");
8  Serial.println(ssid);
9  WiFi.begin(ssid, password);
10 while (WiFi.status() != WL_CONNECTED) {
11     delay(500);
12     Serial.print(".");
13 }
14
15 Serial.println("");
16 Serial.println("WiFi connected!");
17 Serial.print("IP address: ");
18 Serial.println(WiFi.localIP());
19 Serial.print("ESP Mac Address: ");
20 Serial.println(WiFi.macAddress());
21 Serial.print("Subnet Mask: ");
22 Serial.println(WiFi.subnetMask());
23 Serial.print("Gateway IP: ");
24 Serial.println(WiFi.gatewayIP());
25 Serial.print("DNS: ");
26 Serial.println(WiFi.dnsIP());
27     server.begin();
28 }
29 int value = 0;
30

```

Fuente: elaboración propia (2019).

Figura 31

Líneas de uso aplicación-usuario.

```

1  Líneas de uso Aplicación-Usuario
2
3  void loop(){
4      WiFiClient client = server.available();
5      if (client) {
6
7          // if you get a client,
8          Serial.println("New Client.");
9          String currentLine = "";
10         while (client.connected()) {
11             if (client.available()) {
12                 char c = client.read();
13                 Serial.write(c);
14                 if (c == '\n') {
15
16                     if (currentLine.length() == 0) {
17
18                         client.println("HTTP/1.1 200 OK");
19                         client.println("Content-type:text/html");
20                         client.println();
21
22                         client.print("Click <a href=\"/HB\">here</a> Encender Luz baño on.<br>");
23                         client.print("Click <a href=\"/LB\">here</a> Apagar Luz baño off.<br>");
24                         client.print("Click <a href=\"/HC\">here</a> Encender Luz Cocina on.<br>");
25                         client.print("Click <a href=\"/LC\">here</a> Apagar Luz Cocina off.<br>");
26                         client.print("Click <a href=\"/HH\">here</a> Encender Luz Habitación on.<br>");
27                         client.print("Click <a href=\"/LH\">here</a> Apagar Luz Habitación off.<br>");
28                         client.print("Click <a href=\"/AB\">here</a> Abrir puerta baño.<br>");
29                         client.print("Click <a href=\"/CB\">here</a> Cerrar puerta baño.<br>");
30                         client.print("Click <a href=\"/AH\">here</a> Abrir puerta habitación.<br>");
31                         client.print("Click <a href=\"/CH\">here</a> Cerrar puerta habitación.<br>");
32                         client.print("Click <a href=\"/AE\">here</a> Abrir puerta entrada.<br>");
33                         client.print("Click <a href=\"/CE\">here</a> Cerrar puerta entrada.<br>");
34                         client.print("Click <a href=\"/AV\">here</a> Abrir ventana.<br>");
35                         client.print("Click <a href=\"/CV\">here</a> Cerrar ventana.<br>");
36

```

Fuente: elaboración propia (2019).

En las figuras 32 y 33, se definieron las variables que van a guardar las diferentes opciones del aplicativo, para luego, ser interpretadas por el software y generar la orden al Microprocesador.

Figura 32

Lineas de acción del aplicativo.

```

1  Líneas de acción del Aplicativo
2
3  client.println();
4      break;
5  } else {
6      currentLine = "";
7  }
8  } else if (c != '\r') {
9      currentLine += c;
10 }
11
12     if (currentLine.endsWith("GET /HB")) {
13         digitalWrite(2, HIGH); // Encender luz baño
14     }
15     if (currentLine.endsWith("GET /LB")) {
16         digitalWrite(2, LOW); //Apagar Luz Baño
17     }
18     if (currentLine.endsWith("GET /HC")) {
19         digitalWrite(15, HIGH); // Encender luz cocina
20     }
21
22     if (currentLine.endsWith("GET /LC")) {
23         digitalWrite(15, LOW); // Apagar luz cocina
24     }
25
26     if (currentLine.endsWith("GET /HH")) {
27         digitalWrite(4, HIGH); // Encender luz habitacion
28     }
29
30     if (currentLine.endsWith("GET /LH")) {
31         digitalWrite(4, LOW); // Apagar luz habitacion
32     }
33
34
35     if (currentLine.endsWith("GET /AB")) {
36         for(int AB = 0; AB <= 90; AB++) {
37             servo2.write(AB); // Abrir puerta baño
38             delay(15);
39         }
40     }
41
42     if (currentLine.endsWith("GET /CB")) {
43         for(int CB = 90; CB >= 0; CB--) {
44             servo2.write(CB); // Cerrar puerta baño
45             delay(15);
46         }

```

Fuente: elaboración propia (2019).

Figura 33

Líneas de acción del aplicativo.

```

47     }
48     if (currentLine.endsWith("GET /AH")) {
49         for(int AH = 0; AH <= 90; AH++) {
50             servo1.write(AH); // Abrir puerta habitacion
51             delay(15);
52         }
53     }
54     if (currentLine.endsWith("GET /CH")) {
55         for(int CH = 90; CH >= 0; CH--) {
56             servo1.write(CH); // Cerrar puerta habitacion
57             delay(15);
58         }
59     }
60 }
61
62     if (currentLine.endsWith("GET /AE")) {
63         for(int AE = 0; AE <= 90; AE++) {
64             servo3.write(AE); // Abrir puerta entrada
65             delay(15);
66         }
67     }
68     if (currentLine.endsWith("GET /CE")) {
69         for(int CE = 90; CE >= 0; CE--) {
70             servo3.write(CE); // Cerrar puerta entrada
71             delay(15);
72         }
73     }
74     if (currentLine.endsWith("GET /AV")) {
75         for(int AV = 90; AV >= 0; AV--) {
76             servo4.write(AV); // Abrir ventana
77             delay(15);
78         }
79     }
80
81     if (currentLine.endsWith("GET /CV")) {
82         for(int CV = 0; CV <= 90; CV++) {
83             servo4.write(CV); // Cerrar ventana
84             delay(20);
85         }
86     }
87 }
88
89 // close the connection:
90 client.stop();
91 Serial.println("Client Disconnected.");
92 }
93 }
94
    
```

Fuente: elaboración propia (2019).

En la siguiente figura 34, se muestra un ciclo “Si”, que determina una acción producto de una orden ejecutada por el usuario del sistema domótico:

Figura 34

Ciclo para encender luz baño

```

config conexion modulo wifi esp32.txt  Líneas de conexión wi fi  Líneas de activación
1  if (currentLine.endsWith("GET /HB")) {
2      digitalWrite(2, HIGH);           // Encender luz baño
3  }
4
    
```

Fuente: Elaboración propia (2019).

Construcción de la maqueta e implementación del hardware y software

Para la construcción de la maqueta donde se instaló el sistema domótico, se utilizó una escala de 1:25, con el fin, de poder apreciar las funciones preestablecidas y tener un entorno dimensionado con los periféricos o actuadores. Con respecto al material para su construcción, se seleccionó madera tipo “balsa”, esto para dar rigidez a la construcción total y a los elementos electrónicos que estarán dispuestos dentro de ella.

Seguidamente para el despiece, se hizo uso de cortes con máquina de precisión, con el fin, de evitar desperdicios de material, tal como se observa en las figuras 35 y 36:

Finalmente, para la unión y pegado de las piezas se utilizó silicona líquida y colbón para madera, como se muestra en la figura 37:

Figura 35

Despiece del material

Fuente: elaboración propia (2019).

Figura 36
Cortes CNC Laser

Fuente: elaboración propia (2019)

Figura 37
Montaje prototipo

Fuente: elaboración Propia (2019)

Implementación del hardware en la maqueta

Se procedió a agregar el circuito a la locación construida, iniciando por el recorrido del cableado interno donde se ubicaron los motores y los diodos LED con el fin obtener la medida de las distancias a las cuales estarán separados los componentes para evitar desperdicios, posteriormente, se ubicaron los componentes de manera temporal para simular el funcionamiento del sistema domótico por medio del reconocimiento de voz con las funciones preestablecidas. Finalmente, se procedió a darle el acabado final a la maqueta y se agregaron los objetos porcelánicos, se cubrió las paredes interiores y pisos con papel “contact” que simula la tableta de porcelanato dando el efecto de una locación lista para habitar, como se demuestra en la figura 38:

Figura 38

Acabado de maqueta en funcionamiento

Fuente: elaboración propia (2019).

Evaluación

En esta fase se realizaron las pruebas de acuerdo con el método de caja negra (tablas 3 y 5). Este método, permitió evaluar la funcionalidad del sistema sin considerar el código a lo interior, los detalles referentes al despliegue, la ejecución interna en el software y las entradas/salidas (que se encuentran descritos en los requerimientos de software y sus especificaciones funcionales) del sistema domótico. En consecuencia, se utilizó la “Plantilla de Casos de Pruebas de Software” creada por www.pmoinformatica.com, donde se comprobó que la aplicación cumple con los objetivos de la investigación acerca de desarrollar un sistema de reconocimiento de voz a escala para personas con discapacidad motora en beneficio de su inclusión social y laboral (tabla 4 y 6).

Consecuentemente, el software desarrollado aprobó las pruebas ejecutadas a través del método caja negra, ofreciendo una aplicación con total interactividad por medio del usuario, y, donde los errores de reconocimiento de voz se vieron afectados por un ruido mayor a 60dB, sin embargo, se cuenta con un sistema de respaldo, en caso de ruido mayor a 60dB, pérdida del internet, y/o fallas con el reconocimiento de voz garantizando tolerancia a fallas. A continuación, se muestran las pruebas de caja negra y consecutivamente, los resultados arrojados:

Tabla 4
Pruebas de software (Black box testing)

Id	AII	EII	ACI
Caso de Prueba	Apagar iluminarias	Encender Iluminarias	Abrir accesos
Descripción	Para esta prueba se requiere de un medidor de ruido para lo cual se usó un smartphone con la aplicación medidor de sonido, otro smartphone con la aplicación Prevoice, una persona con la capacidad de leer y pronunciar las funciones establecidas	Para esta prueba se requiere de un medidor de ruido para lo cual se usó un smartphone con la aplicación medidor de sonido, otro smartphone con la aplicación Prevoice, una persona con la capacidad de leer y pronunciar las funciones establecidas	Para esta prueba se requiere de un medidor de ruido para lo cual se usó un smartphone con la aplicación medidor de sonido, otro smartphone con la aplicación Prevoice, una persona con la capacidad de leer y pronunciar las funciones establecidas
Fecha	15/10/2019	15/10/2019	15/10/2019
Área Funcional / Sub proceso	Apagar	Encender	Abrir
Funcionalidad / Característica	Reconocimiento de voz	Reconocimiento de voz	Reconocimiento de voz
Datos / Acciones de Entrada	Pronunciar: Luz cocina, Luz baño, Luz habitación	Pronunciar: Luz cocina, Luz baño, Luz habitación	Pronunciar: Puerta entrada, puerta baño, puerta habitación, ventana
Resultado Esperado	Ejecutar la función	Ejecutar la función	Ejecutar la función
Entornos del Plan de Pruebas de Software.	Ruido <60dB	Ruido <60dB	Ruido <60dB
Procedimientos especiales requeridos	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC
Dependencias con otros casos de Prueba	No aplica	No aplica	No aplica

Fuente: elaboración propia (2019)

Tabla 5
Pruebas de software (Black box testing)

Resultado Obtenido	Se ejecuta la acción	Se ejecuta la acción	Se ejecuta la acción
Estado	Finalizado	Finalizado	Finalizado
Última Fecha de Estado	5/11/2019	5/11/2019	5/11/2019
Observaciones	Con un ruido mayor a 60dB el reconocimiento de voz de Android no reconoce la voz de la persona con discapacidad motriz por lo que no hay conexión con los servidores de Google assistant y no se ejecuta la acción establecida	Con un ruido mayor a 60dB el reconocimiento de voz de Android no reconoce la voz de la persona con discapacidad motriz por lo que no hay conexión con los servidores de Google assistant y no se ejecuta la acción establecida	Con un ruido mayor a 60dB el reconocimiento de voz de Android no reconoce la voz de la persona con discapacidad motriz por lo que no hay conexión con los servidores de Google assistant y no se ejecuta la acción establecida

Fuente: elaboración propia (2019)

Tabla 6
Pruebas de software (Black box testing)

Id	CA1	ACA1	EAI1
Caso de Prueba	Cerrar accesos	Abrir/cerrar accesos	Encender/apagar luminarias
Descripción	Para esta prueba se requiere de un medidor de ruido para lo cual se usó un smartphone con la aplicación medidor de sonido, otro smartphone con la aplicación Prevoice, una persona con la capacidad de leer y pronunciar las funciones establecidas	Para esta prueba se requiere de un smartphone Android que tenga la aplicación prevoice, y una persona con motricidad fina	Para esta prueba se requiere de un smartphone android que tenga la aplicación prevoice, y una persona con motricidad fina
Fecha	15/10/2019	15/10/2019	15/10/2019
Área Funcional / Sub proceso	Cerrar	Abrir/Cerrar	Encender/apagar
Funcionalidad / Característica	Reconocimiento de voz	Sistema de respaldo	Sistema de respaldo
Datos / Acciones de Entrada	Pronunciar: Puerta entrada, puerta baño, puerta habitación, ventana	Oprimir: Abrir y cerrar (Puerta baño, Puerta baño, Puerta habitación, ventana)	Oprimir: Encender y apagar (Luz cocina, luz baño, luz habitación)
Resultado Esperado	Ejecutar la función	Ejecutar la función	Ejecutar la función
Entornos del Plan de Pruebas de Software.	Ruido <60dB	Sin conexión a internet	Sin conexión a internet
Procedimientos especiales requeridos	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC	Estar dentro de la misma red local del controlador y servidor http ESP32 SoC
Dependencias con otros casos de Prueba	No aplica	No aplica	No aplica

Fuente: elaboración propia (2019).

Tabla 7
Pruebas de software (Black box testing)

Resultado Obtenido	Se ejecuta la acción	Se ejecuta la acción	Se ejecuta la acción
Estado	Finalizado	Finalizado	Finalizado
Última Fecha de Estado	5/11/2019	5/11/2019	5/11/2019
Observaciones	Con un ruido mayor a 60dB el reconocimiento de voz de Android no reconoce la voz de la persona con discapacidad motriz por lo que no hay conexión con los servidores de Google assistant y no se ejecuta la acción establecida	Sin conexión a Internet el sistema de respaldo funciona correctamente permitiendo tener control de los diferentes objetos en tiempo real y sirviendo como un módulo que garantice la tolerancia a falla	Sin conexión a Internet el sistema de respaldo funciona correctamente permitiendo tener control de los diferentes objetos en tiempo real y sirviendo como un módulo que garantice la tolerancia a falla

Fuente: elaboración propia (2019).

Conclusiones

Una vez efectuado el estudio de los resultados, se pudo constatar la contribución de la domótica como un conglomerado de tecnologías que coadyuban a la automatización inteligente de una locación para optimizar sus espacios, en este caso, con pertinencia e inclusión de personas con deficiencias motoras, es así, como se plantea la mejora en su calidad de vida dando paso al desarrollo de actividades que antes serían difíciles de lograr.

De la misma forma existen numerosas evidencias discutidas en muchos espacios de la ciencia que vinculan la herramienta domótica para el tema mundial como lo es la inclusión.

Por otra parte, las pruebas funcionales como resultado del desarrollo de este sistema domótico a escala, permitió confirmar que esta tecnología podría conseguir mejores sustanciales en el quehacer diario para personas con incapacidad funcional, pues, a través de comandos de voz se pueden ejecutar funciones que brindan mayor comodidad y son perfectamente funcionales y discretas.

Finalmente, con el desarrollo de este sistema demótico, se confirmó cómo a través del enfoque de inteligencia artificial se puede tomar control de objetos en un modelo a escala a través de comandos de voz, para apoyar a las personas con discapacidad motriz y cómo, a futuro, esta propuesta a escala real brindará mayor calidad de vida adecuada y sostenible a los mismos, acercándolos a una vida más digna, considerando además que esta alternativa podría ser viable para personas con otras discapacidades y consentir la dotación de espacios óptimos como mecanismo de inclusión.

A manera de conclusión general con este estudio se desarrolló un software basado en la tecnología de reconocimiento de voz y su impacto positivo en la calidad de vida de personas en condiciones de discapacidad motriz.

Referencias bibliográficas

- Arias, L. (2006). *Investigación Científica*. . Editorial Mc Graw Hill. Segunda Edición.
- Armas, E. Á. (2019). Aids Based on Speech Recognition Technology for People with Motor Disabilities and Reduced Mobility. *Revista Politécnica*, 43(1).
- Cegarra, J. (2004). *Metodología de la investigación científica y tecnológica*. Madrid, España: Ediciones Díaz de Santos .
- Chavez, N. (2007). *Introducción a la investigación educativa /por Nilda Chavez Alizo*. Maracaibo: Gráfica González.
- De Mora, C., Castro, M., Yeves, F., Peire, J., Mur, F., Pérez, C., . . . Fernández, R. (2013). *Estructura y Tecnología de Computadores I (Gestión y Sistemas)*. Madrid, España: Universidad Nacional de Educación a Distancia.
- Esparza , M. E. y Avalos Briseño, J. B. (23 de Julio de 2019). *Reconocimiento de voz*. Ciencia Tecnológica, págs. 1-3.
- GitHub. (30 de 08 de 2019). *GitHub*. <https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/windows.md>
- González y Rodríguez. (2011). *Metodología de la investigación*. México: McGraw-Hill.
- Hernández, R., Fernández, C., y Baptista, P., (2010). *Metodología de la Investigación*. 5ª edición. México: Mc. Graw-Hill.
- Joumaa, H., Ploix, S., Abras, S., y De Oliveira, G. (2011). *A MAS integrated into Home Automation system, for the resolution of power management problem in smart homes*. Energy Procedia, 786-794. doi:<https://doi.org/10.1016/j.egypro.2011.05.089>
- Korkmaz, I., Metin, S., Gurek, A., Gur, C., Gurakin, C., y Akdeniz, M. (2015). *A cloud based and Android supported scalable home automation system*. Computers & Electrical Engineering, 43, 112-128.
- Louis, J.-N., Calo, A., Leiviskä, K., y Pongrácz, E. (2015). *Environmental Impacts and Benefits of Smart Home Automation: Life Cycle Assessment of Home Energy Management System*. IFAC-PapersOnLine, 48, 880-885. doi:<https://doi.org/10.1016/j.ifacol.2015.05.158>
- Mendez, N. (2006). *La investigación Documental*. . Editorial Romor. Segunda Edición.
- Molero Suárez, L., Batista, J., y Puentes, M. (2019). *Transformación digital desde la cultura tecnológica en las instituciones de educacion superior del estado zulia*. Consensus, Santiago de chile, Chile.
- Molero, L., Sáenz, J., Montenegro, J., Castiblanco, L., y Ruiz, J. (2019b). *Interfaz 3D móvil con HMD para personas con discapacidad motora apoyada en metodología SUM*. En L. Molero Suárez, J. Sáenz Leguizamón , J. Montenegro Romero, L. Castiblanco Rosero, & J. Ruiz Carrillo, Gamificación y discapacidad. Una

alternativa socialmente responsable, Volumen I. Santa Ana de Coro, Venezuela: Fondo Editorial Universitario Servando Garcés de la Universidad Politécnica Territorial de Falcón Alonso Gamero.

- Monsó, J. (1994). *Sistemas de identificación y control automatizados (II). Sistemas de control de flujo físico*. Barcelona, España: marcombo BOIXAREU Editores.
- Parsch, P., y Masrur, A. (2018). *On reliable communication in transmit-only networks for home automation*. Journal of Network and Computer Applications, 101(1), 96-110. doi:<https://doi.org/10.1016/j.jnca.2017.10.023>
- Perez. (2010). *Enfoques metodológicos*. Editorial La Salle.
- Puertas, J. (2000). *Robustez en el reconocimiento fonetico de voz para aplicaciones*. Madrid. doi:<http://oa.upm.es/657/>
- RLCPD. (29 de 01 de 2020). *RLCPD*. https://www.minsalud.gov.co/proteccionsocial/Paginas/DisCapacidad_RLCPD.aspx
- Saavedra, E., Durán, C., Escalera, M., Mora, B., Pacheco, A., y Pérez, M. (2018). *Discapacidad motora y Resiliencia en adultos*. En: A. D. Marengo-Escuderos. Estudios del desarrollo humano y socioambiental. Barranquilla, Colombia: Ediciones C.
- Talbot, C., Temple, M., Carbino, T., y Betances, J. (2018). *Detecting rogue attacks on commercial wireless Insteon home automation systems*. Computers & Security, 74, 296-307. doi:<https://doi.org/10.1016/j.cose.2017.10.001>
- Tamayo y Tamayo, M. (2006). *El proceso de la investigación científica*. México: Limusa.