Universidad de Lima

Facultad de Ingeniería y Arquitectura

Carrera de Ingeniería de Sistemas

SISTEMA DE CONTROL POR VOZ PARA UN ENTORNO DOMÓTICO ADAPTADO A PERSONAS CON DISCAPACIDAD FÍSICA UTILIZANDO MODELOS OCULTOS DE MARKOV

Tesis para optar el Título Profesional de Ingeniero de Sistemas

Miguel Ricardo Lladó Herrera

Código 20120723

Asesor

Lennin Paul Quiroz Villalobos

Lima – Perú

Mayo de 2020

UNIVERSIDAD DE LIMA
MCMLXV
SCIENTIA ET PRAXIS

VOICE CONTROL SYSTEM FOR A DOMOTIC ENVIRONMENT ADAPTED TO PEOPLE WITH PHYSICAL DISABILITIES USING HIDDEN MARKOV MODELS

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El procesamiento de voz y la domótica son tecnologías que se han desarrollado ampliamente en los últimos años. Sin embargo, las aplicaciones que hacen uso de estas dos tecnologías de manera conjunta para ayudar a personas con discapacidad física son prácticamente inexistentes, las pocas disponibles tienen un costo de licenciamiento exageradamente alto y con el agravante que la mayoría de estas soluciones no incluyen el español latinoamericano entre sus capacidades de reconocimiento de voz. Esto implica, que las personas con algún tipo de limitación física en sus movimientos no puedan tener acceso a una solución tecnológica factible que les permita interactuar con su entorno de forma natural y mejorar así su calidad de vida.

Es por ello que el presente trabajo tuvo como objetivo crear un sistema de control por voz para un entorno domótico adaptado a personas con discapacidad en español latinoamericano. Para lo cual se utilizó la técnica de los modelos ocultos de Markov, por ser una de las que mejores resultados ha demostrado en esta tarea, implementada mediante el toolkit CMUSphinx.

Además, con el fin de reducir la posibilidad de error en el reconocimiento, se hizo un enfoque especial hacia la personalización, adaptando el reconocedor a las características propias de la voz del usuario final. Esto fue validado mediante pruebas en las que se mide el porcentaje de error que brinda el reconocedor en distintas condiciones.

Finalmente, se realizó una comparativa en la que se analizan distintos parámetros de rendimiento entre la solución propuesta y otras soluciones existentes en el mercado. En ella se muestran las características únicas del reconocedor implementado y el por qué es una solución válida a los problemas identificados; concluyendo que es posible crear un sistema de control por voz para un entorno domótico adaptado a personas con discapacidad mediante el uso de modelos ocultos de Markov, sobre un estándar libre y de bajo costo, que permita a las personas con discapacidad física interactuar con su entorno.

Palabras clave: modelos ocultos de Markov, domótica, reconocimiento de voz, modelo acústico, modelo de lenguaje, diccionario fonético.

ABSTRACT

Speech recognition and home automation are technologies that have been widely developed in recent years. However, the applications that make use of these two technologies together to help people with disabilities are virtually non-existent, the few available have a high license cost and with the aggravating fact that most of these solutions do not include Latin American Spanish among their options. This implies that people with some kind of physical limitation in their movements cannot have access to a feasible technological solution that allows them to interact with their environment in a natural way and thus improve their quality of life.

That is why, this work aimed to create a home automation control system based on a voice command recognizer in Latin American Spanish. For which the Hidden Markov Models technique was used, as it is one of the best results demonstrated in this task, implemented using the CMUSphinx toolkit.

In addition, in order to minimize the possibility of error in recognition, a special approach was made towards personalization, adapting the recognizer to the characteristics of the voice of the end user. This has been validated by tests in which the percentage of error that the recognizer provides in different situations is measured.

Finally, a comparison was made in which different performance parameters are analyzed between the proposed solution and other existing solutions in the market. It shows the unique characteristics of the recognizer implemented and why it is a valid solution to the problems identified; concluding that it is possible to create a home automation control system based on voice recognition commands through Hidden Markov Models on a free and low-cost standard that allows people with physical disabilities to interact with their environment.

Keywords: Hidden Markov Models, domotics, speech recognition, acoustic model, language model y pronunciation dictionary

INTRODUCCIÓN

Actualmente, se está viviendo un aumento progresivo de la tasa de personas con discapacidad física de algún tipo (monoplejía, paraplejía, hemiplejía, distrofia muscular, amputación, etc.), esto debido, principalmente: (1) al envejecimiento general de la población (Europa, Asia), (2) el aumento de las enfermedades crónicas (Organización Mundial de la Salud, 2018), (3) al haber un aumento constante de la población mundial, los números absolutos de personas con discapacidad también aumentan (Bongaarts, 2009).

Por otro lado, la domótica no ha sido ampliamente adoptada, a pesar de que ha estado disponible por más de tres décadas. Esto se debe a cuatro factores: el alto costo, inflexibilidad, pobre manejabilidad y dificultad de garantizar seguridad (Brush et al., 2011, p. 2115). Su adopción facilita enormemente la vida diaria de las personas con discapacidad, otorgándoles independencia y comunicación (López, Ponce, Piccinini, Perez y Roberti, 2016, p. 41).

Es por ello, que el presente trabajo tiene como finalidad presentar la creación de un sistema de control por voz de bajo costo para un entorno domótico adaptado a personas con discapacidad física. La implementación del sistema de reconocimiento de voz se realizó sobre un sistema domótico abierto, debido a que este tipo de sistemas presenta una mayor flexibilidad, es decir, pueden comprarse distintas marcas de dispositivos e integrarse entre sí (Brush et al., 2011, p. 2119). Además, esta interoperabilidad permite una disminución de costos, debido a que rompe con la limitación que traen consigo los sistemas propietarios de usar una o un conjunto de marcas específicas (Pham-Huu, Nguyen, Trinh, Bui y Pham, 2015). Así mismo, con el objetivo de reducir los costos al mínimo posible, el reconocedor implementado no requiere internet para procesar el reconocimiento lo que posibilita su uso en viviendas que no cuenten con este servicio (Prasanna y Ramadass, 2014).

Por otro lado, la pobre manejabilidad tradicional de los sistemas domóticos es solucionada mediante el uso de comandos de voz para simplificar el manejo de las tareas

sencillas. Esto es especialmente útil en el caso de personas con limitación de movimientos (Obaid et al., 2014, p. 49), público objetivo, al que se pretende otorgar una mejora en su calidad de vida mediante el presente trabajo.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Formulación del problema

1.1.1 Antecedentes

1.1.1.1 Aumento de la población discapacitada

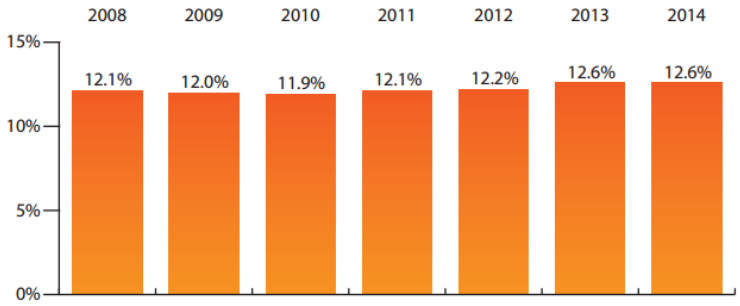
Para poder determinar si existe un aumento del número de personas con discapacidad, primero es necesario definir qué es discapacidad. Esta, según la Organización Mundial de la Salud (2018) puede ser definida como un término genérico que abarca cualquiera de dos tipos de deficiencias: limitaciones de la actividad o restricciones a la participación. Las limitaciones de la actividad son dificultades para ejecutar acciones o tareas, y las restricciones de la participación son problemas para participar en situaciones vitales.

El término engloba las deficiencias físicas o mentales, como la discapacidad motriz, cognitiva o intelectual, la enfermedad mental o varios tipos de enfermedades crónicas. Estas discapacidades impiden a la persona relacionarse correctamente con el entorno; por ejemplo, el tener una discapacidad física que afecte a la movilidad dificulta el desenvolvimiento cotidiano en una casa, mientras que una discapacidad intelectual dificulta el desempeño académico (Organización Mundial de la Salud, 2018).

Actualmente, más de 1000 millones de personas padecen algún tipo de discapacidad, es decir, 15 de cada 100 personas. En el caso de discapacidades severas, se calcula que entre el 2,2% y el 3,8% de la población está dentro de esta categoría (Organización Mundial de la Salud, 2018). En Estados Unidos se puede observar la siguiente tendencia al alza:

Figura 1.1

Porcentaje de personas con discapacidad en los Estados Unidos, 2008-2014



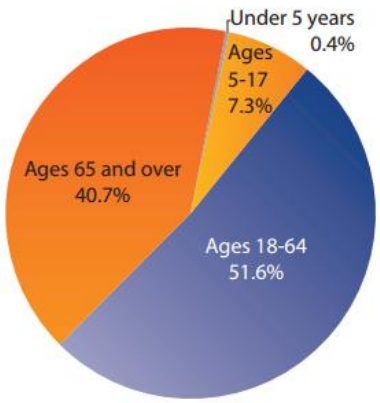
Nota. De *Disability Statistics Annual Report*, por Kraus L., 2015, https://disabilitycompendium.org/sites/default/files/user-uploads/2015%20Annual%20Report%20%28PDF%29.pdf

Estas tasas están en crecimiento debido al envejecimiento de la población y el aumento de las enfermedades crónicas, entre otras causas (Organización Mundial de la Salud, 2018).

Esto puede dar a entender que la mayoría de las personas discapacitadas es la de mayor edad, pero eso no es del todo cierto. Por ejemplo, en Estados Unidos, en el 2014, se presentó la siguiente distribución:

Figura 1.2

Personas con discapacidad en los Estados Unidos en 2014 según la edad



Nota. De *Disability Statistics Annual Report*, por Kraus L., 2015, https://disabilitycompendium.org/sites/default/files/user-uploads/2015%20Annual%20Report%20%28PDF%29.pdf

En el Perú, la información sobre las personas con discapacidad ha sido muy variable a lo largo del tiempo y no se ha llegado a un estándar. Esto se debe a las distintas metodologías usadas para obtener la información. Por ejemplo, según el censo del año 2007, el porcentaje de personas discapacitadas ascendía a 10,9%; mientras que la Primera Encuesta Nacional Especializada sobre Discapacidad, realizada en el año 2012, otorga un porcentaje del 5,2% (Instituto Nacional de Estadística e Informática, 2014).

Según el último estudio realizado, la Primera Encuesta Nacional Especializada sobre Discapacidad, la situación en el año 2012 se presenta de la siguiente manera:

Figura 1.3

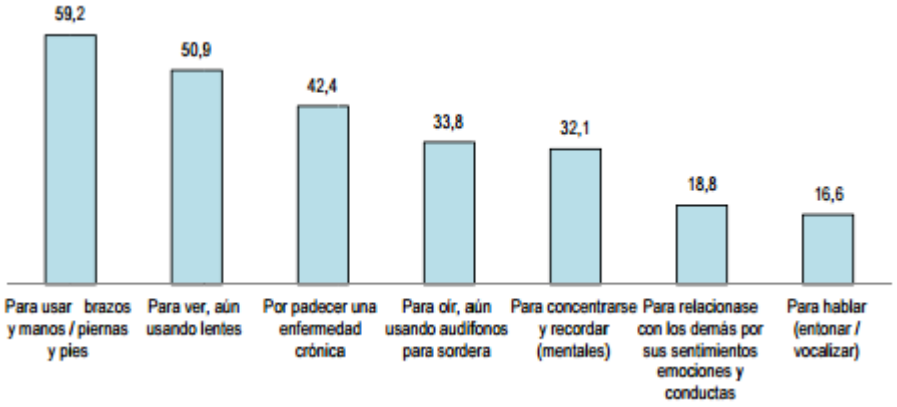
Perú: personas con alguna discapacidad por sexo y grupos de edad, 2012

Nota. De *Primera Encuesta Nacional Especializada Sobre Discapacidad 2012*, por Instituto Nacional de Estadística e Informática, 2014, https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1171/ENEDIS%202012%20-%20COMPLETO.pdf

Dividiéndolo por tipo de limitación:

Figura 1.4

Perú: personas con discapacidad según tipo de limitación para realizar sus actividades diarias en porcentaje, 2012



Nota. De *Primera Encuesta Nacional Especializada Sobre Discapacidad 2012*, por Instituto Nacional de Estadística e Informática, 2014, https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1171/ENEDIS%202012%20-%20COMPLETO.pdf

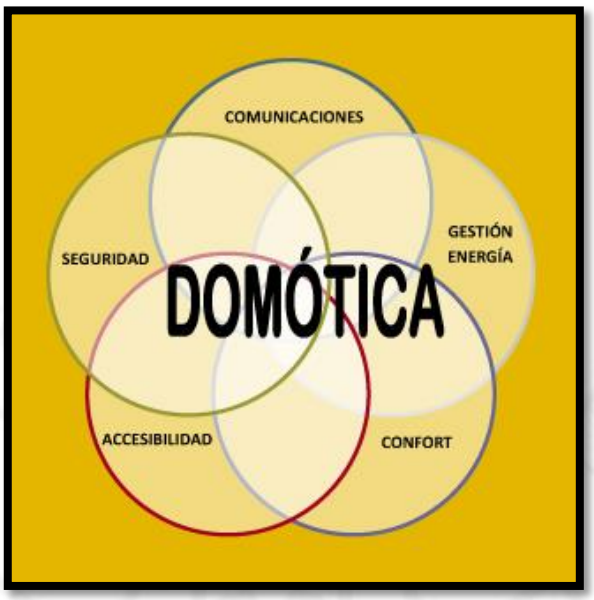
1.1.1.2 Domótica

La domótica, según la Asociación Española de Domótica e Inmótica (CEDOM), "es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema." (Asociación Española de Domótica e Inmótica, s. f, sección Qué es Domótica, párr. 1).

Según la CEDOM, la domótica brinda una serie de aportes al usuario final, entre las que se encuentran: ahorro energético, accesibilidad, seguridad, confort o comunicaciones (Asociación Española de Domótica e Inmótica, s. f.).

Figura 1.5

Aportes de la domótica



Nota. De *Qué es Domótica*, por Asociación Española de Domótica e Inmótica, s. f, http://www.cedom.es/sobre-domotica/que-es-domotica

El concepto de casa inteligente ha atraído la atención de varios investigadores; sin embargo, actualmente, la domótica aún no ha sido ampliamente adoptada, fundamentalmente debido al alto costo de los equipos debido a tecnologías propietarias, escasa manejabilidad y riesgos de seguridad (Brush et al., 2011, p. 2115).

Esta situación está cambiando poco a poco y se tienen previsiones optimistas de cara al futuro. Para el 2022, las previsiones indican que el mercado de la domótica tendrá un valor de 78.27 mil millones de dólares (Markets and Markets, 2016).

1.1.1.3 Reconocimiento del habla

Un reconocedor de voz es un sistema que captura un mensaje de voz recibido y lo convierte en texto.

El reconocimiento del habla se remonta a la década de 1950, momento en el cual se crearon los primeros reconocedores de voz, aunque estos tenían grandes limitaciones. Durante la década de 1980, se presentaron grandes avances en la representación

estadística de la voz. Estos avances ayudaron a mejorar el reconocimiento de voz y lograron que hoy en día los reconocedores de voz tengan una amplia variedad de aplicaciones y un uso cada vez mayor (Juang y Rabiner, 2005).

1.2 Objetivo de la investigación

1.2.1 Objetivo general

- Diseñar e implementar un sistema de control por voz fuera de línea de bajo costo para un entorno domótico adaptado a personas con discapacidad física usando modelos ocultos de Markov.

1.2.2 Objetivos específicos

- Implementar un sistema de control por voz basado en modelos ocultos de Markov.
- Implementar un sistema de control por voz de bajo costo basado en una plataforma open hardware.
- Implementar un sistema de control por voz domótico fuera de línea.

1.3 Justificación

El presente trabajo es importante ya que ofrece una alternativa viable a las principales dificultades de las soluciones domóticas actuales para personas con discapacidad: (1) el alto costo, (2) inflexibilidad, (3) pobre manejabilidad y (4) seguridad.

El alto costo y la inflexibilidad son conceptos relacionados debido al uso de software, componentes e interfaces propietarias; pues al limitarse la compatibilidad únicamente a ciertas marcas, los costos se elevan. El uso de hardware y software libre combate uno de los principales problemas de la baja adopción de la domótica: el alto costo del software y los equipos.

La pobre manejabilidad para personas con discapacidad física está relacionada al limitado uso combinado de la domótica y el reconocimiento de voz. El poder utilizar comandos de voz para la interacción es de gran ayuda para este grupo de personas.

Además, se cuentan con muy pocas opciones de control por voz en español latinoamericano; esto dificulta aún más el uso de esta tecnología para personas que hablen esta lengua.

La dificultad de conseguir seguridad siempre ha sido un problema intrínseco a la domótica. El riesgo de que algún agente externo pueda manipular objetos del hogar siempre está latente, este riesgo es aún mayor en sistemas que para funcionar requieren estar conectados permanentemente a internet. La no necesidad de conexión a internet de la presenta propuesta para realizar el procesamiento, elimina estos riesgos de seguridad.

Estas cuatro dificultades son el motivo por el cual el presente trabajo desarrolló una solución en la que estas dificultades son eliminadas o mitigadas con el objetivo de acercar estas tecnologías a las personas con discapacidad.

1.4 Aportes

El presente trabajo implementa una herramienta open source de bajo costo que estará disponible para que personas con discapacidad física puedan implementar de forma sencilla un sistema de control por voz para un entorno domótico en su hogar.

Este sistema mejora la calidad de vida de las personas con discapacidad mediante la integración de la domótica y el procesamiento de voz; para esto, se propone el uso de plataformas open hardware aplicadas al control automático y el funcionamiento de los modelos ocultos de Markov aplicados al reconocimiento de comandos de voz.

La propuesta busca ofrecer una solución a cada una de las cuatro principales dificultades actuales en ambientes domóticos para personas con discapacidad.

Para solucionar el alto costo y la falta de flexibilidad, el sistema utiliza hardware y software libre. Con esto se evita pagar por el uso del procesador de voz o licencias por software o hardware propietario, logrando un costo reducido comparado al de un sistema cerrado (Pham-Huu et al., 2015). Además, se aumenta la compatibilidad con periféricos, lo que brinda más opciones para el usuario y a su vez también reduce costos.

La solución desarrollada también aborda el problema de la pobre manejabilidad enfocándose en lograr una interfaz intuitiva y accesible por voz para que las personas que tengan dificultades en las actividades cotidianas del hogar puedan desenvolverse sin

problemas. Para lograr eficiencia en el reconocimiento de los comandos de voz del usuario, el lenguaje utilizado es el español latinoamericano, además que ayuda a lograr un ambiente amigable, pues es el idioma nativo del público objetivo del presente trabajo.

A diferencia de otros reconocedores de propósito general enfocados a todo tipo de usuarios; la solución propuesta se ha diseñado específicamente para personas con discapacidad mediante la adaptación de este al usuario que utilice el sistema.

Finalmente, el sistema desarrollado funciona sin necesidad de conexión a internet, eliminando así los posibles problemas de seguridad que se puedan presentar al estar conectado. Adicionalmente, al no necesitar conexión, también se brinda la posibilidad de usar el reconocedor en una vivienda que no cuente con internet, ampliando así las posibilidades de uso.

CAPÍTULO II: ESTADO DEL ARTE

En primer lugar, se debe distinguir un sistema de reconocimiento del habla de un sistema de comprensión del habla. La comprensión del habla es un concepto más amplio, que engloba ciertos aspectos del reconocimiento del habla. Tiene como objetivo capturar el mensaje y entenderlo correctamente; a diferencia del reconocimiento del habla, cuyo objetivo pasa por capturar el mensaje y transcribirlo a texto (Milone, 2004).

El reconocimiento del habla se remonta a la década de 1950, específicamente al año 1952. En ese año, Biddulph, Balashek y Davis, trabajadores de la empresa Bell, crearon un sistema de reconocimiento de dígitos para un solo hablante (Juang y Rabiner, 2005).

En la década de los 60 se presentaron numerosos avances. Uno de estos avances fue el reconocedor de fonemas creado por Sakai y Doshita en la Universidad de Kyoto. En su trabajo, Sakai y Doshita, incorporaron el primer segmentador del habla. Gracias a esto, el reconocedor era capaz de separar los fonemas de la sentencia de voz dicha por el hablante para luego proceder con la identificación de los fonemas por separado. Previamente, los reconocedores asumían que el mensaje emitido por el hablante representaba únicamente a una unidad (dígito, fonema, etc.). Otro hito fue la creación de la técnica Dynamic Time Warping (DTW) por Vintsyuk en la Unión Soviética. Esta técnica permite la alineación temporal de 2 sentencias con el objetivo de medir su similitud (Antón, 2015; Juang y Rabiner, 2005).

En los años 70 se continuó avanzando en el reconocimiento del habla, llegando a reconocerse vocabularios más amplios. A finales de los años 60, se desarrolló el método Linear Predictive Coding (LPC) que simplificaba enormemente la estimación del tracto vocal desde una forma de onda proveniente del habla. Esto fue aprovechado por Itakura, Rabiner, Levinson, entre otras personas para aplicar al reconocimiento de voz técnicas de reconocimiento de patrones basadas en LPC. Durante esta década, también se alcanzaron otros hitos como el sistema Harpy desarrollado por la Carnegie Mellon University. Este reconocedor era capaz de reconocer hasta 1011 palabras con una precisión aceptable (Antón, 2015; Juang y Rabiner, 2005).

En la década de 1980, ya con bases más asentadas, se daría el siguiente gran cambio: el paso de métodos basados en comparación de plantillas a otros basados en modelos estadísticos, extendiéndose el uso de los modelos ocultos de Markov o HMMs. Este cambio surgió debido a la necesidad de crear un sistema de reconocimiento de voz que funcione independientemente del hablante. Por ejemplo, AT&T Bell Laboratories quería un sistema así para poder proveer servicios automatizados de telecomunicaciones a su público de millones de personas por lo que era viable realizar un entrenamiento para cada usuario (Antón, 2015; Juang y Rabiner, 2005).

En los 90, se ampliaron los campos de aplicación con especial énfasis en el uso en las líneas telefónicas. Además, para este punto, el tamaño del vocabulario se volvió virtualmente ilimitado, es decir, el hablante ya no debía preocuparse por saber que palabras estaban o no en el diccionario del sistema (Antón, 2015; Huang, Baker y Reddy, 2015).

En la actualidad, el reconocimiento de voz sigue mejorando cada vez más y tiene un uso cada vez más amplio en aplicaciones, siendo una parte fundamental de la interfaz de usuario como se puede observar en el iPhone con Siri (Wildstrom, 2011).

Figura 2.1

Siri funcionando en un iPhone

Nota. De *Usar Siri en todos tus dispositivos de Apple*, por Apple, 2019, https://support.apple.com/es-es/HT204389

2.1 Técnicas

Durante todo el desarrollo del reconocimiento de voz, se han abordado distintas aproximaciones sobre cómo se debe representar y reconocer la señal de voz de entrada. Hernando (1993), destaca especialmente cuatro técnicas:

2.1.1 Técnicas basadas en comparación de patrones

Las técnicas de comparación de patrones se dividen en dos fases: entrenamiento y reconocimiento.

En la fase de entrenamiento, el usuario pronuncia las palabras que formen parte de un vocabulario que se desee reconocer. La señal generada por las palabras se discretiza, se extrae la secuencia de vectores correspondientes y se etiqueta respecto a cada fonema/palabra con el objetivo de tomarlo como patrón de referencia (Hernando,1993; Juang y Rabiner, 2005).

En la fase de reconocimiento, la señal de voz es discretizada, se convierte en vectores y se compara con los vectores de referencia. El vector de referencia más cercano será elegido y la palabra o fonema que represente será tomado como respuesta correcta. Esta fase se apoya en algoritmos como el Dynamic Time Warping (Hernando,1993).

Esta técnica fue la primera en ser utilizada y se ha tenido que recurrir a otras aproximaciones para solucionar los problemas que conlleva (Hernando,1993).

2.1.2 Técnicas basadas en modelos ocultos de Markov

En los modelos ocultos de Markov cada una de las referencias es un modelo estocástico. Considerando una señal A, los modelos ocultos de Markov calculan la probabilidad de que la palabra o frase W haya sido emitida por la señal A. La finalidad es obtener la palabra o secuencia de palabras más probable y devolverla como respuesta (Rabiner, 1989).

2.1.3 Técnicas basadas en Redes neuronales

En esta aproximación, los valores de referencia son representados como patrones de actividad. Estos patrones se encuentran distribuidos sobre una red de unidades conectadas entre sí. Estas unidades transmiten señales entre sí, mientras la información transmitida es sometida a diversas operaciones. El funcionamiento de las unidades y sus conexiones es similar al funcionamiento de las neuronas, es por ello que este modelo se conoce como redes neuronales artificiales (Hernando,1993).

2.1.4 Técnicas basadas en conocimiento

Esta aproximación consiste en separar el conocimiento que va a usarse en el proceso de razonamiento para poder incorporarla y modificarla conforme se avance en el desarrollo del sistema. Esto implica que el sistema no puede entrenarse (autoaprender), sino que todo el procesado se maneja de forma manual. Esta aproximación no se utiliza debido a que implica una enorme cantidad de esfuerzo, llegándose a estimar que crear un sistema de reconocimiento independiente del locutor podría tardar unos 15 años (Hernando,1993).

2.2 Reconocimiento de voz para entornos domóticos adaptados a personas con discapacidad física

La aplicación del reconocimiento de voz para ayudar a las personas con discapacidad no es un concepto nuevo. Uno de los primeros trabajos en señalar esto es el de Leggett y Williams (1984), quienes hace 36 años mencionaron que una de las áreas más prometedoras en las que aplicar el reconocimiento de voz es en la ayuda de las personas con discapacidad.

A finales de la década de 1980, se investigó acerca de la viabilidad del reconocimiento de voz para la ayuda personas con discapacidad. Noyes, Haigh y Starr (1989) analizaron el uso del reconocimiento de voz en el control de brazos robóticos y en el uso en domótica enfocada en la discapacidad concluyendo que, si bien las aplicaciones tenían un correcto funcionamiento para la tecnología de la época, aún contaban con problemas para ser implementados en un entorno real, principalmente la baja precisión

de los reconocedores de voz de la época. El artículo también menciona la dificultad de conseguir un correcto desempeño en las personas que cuenten con discapacidad en el habla.

En la actualidad, se han presentado distintos acercamientos con diferentes resultados. Yuksekkaya, Kayalar, Tosun, Ozcan y Alkar (2006) proponen un sistema domótico inalámbrico para personas con discapacidad que puede ser controlado de tres formas: a través de redes GSM, a través de internet o por reconocimiento de voz. La solución de reconocimiento de voz desarrollada utiliza comparación de patrones con Dynamic Time Warping que presenta ciertas limitaciones respecto a otras técnicas de reconocimiento de voz (Hernando, 1993). El sistema desarrollado por Yuksekkaya et al. (2006) es una excelente opción para las personas con discapacidad funcional debido a la posibilidad de utilizar el control por voz.

Mosbah (2006) fue un paso más allá y desarrolló un sistema de reconocimiento de comparación de patrones con Dynamic Time Warping enfocado especialmente a personas con discapacidad. Para lograr esto, se utilizó como base un sistema de reconocimiento ya existente y se adaptó a cada usuario y a las palabras que este utilizaría. Esta adaptación demostró mejorar la precisión del reconocedor.

Otros artículos utilizan técnicas más modernas con un mejor desempeño y menos limitaciones. Nilsson y Ejnarsson (2002) analizan el desempeño de los modelos ocultos de Markov en el reconocedor hidden Markov toolkit (HTK) en entornos con ruido. El desempeño logrado es variable, yendo desde un 75% de precisión en un entorno con bastante ruido hasta un 100% de precisión en entorno libre de ruido. El artículo señala que el reconocedor puede ser utilizado en el tráfico mientras se conduce o para facilitar la vida de personas con discapacidades funcionales. Sin embargo, el reconocedor de voz no se ha diseñado específicamente con ese objetivo.

Los modelos ocultos de Markov también pueden ser utilizados en un entorno domótico de bajo costo debido a la posibilidad de usarlos sobre un dispositivo de asequible como un Raspberry Pi. Además, este tipo de reconocedor no necesita una conexión a internet para su funcionamiento (Prasanna y Ramadass, 2014).

Las redes neuronales artificiales también han sido utilizadas para el reconocimiento de voz en entornos domóticos con distintos enfoques y resultados. Una

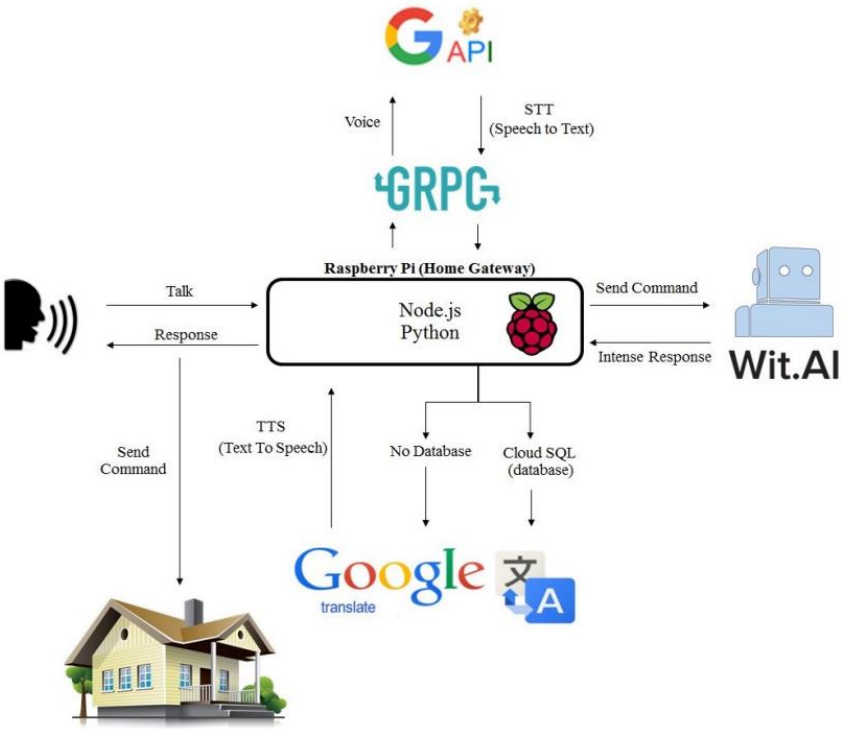
solución utilizada para implementarlas es el chip HM2007 que cuenta con un reconocedor de voz sencillo integrado capaz de reconocer entre 20 y 40 palabras (Carbureanu, Mihalache y Gheorghe, 2016).

Jabbar, Abbas y Safi (2018) implementan este chip en un entorno domótico con éxito resaltando su uso para las personas con discapacidad y ancianos. Aqeel-ur-Rehman y Khursheed (2014) utilizan el HM2007 para implementar una solución personalizada para anciano y personas con discapacidad. Esta solución personalizada tiene una precisión variable dependiendo del contexto. Por ejemplo, se obtiene un 20% de precisión para el reconocimiento de 5 palabras en una habitación grande, mientras en una habitación pequeña para la misma cantidad de palabras se logra un 60% de precisión.

Otra opción más robusta es utilizar un reconocedor basado en redes neuronales que funcione en la nube. Putthapipat, Woralert y Sirinimnuankul (2018) implementan Google Cloud Platform, plataforma incluye una API de pago para el reconocimiento de voz. Esta API funciona en base a redes neuronales artificiales. El artículo demuestra que la implementación de un reconocedor con funcionamiento en la nube es viable, pudiendo aprovechar así las ventajas de este tipo de soluciones, principalmente un porcentaje de error muy reducido, siendo el de la solución de Google del 4.9% (Protalinski, 2017).

Figura 2.2

Arquitectura de la solución con el uso de Google Cloud Platform



Nota. De *Speech recognition gateway for home automation on open platform*, por Putthapipat, P., Woralert, C. y Sirinimnuankul, P., 2018, In International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

La revisión de la literatura previa muestra como ha ido evolucionando el reconocimiento de voz y, como conforme este avanza, también se va mejorando la aplicación que este tiene en entornos domóticos para personas con discapacidad. En la actualidad, las técnicas basadas en modelos estadísticos son las que ofrecen un mejor rendimiento, destacando dos: modelos ocultos de Markov y redes neuronales artificiales (Hernando, 1993; Rabiner, Juang y Rabiner, 2005). Las redes neuronales artificiales son capaces de ofrecer un mayor reconocimiento de voz; sin embargo, para poder lograrlo necesitan un mayor poder de cómputo, reduciendo así su eficacia en soluciones de bajo costo (Protalinski, 2017; Biewald, 2017; Aqeel-ur-Rehman y Khursheed, 2014). Por otro lado, los modelos ocultos de Markov son capaces de obtener un correcto desempeño en entornos dómoticos de bajo costo, además de no necesitar de internet para un correcto funcionamiento.

CAPÍTULO III: MARCO TEÓRICO

Debido al alcance del presente trabajo, se presentan los fundamentos teóricos de la técnica utilizada para el reconocimiento de voz: los modelos ocultos de Markov y sobre la domótica con especial énfasis a la plataforma utilizada: Arduino.

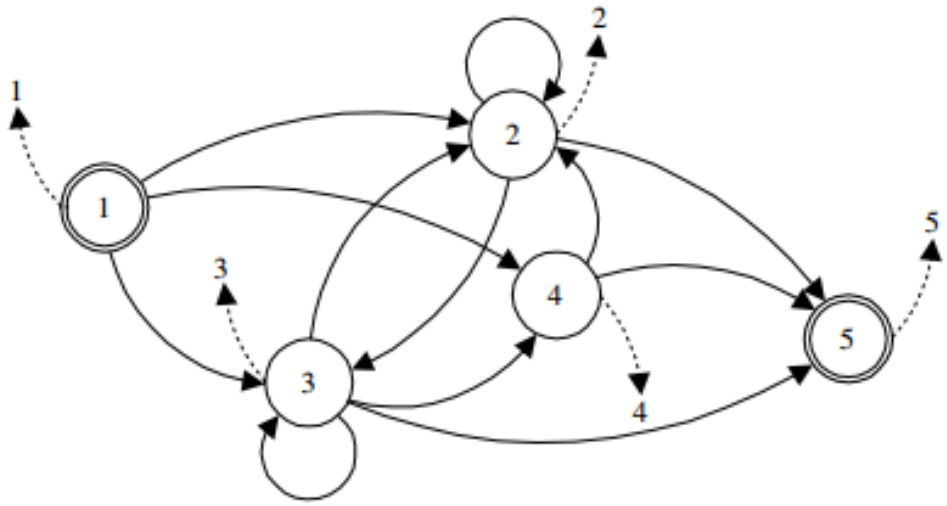
3.1 Cadena de Markov

En primer lugar, para poder entender qué es un modelo oculto de Markov es necesario definir qué es una cadena de Markov, ya que un modelo oculto de Markov es, básicamente, una cadena de Markov con parámetros ocultos (Rabiner, 1989).

Una cadena de Markov es un caso especial de un modelo de autómata finito. Los modelos de autómatas finitos representan secuencias temporales de variables discretas cuando el conjunto de estados es finito. Existe una función de transición de estados que determina la forma en que se realizan los cambios de un estado a otro. Además, cuenta con entradas y salidas. Cada estado tiene asociada una salida para una entrada dada (Rabiner, 1989). Para representar esta estructura se utiliza un diagrama como el de la figura 3.1:

Figura 3.1

Diagrama de estados para un autómata finito



Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla*, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

En el diagrama de la figura 3.1 se puede observar un estado inicial (1), un estado final (5), los estados internos (2, 3 y 4) y las flechas que indican las posibles transiciones entre los estados. También se han representado las salidas de cada estado en líneas de puntos que, para simplificar, coinciden con el número de estado (Milone, 2004).

Para entender el funcionamiento de este tipo de modelos, se brinda un ejemplo sencillo. Para este ejemplo, se asumen dos supuestos: La función salida da como resultado el número del estado y la función transición elige el estado siguiente como el número más cercano a la entrada actual. Si se tiene como entrada la secuencia de números: 2, 2, 2, 4, 4, 4; la secuencia de pasos para determinar la secuencia sería:

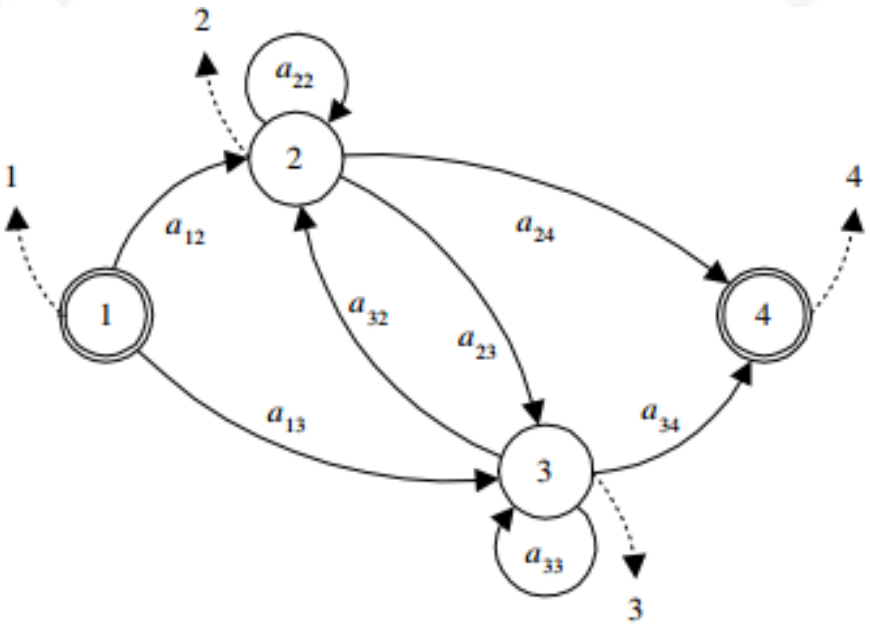
1. La entrada es 2, pero el estado inicial es 1 independientemente de la entrada.
2. La entrada es 2, por lo tanto, el número más cercano es 2.
3. La entrada es 2, por lo tanto, el número más cercano es 2.
4. La entrada es 4. El número más cercano es 4, pero desde el 2 no se puede pasar directamente al 4, por lo tanto, el número más cercano posible es 3.
5. La entrada es 4, por lo tanto, el número más cercano es 4.

6. La entrada es 4, pero el estado final debe ser 5 independientemente de la entrada.

Otro tipo de autómata es el que puede albergar una descripción probabilística de lo que modela. En este caso, la función de transición de estados pasa a ser probabilidad de transición de estados. En el caso de las salidas, cada estado se asocia a uno de los posibles símbolos de un conjunto de salidas. Este modelo se conoce como cadena de Markov (Rabiner, 1989). Para representar estas características se utiliza un diagrama como el de la figura 3.2:

Figura 3.2

Diagrama de estados para un autómata probabilístico



Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla,* por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

Para las probabilidades de transición se utiliza la notación a_{ij} que indica la probabilidad de pasar del estado i (estado actual) al estado j. Estas probabilidades se definen en una matriz (Rabiner, 1989). En la figura 3.3 se presenta una posible matriz de probabilidad para el diagrama de la figura 3.2:

Figura 3.3

Matriz de probabilidades de transición de estados

$$A = \{a_{ij}\} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 1/4 & 1/4 & 1/2 \\ 0 & 1/2 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla,* por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

A diferencia del ejemplo anterior, en las cadenas de Markov se busca la probabilidad de que una determinada secuencia de salidas haya sido determinada por el modelo. En otras palabras, en base a una secuencia observada en el mundo real, se determina cuál es la probabilidad de que el modelo la haya generado (Milone, 2004). Para poder explicar mejor este comportamiento, se plantea un ejemplo con las siguientes características:

- El diagrama utilizado es el de la figura 3.2.
- La matriz de probabilidades de transición del ejemplo es la de la figura 3.3.
- La secuencia de entrada es: 1, 2, 2, 3, 2, 4

Con estos datos, se determina que la transición de estados es: 1 → 2, 2 → 2, 2 → 3, 3 → 2 y 2 → 4. El siguiente paso es buscar estas transiciones en la matriz. Por ejemplo, la transición del estado 1 al estado 2 tiene como probabilidad ½. Finalmente, las probabilidades se multiplican y dan como resultado:

$$p_{122324} = a_{12}a_{22}a_{23}a_{32}a_{24} = \frac{1}{2}\frac{1}{4}\frac{1}{4}\frac{1}{2}\frac{1}{2} = \frac{1}{128}$$

Ecuación 1. Cálculos para obtener la probabilidad para la secuencia 1, 2, 2, 3, 2, 4 (Milone, 2004).

3.2 Modelos ocultos de Markov

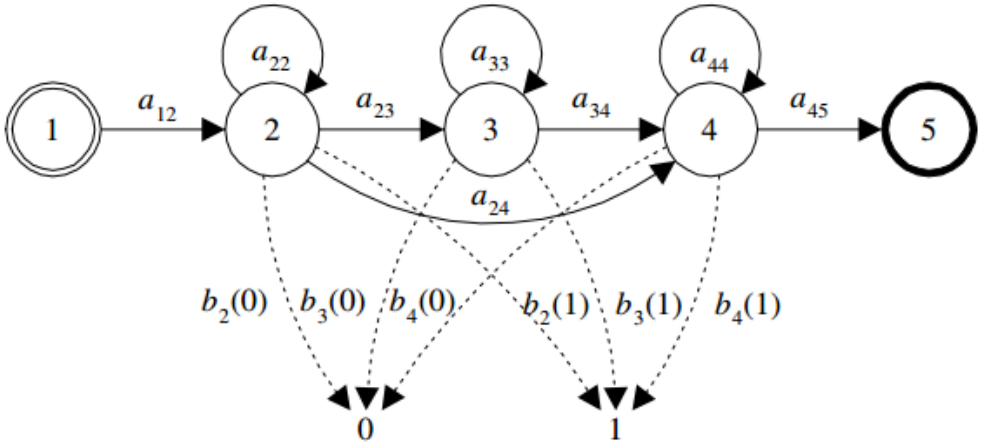
Las cadenas de Markov son también conocidas como modelos observables de Markov, debido a que de acuerdo con una salida se puede observar en qué estado se encuentra el modelo. En el caso de los modelos ocultos de Markov, la función que asocia las salidas a cada estado una salida es una distribución de probabilidades sobre todas las posibles salidas (Rabiner, 1989).

Se agrega un nuevo parámetro: $b_j(k)$ Este parámetro describe la probabilidad de que el estado j observe la salida k dentro del conjunto de salidas y, al igual que el parámetro a_{ij} es definido en una matriz. Esta matriz indica la probabilidad de obtener la salida k desde el estado j. Bajo estas condiciones, no se puede saber con certeza en qué estado se encuentra el modelo sabiendo la salida, pues la salida puede estar dada por cualquier estado. En otras palabras, el funcionamiento interno del modelo queda oculto, motivo por el cual este tipo de modelos se conocen como modelos ocultos de Markov (Milone, 2004).

En la figura 3.4 se muestra un diagrama de estados para un modelo oculto de Markov comúnmente utilizado en reconocimiento del habla. El estado inicial (estado 1) y el estado final (estado 5) se denominan estados no emisores. Las posibles transiciones entre estados se indican mediante las flechas de líneas continuas; mientras que las probabilidades de observación para cada estado se indican mediante las flechas en líneas de puntos (Milone, 2004).

Figura 3.4

Diagrama de estados para un modelo oculto de Markov comúnmente utilizado en reconocimiento del habla



Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla,* por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

Para el reconocimiento del habla, los modelos ocultos de Markov tienen una estructura simple de izquierda a derecha como se puede apreciar en la figura 3.4 (Milone, 2004). Para poder entender mejor estos conceptos, se muestra a continuación el funcionamiento de este tipo de modelo mediante un ejemplo basado en el diagrama de la figura 3.4. Las probabilidades de a_{ij} y $b_j(k)$ se muestran en la figura 3.5:

Figura 3.5

Probabilidades de a_{ij} y $b_j(k)$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1/3 & 1/5 & 2/3 & 0 \\ 0 & 2/3 & 4/5 & 1/3 & 0 \end{bmatrix}$$

Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla,* por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

La probabilidad que se busca es la de obtener la secuencia 0, 0, 1, 0 (solo existen estas dos salidas como se puede observar en la figura 3.4). A diferencia del ejemplo mostrado en la figura 3.4, ahora cualquier estado emisor puede emitir cualquier salida, por lo que es necesario analizar todas las posibilidades. Estas se listan en la tabla 3.1:

Tabla 3.1

Cálculo de probabilidades para la secuencia de salida 0, 0, 1, 0

Secuencias de estados	Probabilidades de transición	Probabilidades de observación	Probabilidades de la secuencia
1, 2, 2, 2, 4, 5	$1\ \frac{1}{4}\frac{1}{4}\frac{1}{2}\frac{1}{2}=\frac{1}{64}$	$\frac{1}{3}\frac{1}{3}\frac{2}{3}\frac{2}{3}=\frac{4}{81}$	$\frac{1}{1296}$
1, 2, 2, 3, 4, 5	$1\ \frac{1}{4}\frac{1}{4}\frac{1}{2}\frac{1}{2}=\frac{1}{64}$	$\frac{1}{3}\frac{1}{3}\frac{4}{5}\frac{2}{3}=\frac{8}{135}$	$\frac{1}{1080}$
1, 2, 2, 4, 4, 5	$1\ \frac{1}{4}\frac{1}{2}\frac{1}{2}\frac{1}{2}=\frac{1}{32}$	$\frac{1}{3}\frac{1}{3}\frac{1}{3}\frac{2}{3}=\frac{2}{81}$	$\frac{1}{1296}$
1, 2, 3, 3, 4, 5	$1\ \frac{1}{4}\frac{1}{2}\frac{1}{2}\frac{1}{2}=\frac{1}{32}$	$\frac{1}{3}\frac{1}{5}\frac{4}{5}\frac{2}{3}=\frac{8}{225}$	$\frac{1}{900}$
1, 2, 3, 4, 4, 5	$1\ \frac{1}{4}\frac{1}{2}\frac{1}{2}\frac{1}{2}=\frac{1}{32}$	$\frac{1}{3}\frac{1}{5}\frac{1}{3}\frac{2}{3}=\frac{2}{135}$	$\frac{1}{2160}$
1, 2, 4, 4, 4, 5	$1\ \frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2}=\frac{1}{16}$	$\frac{1}{3}\frac{2}{3}\frac{1}{3}\frac{2}{3}=\frac{4}{81}$	$\frac{1}{324}$
Probabilidad Total			$\sum=\frac{77}{10800}\sim 0{,}007$

Nota. Adaptado de *Modelos ocultos de Markov para el reconocimiento automático del habla*, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

En la columna ‘Secuencia de estados’ de la tabla 3.1 se muestran todas las secuencias de estado que pueden dar como salida 0, 0, 1, 0. Las probabilidades de transición y observación se calculan como en una cadena de Markov, es decir, se van multiplicando las probabilidades obtenidas de la matriz. La probabilidad de la secuencia se obtiene de multiplicar la probabilidad de transición y la probabilidad de observación. Finalmente, se toma la probabilidad de secuencia más alta como la respuesta correcta (Milone, 2004).

En el ejemplo se ha asumido que existe un estado inicial por el que el modelo debe iniciar sí o sí, pero esto no necesariamente es así. La probabilidad de iniciar en un estado se define como $\pi = \{\pi_i\}$, siendo esto una matriz que define la probabilidad de iniciar en el estado i (Antón, 2015).

3.2.1 Retos de los modelos ocultos de Markov

- Evaluación: Cómo calcular eficientemente la probabilidad de la secuencia de observación producida por un modelo. Esta probabilidad se determina mediante el algoritmo forward-backward. Este algoritmo se divide en 2 partes: forward y backward. Solo es necesario el algoritmo forward para solucionar este problema (Antón, 2015; Oropeza y Suárez, 2006; Rabiner, 1989).
- Decodificación: Cómo calcular la secuencia de estados óptima dada una secuencia de observación. Este inconveniente se resuelve mediante el algoritmo de Viterbi (Oropeza y Suárez, 2006).
- Entrenamiento. - Cómo ajustar los parámetros del modelo λ= (A, B, π) para maximizar P(O/λ). El objetivo es optimizar los parámetros del modelo de manera que describan de la mejor forma posible la secuencia de observación dada. La secuencia de observación usada es llamada secuencia de entrenamiento, porque, valga la redundancia, es usada para entrenar al modelo dentro de la fase de entrenamiento. Este problema se soluciona mediante el uso del algoritmo Baum-Welch (Oropeza y Suárez, 2006).

3.2.1.1 Algoritmo Forward

Si bien es posible calcular la probabilidad como se menciona en un ejemplo anterior, ese cálculo requiere un número elevado de operaciones, haciéndolo inviable e ineficiente. Este algoritmo calcula la probabilidad de estar en un estado en un tiempo t teniendo el conjunto de observaciones hasta el instante t, a diferencia de Viterbi que se encarga de devolver la secuencia de estados más probables de acuerdo con las observaciones. El algoritmo forward soluciona el problema en tres pasos (Rabiner, 1989):

- Inicialización: Se inicializa la variable forward de manera que represente la probabilidad de observar la secuencia parcial hasta el instante t y estar en el estado S_i:

$$\alpha_t(i) = P\left(O_1 O_2 \cdots O_t,\, q_t = S_i \,|\, \lambda\right)$$

Ecuación 2. Inicialización de la variable Forward (Oropeza y Suárez, 2006).

En caso t sea igual a 1, se dará el siguiente comportamiento:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

Ecuación 3. Inicialización de la variable Forward cuando t=1 (Oropeza y Suárez, 2006).

- Inducción: En este paso se calculan las variables forward en el instante t + 1 a partir de las variables forward en el instante t, de las probabilidades de transición y probabilidades de observación. Este comportamiento se da cuando 1 <= t <= T-1.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1})$$

Ecuación 4. Cálculo de la variable Forward en el instante t+1 (Oropeza y Suárez, 2006).

- Finalización: La probabilidad deseada se calcula como suma de las probabilidades máximas en cada estado hacia delante en el último instante posible, T.

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Ecuación 5. Cálculo de la probabilidad total (Oropeza y Suárez, 2006).

El ahorro de operaciones se da porque, al elegir el estado más probable, automáticamente se descartan los otros estados y sus respectivos caminos, reduciendo de forma considerable el número de cálculos.

3.2.1.2 Algoritmo de Viterbi

Este algoritmo tiene como objetivo elegir el camino completo de estados con mayor probabilidad. Milone (2004) señala: "En este algoritmo la idea central es recorrer el diagrama de transiciones de estados a través del tiempo, almacenando para cada estado solamente la máxima probabilidad acumulada y el estado anterior desde el que se llega con esta probabilidad" (p. 10).

El algoritmo cuenta con fases:

- Inicialización: Se inicializa la variable auxiliar $\delta_t(i)$, variable que representa la mayor probabilidad obtenida a través de una secuencia única de estados hasta llegar en el instante t, al estado i. Esta variable permite calcular la probabilidad en el instante t+1, si y solo si, se conocen todos los estados en el instante t. Además, se inicializa otra variable $\varphi_1(i)$ que almacena las máximas probabilidades obtenidas.

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

Ecuación 6. Inicialización de la variable $\boldsymbol{\delta_t(i)}$ (Antón, 2015).

$$\varphi_1(i) = 0$$

Ecuación 7. Inicialización de la variable $\boldsymbol{\varphi_1(i)}$ (Antón, 2015).

- Recursión: Se guardan los valores máximos que se van obteniendo, que servirán para obtener el camino óptimo. Los valores máximos se van guardando en la variable φ_1(i). Esta variable se va actualizando conforme se encuentren nuevos máximos y el camino de las probabilidades menores se descarta (ya no se recorre). El cálculo de la máxima probabilidad en t+1 se da de la siguiente manera:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i)\, a_{ij}\right] b_j\,(O_t)\,, \qquad 2 \leq t \leq T, 1 \leq j \leq N$$

Ecuación 8. Cálculo de la probabilidad máxima (Antón, 2015).

$$\varphi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}], \qquad 2 \le t \le T,\ 1 \le j \le N$$

Ecuación 9. Asignación de la probabilidad máxima a la variable $\boldsymbol{\varphi_1(i)}$ (Antón, 2015).

- Finalización. - Se utiliza la función arg max para obtener la secuencia de estados más probable, a diferencia del algoritmo forward, donde solo se sumaban las probabilidades. La función se aplica sobre la variable auxiliar $\delta_t(i)$.

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]$$

Ecuación 10. Cálculo de la probabilidad final (Antón, 2015).

- Back tracking. - Etapa donde se reconstruye la secuencia de estados más probable realizando el camino desde el instante final al inicial, adoptando los valores máximos por cada paso de la etapa de recursión, de manera que se obtenga la secuencia óptima.

Pseudocódigo

Definición de variables:

- Conjunto de observaciones: $O = \{o_1, o_2, \ldots, o_N\}$
- Conjunto de estados: $S = \{s_1, s_2, \ldots, s_K\}$
- Probabilidades iniciales: $\Pi = (\pi_1, \pi_2, \ldots, \pi_K)$
- Secuencia de observaciones: $Y = (y_1, y_2, \ldots, y_T)$
- Matriz de transición: A de tamaño K x K
- Matriz de emisiones: B de tamaño K x N que muestra la probabilidad de observar o_j desde el estado s_i

- Secuencia más probable: $X = (x_1, x_2, \ldots, x_N)$
- Matriz auxiliar T1 que guarda la máxima probabilidad de la secuencia más probable hasta el momento j. Tamaño K x T.
- Matriz auxiliar T2 que guarda la secuencia de estados más probables hasta el momento j. Tamaño K x T.

Función VITERBI (O, S, π, Y, A, B)

Para cada estado j := 1 hasta K hacer — Inicialización

T1 [j,1] := $\pi_j \; x \; B_{jy1}$

T2 [j,1] := 0

Fin

Para cada observación i := 2 hasta T hacer — Recursión

Para cada estado j := 1 hasta T hacer

T1 [j,i] := $\max_k (T_1[k, i-1] \; x \; A_{kj} \; x \; B_{jy1})$

T2 [j,1] := $\operatorname{argmax}_k (T_1[k, i-1] \; x \; A_{kj} \; x \; B_{jy1})$

Fin

Fin

$Z_T := \operatorname{argmax}_k (T_1[k, T])$ — Finalización

$X_T := S_{Z_T}$

Para i := T, T-1, T-2 hasta 2 hacer — Back tracking

Z_{T-1} := T2 [Z_1,1]

$X_{i-1} := S_{Z_{i-1}}$

Fin

Retornar X

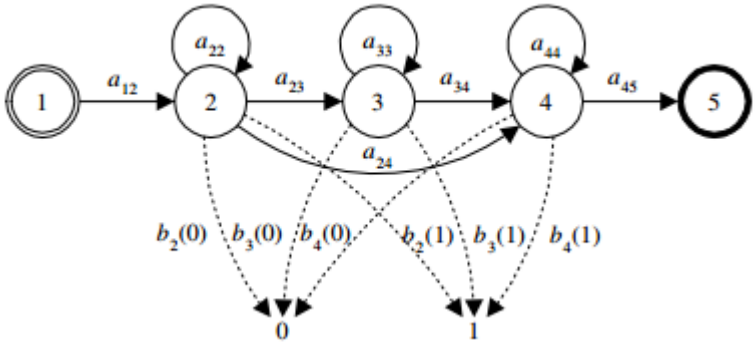
Fin de la función

La complejidad del algoritmo de Viterbi es igual a la del algoritmo forward, debido a su similitud (Antón, 2015). Para un mejor entendimiento del algoritmo y sus

ventajas, se muestra a continuación un ejemplo basado en el diagrama de la figura 3.4 y sus probabilidades. También la secuencia de observaciones será la misma: 0, 0, 1, 0 Las condiciones se vuelven a mostrar en la figura 3.6 para ayudar a la explicación:

Figura 3.6

Condiciones del ejemplo



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1/3 & 1/5 & 2/3 & 0 \\ 0 & 2/3 & 4/5 & 1/3 & 0 \end{bmatrix}$$

Nota. De *Modelos ocultos de Markov para el reconocimiento automático del habla*, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

El primer paso es único, porque la única transición posible es del estado 1 al estado 2.

$$p_{12} = b_2(0)\,[p_1 a_{12}] = \frac{1}{3}[1 \times 1] = \frac{1}{3}$$

Ecuación 11. Cálculo de la probabilidad de pasar del estado 1 al estado 2 (Milone, 2004).

Desde el estado 2 se puede pasar al 2, al 3 o al 4 obteniendo:

$$p_{122} = b_2(0)\,[p_{12}a_{22}] = \frac{1}{3}\left[\frac{1}{3}\frac{1}{4}\right] = \frac{1}{36}$$

$$p_{123} = b_3(0)\,[p_{12}a_{23}] = \frac{1}{5}\left[\frac{1}{3}\frac{1}{4}\right] = \frac{1}{60}$$

$$p_{124} = b_4(0)\,[p_{12}a_{24}] = \frac{2}{3}\left[\frac{1}{3}\frac{1}{2}\right] = \frac{1}{9}$$

Ecuación 12. Cálculos de las probabilidades de pasar del estado 2 a los estados 2, 3 o 4 en t1 (Milone, 2004).

Desde el estado 2 en el tiempo t2 se puede pasar a los estados 2, 3, y 4:

$$p_{1222} = b_2(1)\,[p_{122}a_{22}] = \frac{2}{3}\left[\frac{1}{36}\frac{1}{4}\right] = \frac{1}{216}$$

$$p_{1223} = b_3(1)\,[p_{122}a_{23}] = \frac{4}{5}\left[\frac{1}{36}\frac{1}{4}\right] = \frac{1}{180}$$

$$p_{1224} = b_4(1)\,[p_{122}a_{24}] = \frac{1}{3}\left[\frac{1}{36}\frac{1}{2}\right] = \frac{1}{216}$$

Ecuación 13. Cálculos de las probabilidades de pasar del estado 2 a los estados 2, 3 o 4 en t2 (Milone, 2004).

Desde el estado 3 en tiempo t2 se puede pasar a los estados 3 y 4:

$$p_{1233} = b_3(1)\,[p_{123}a_{33}] = \frac{4}{5}\left[\frac{1}{60}\frac{1}{2}\right] = \frac{1}{600}$$

$$p_{1234} = b_4(1)\,[p_{123}a_{34}] = \frac{1}{3}\left[\frac{1}{60}\frac{1}{2}\right] = \frac{1}{360},$$

Ecuación 14. Cálculos de las probabilidades de pasar del estado 3 a los estados 3 o 4 en t2 (Milone, 2004)

Desde el estado 4 en el tiempo t2 solo se puede pasar al estado 4:

$$p_{1244} = b_4(1)\left[p_{124}a_{44}\right] = \frac{1}{3}\left[\frac{1}{9}\frac{1}{2}\right] = \frac{1}{54}$$

Ecuación 15. Cálculos de las probabilidades de pasar del estado 4 a los estados 4 en t2 (Milone, 2004).

Al llegar al tiempo t3, desde cualquiera de los estados solo se puede llegar al estado 4. La función máx elige entre las probabilidades de caminos anteriormente calculados, devolviendo solo la probabilidad más alta:

$$\begin{aligned}
p_{12224} &= b_4(0)\left[p_{1222}a_{24}\right] = \frac{1}{3}\left[\frac{1}{216}\frac{1}{2}\right] = \frac{1}{1296} \\
p_{12?34} &= b_4(0)\,\text{máx}\left\{\left[p_{1223}a_{34}\right]\left[p_{1233}a_{34}\right]\right\} \\
&= b_4(0)\,\text{máx}\left\{p_{1223}, p_{1233}\right\}a_{34} \\
&= \frac{1}{5}\,\text{máx}\left\{\frac{1}{216}, \frac{1}{600}\right\}\frac{1}{2} \\
&= \frac{1}{5}\frac{1}{216}\frac{1}{2} = \frac{1}{2160} \\
&= p_{12234} \\
p_{12?44} &= b_4(0)\,\text{máx}\left\{\left[p_{1224}a_{44}\right]\left[p_{1234}a_{44}\right]\left[p_{1244}a_{44}\right]\right\} \\
&= b_4(0)\,\text{máx}\left\{p_{1224}, p_{1234}, p_{1244}\right\}a_{44} \\
&= \frac{2}{3}\,\text{máx}\left\{\frac{1}{216}, \frac{1}{360}, \frac{1}{54}\right\}\frac{1}{2} \\
&= \frac{1}{5}\frac{1}{54}\frac{1}{2} = \frac{1}{162} \\
&= p_{12444}
\end{aligned}$$

Ecuación 16. Cálculos de las probabilidades de pasar de algún estado al estado 4 en t3 (Milone, 2004).

Finalmente, en el tiempo t4, la única opción posible es pasar al estado 5 que, como el estado 1, es un estado no emisor. Por lo tanto, la probabilidad de observación del estado 5 no debe ser considerada.

$$
\begin{aligned}
p_{12??45} &= \max\{[p_{12224}a_{45}]\,[p_{12234}a_{45}]\,[p_{12444}a_{45}]\} \\
&= \max\{p_{12224}, p_{12234}, p_{12444}\}\,a_{45} \\
&= \max\left\{\frac{1}{1296}, \frac{1}{2160}, \frac{1}{162}\right\}\frac{1}{2} \\
&= \frac{1}{162}\frac{1}{2} = \frac{1}{324} \\
&= p_{124445}
\end{aligned}
$$

Ecuación 17. Cálculo de la probabilidad de pasar del estado 4 al estado 5 (Milone, 2004).

Como resultado, se llega a la misma secuencia que en el ejemplo anterior (1, 2, 4, 4, 4, 5); pero, a diferencia del análisis anterior, se han realizado solo 27 cálculos en vez de 48.

3.2.1.3 Algoritmo Baum-Welch

Este algoritmo, usado para el entrenamiento del modelo $\lambda = (A, B, \pi)$, trata de definir los parámetros del modelo que maximizan las probabilidades de observación $P(O \mid \lambda)$. Es un caso particular del algoritmo Expectation-Maximization (EM) aplicado a los modelos ocultos de Markov (Antón, 2015). El algoritmo EM sigue la siguiente lógica en cada una de sus partes:

- Expectation: Tiene como objetivo el número de transiciones del estado i en la secuencia de observaciones O y el número esperado de transiciones desde el estado i al estado j en la secuencia O.
- Maximization: Consiste en maximizar la función en base a los parámetros calculados en el paso anterior, lo que resulta en otros parámetros nuevos que describen mejor la secuencia de observación.

En otras palabras, lo que hace el algoritmo es comenzar por usar un modelo con unos parámetros al azar, con esos parámetros se calculan: el número de transiciones del estado i en la secuencia de observaciones O y el número esperado de transiciones desde el estado i al estado j en la secuencia O. Una vez se haya realizado este paso, se vuelven a reestimar los parámetros en base a lo obtenido. Esto se repite hasta que ya no haya mejora.

Los pasos que sigue el algoritmo de Baum-Welch y, por lo tanto, el entrenamiento son los siguientes:

1. Selección de un modelo inicial elegido aleatoriamente.
2. Cálculo de las transiciones más probables con el algoritmo forward.
3. Cálculo de las emisiones más probables con el algoritmo backward.
4. Construcción de un nuevo modelo en el que se incrementen las posibilidades calculadas por el algoritmo forward-backward.

Estos pasos se repiten hasta que la mejora en el modelo sea mínima o no exista (Antón, 2015).

Baum-Welch se apoya en el algoritmo Forward-Backward para su funcionamiento. Este último algoritmo se encarga de calcular las probabilidades de los estados ocultos de un HMM dada una secuencia de observaciones (Oropeza y Suárez, 2006). Ambos algoritmos se describen de la siguiente manera:

Algoritmo Forward-Backward

$$\alpha_t^{(rj)} = \left[\sum_i \alpha_{t-1}^{(ri)} a_{ij}\right] b_j\left(\boldsymbol{y}_t^{(r)}\right)$$

$$\beta_t^{(ri)} = \left[\sum_j a_{ij} b_j(\boldsymbol{y}_{t+1}^{(r)}) \beta_{t+1}^{(rj)}\right]$$

Algoritmo 1: Algoritmo de Forward-Backward. La primera ecuación corresponde a la parte forward, mientras que la segunda corresponde a la parte backward.

Baum-Welch

Definición de $\xi_t(i,j)$ como la probabilidad de estar en el estado i en el instante t y en el estado j en el instante t+1.

$$\varepsilon_t(i,j) = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})}$$

Ecuación 18. Definición de $\xi_t(i,j)$ (Oropeza y Suárez, 2006).

Reestimación de las probabilidades de transición.

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$1 \leq i \leq N, 1 \leq j \leq N$$

Ecuación 19. Reestimación de probabilidades de transición (Oropeza y Suárez, 2006).

Reestimación de las probabilidades de emisión.

$$\bar{b}_j(o_k) = \frac{\sum_{t=1:o_t=o_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$1 \leq j \leq N, 1 \leq k \leq N$$

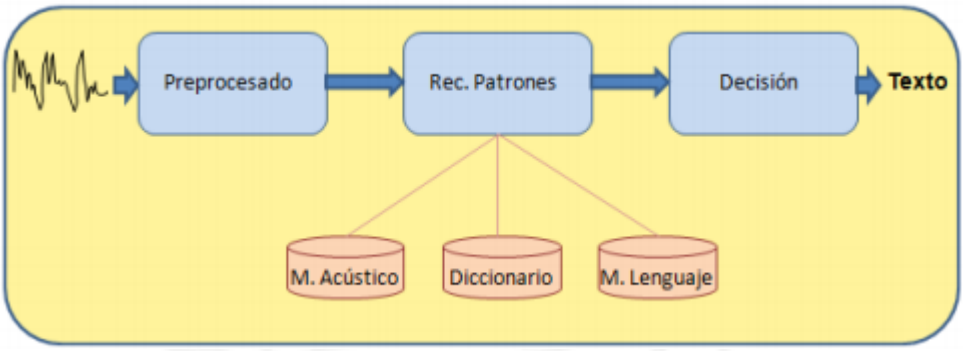
Ecuación 20. Reestimación de probabilidades de emisión (Oropeza y Suárez, 2006).

3.3 Reconocedor de voz basado en modelos ocultos de Markov

Un reconocedor de voz es un sistema que tiene como objetivo capturar un mensaje de voz recibido y transcribirlo a texto. Estos sistemas presentan una arquitectura y se apoyan en distintos modelos estadísticos y metodologías para desarrollar su labor Antón (2015).

Figura 3.7

Esquema básico de la arquitectura de un reconocedor del habla

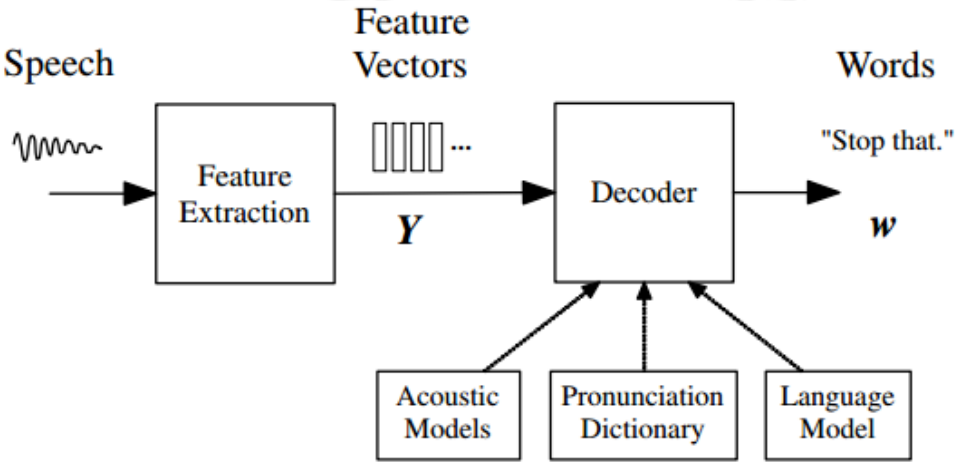


Nota. De *Desarrollo de un sistema de reconocimiento de habla natural independiente del locutor*, por Antón, J., 2015, [tesis de licenciatura, Universidad Autónoma de Madrid]. Departamento de Tecnología Electrónica y de las Comunicaciones, España. https://repositorio.uam.es/handle/10486/667850

La figura 3.7 describe de manera simplificada la estructura de un reconocedor del habla basado en modelos ocultos de Markov. Este se divide en 3 partes principales: Pre procesado, reconocimiento de patrones y decisión. Otra representación se presenta en la figura 3.8:

Figura 3.8

Diagrama de la arquitectura de un reconocedor del habla



Nota. De *The application of hidden Markov models in speech recognition*, por Gales, M. y Young, S., 2008, Foundations and trends in signal processing, 1(3), 195-304.

Como se puede observar en la figura 3.8, se omite el paso de decisión, pero la salida es la misma: palabras. Esto nos da a entender que las partes fundamentales de un reconocedor de voz son las siguientes:

- Entrada: Voz emitida por una persona.
- Pre procesado
- Reconocimiento de patrones
- Salida: palabras identificadas.

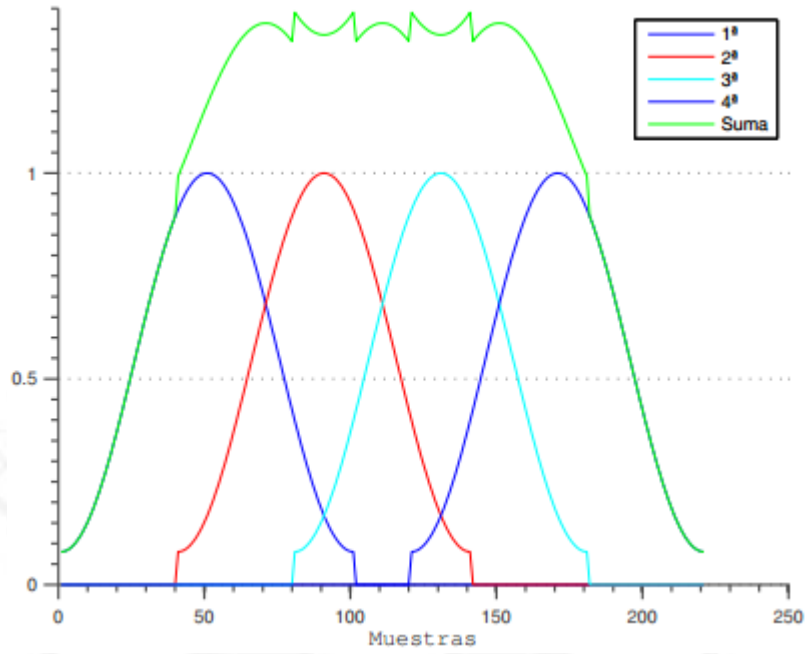
3.3.1 Preprocesamiento

Este paso tiene como objetivo principal la conversión del audio de entrada a una secuencia de vectores acústicos de tamaño fijo. Este proceso también es llamado extracción de características (Gales y Young, 2008).

Los vectores acústicos que se quieren obtener son típicamente computados cada 10 milisegundos usando enventanado superpuesto de alrededor de 25 milisegundos. Es decir, la ventana se completa es de 25 ms y el desplazamiento de 10 ms, que corresponde a un 60% de solape entre ventanas. Al momento de obtener la señal, es posible que ocurran discontinuidades en los extremos, por lo que el espectro obtenido no representaría el espectro de la señal original. Esto se soluciona utilizando ventanas, es decir, funciones matemáticas que disminuyen estas distorsiones. Para la obtención de los vectores acústicos se utiliza una ventana de tipo Hamming y el solapamiento antes mencionado se da porque, al ser irregular la forma de la ventana de Hamming, las muestras en los extremos sufren una ponderación que es anulada gracias al solapamiento (Antón, 2015; Gales y Young, 2008).

Figura 3.9

Ejemplo de ventanas de Hamming solapadas un 60%.



Nota. De *The application of hidden Markov models in speech recognition*, por Gales, M. y Young, S., 2008, Foundations and trends in signal processing, 1(3), 195-304.

Para conseguir los vectores mencionados anteriormente, se suele usar un esquema de codificación basado en Coeficientes Cepstrales en las Frecuencias de Mel (MFCC, proveniente del inglés Mel Frequency Cepstral Coefficients). Estos se generan aplicando la transformada de coseno discreta (DTC, proveniente del inglés Discrete Cosine Transform) truncada a un registro estimado del espectro calculado por una transformada rápida de Fourier (Gales y Young, 2008). Además, La señal de entrada se atenúa a medida que aumenta la frecuencia, por lo que al inicio se aplica un filtro de tipo Respuesta Finita al Impulso (FIR, proveniente del inglés Finite Impulse Response) (Antón, 2015).

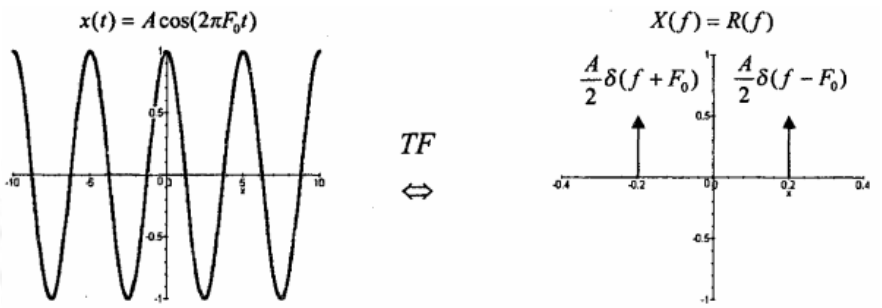
3.3.1.1 Transformada discreta de Fourier

Una transformada de Fourier "transforma una señal representada en el dominio del tiempo al dominio de la frecuencia, pero sin alterar su contenido de información, sólo es una forma diferente de representarla" (Bernal, Gómez y Bobadilla, 1999, p. 3). Además, es posible realizar el proceso inverso mediante la transformada inversa de Fourier (Antón,

2015). Gráficamente se visualiza como en la figura 3.10 donde la función inicial, en dominio del tiempo, se muestra en la imagen izquierda; mientras que la función en dominio de la frecuencia, obtenida por la transformada de Fourier, se muestra en la imagen derecha.

Figura 3.10

Transformada de Fourier de una función básica

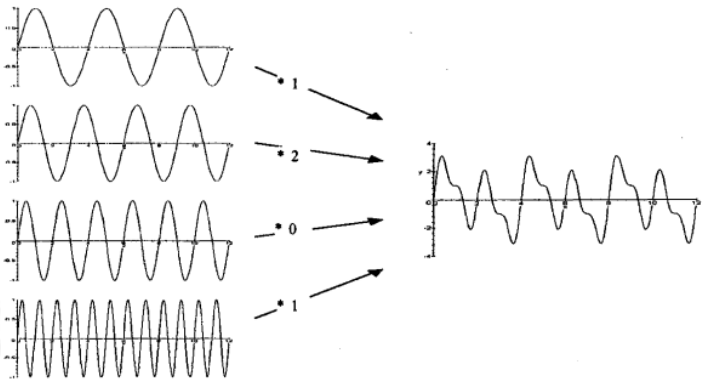


Nota. De *Una visión práctica en el uso de la Transformada de Fourier como herramienta para el análisis espectral de la voz,* por Bernal, J., Gómez, P. y Bobadilla, J., 1999, Estudios de fonética experimental, 10, 75-105.

La base de la transformada de Fourier es: "Toda señal genérica, por compleja que sea se puede descomponer en una suma de funciones periódicas simples de distinta frecuencia. En definitiva, la Transformada de Fourier visualiza los coeficientes de las funciones sinusoidales que forman la señal original" (Bernal, Gómez y Bobadilla, 1999, p. 4). Este proceso devuelve una proporción de los coeficientes que se utilizaron para crear la señal (Bernal, Gómez y Bobadilla, 1999).

Figura 3.11

Señal genérica formada por una sumatoria de señales sinusoidales



Nota. De *Una visión práctica en el uso de la Transformada de Fourier como herramienta para el análisis espectral de la voz,* por Bernal, J., Gómez, P. y Bobadilla, J., 1999, Estudios de fonética experimental, 10, 75-105.

La transformada de Fourier es una herramienta muy útil cuando se trabaja con modelos matemáticos, pero que presenta complicaciones cuando se quiere trabajar con señales reales físicas. Para solucionar este inconveniente se pasa a trabajar con modelos finitos y discretos (Bernal, Gómez y Bobadilla, 1999).

El primer paso para lograr esto es realizar el muestreo de la señal de voz. Es fundamental conocer la frecuencia de muestreo que se aplicará y para ello es necesario saber el ancho de banda de la onda original. Una vez se encuentre muestreada la señal, se procede a convertirla en finita. Para ello se debe limitar el número de puntos a tomar. Matemáticamente, esto se logra al multiplicar la señal por una ventana temporal como puede ser la ventana de Hamming mencionada en el punto anterior. Esto aún sigue dando como resultado un espectro continuo donde un grupo de muestran valen cero, así que como último paso se procede a limitar el número de puntos que se toma, lo que provoca un espectro discreto (Bernal, Gómez y Bobadilla, 1999).

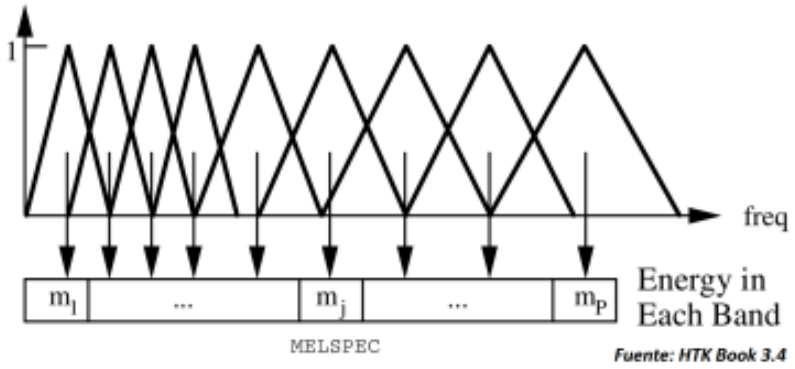
Si bien es posible utilizar la transformada discreta de Fourier, en la práctica se utiliza el algoritmo de la transformada rápida de Fourier. Este algoritmo permite:

> Realizar la transformada discreta de Fourier y su inversa con un coste computacional mucho menor.

> La salida de la transformada de Fourier se filtra con un banco de filtros en la escala Mel, siendo las características de éstos inspiradas en el proceso auditivo humano. (Antón, 2015, p. 11)

Figura 3.12

Banco de filtros en escala Mel



Nota. De *Desarrollo de un sistema de reconocimiento de habla natural independiente del locutor*, por Antón, J., 2015, [tesis de licenciatura, Universidad Autónoma de Madrid]. Departamento de Tecnología Electrónica y de las Comunicaciones, España. https://repositorio.uam.es/handle/10486/667850

3.3.1.2 Dominio Cepstral

Los pasos previos dan como resultado un vector de, frecuentemente, 13 coeficientes de frecuencia Mel (MFCC, por sus siglas en inglés). Según Antón (2015), estos se obtienen al: "aplicar un filtro de decorrelación homomórfica (Cepstrum) mediante la transformada inversa de Fourier del logaritmo del espectro de potencias, llevando los coeficientes cepstrales al dominio de la frecuencia y obteniendo así coeficientes cepstrales" (p. 12).

3.3.1.3 Vector Quantization

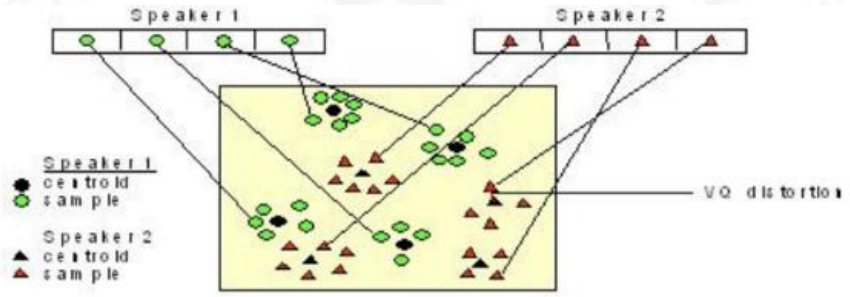
Técnica que se utiliza para aplicar los modelos ocultos de Markov al reconocimiento de voz. Es una parametrización de la señal de voz en vectores dentro de un espacio infinito. En este espacio infinito se construye un codebook de vectores únicos que representan unidades básicas de voz (fonemas) y sus variantes. Los vectores obtenidos de la señal de voz son asignados a estos vectores únicos que representan fonemas con la idea de asignar

un fonema a cada señal de voz y ahorrar recursos (Martínez, Portale, Klein y Olmos, s. f.).

El espacio vector queda dividido en regiones, cada una con un vector representativo llamado centroide. Cada vector que se ubique dentro de una región es asignado al respectivo centroide. La cantidad de regiones es definida por el usuario. En la figura 3.13 se puede observar un ejemplo en el que los puntos de color negro son los centroides, mientras que los puntos de otros colores son las entradas que emiten los usuarios (voz) (Antón, 2015; Srinivasan, 2015).

Figura 3.13

Ejemplo del uso de vector quantization



Nota. De *Speech recognition using Hidden Markov model*, por Srinivasan, A., 2011, Applied Mathematical Sciences, 5(79), 3943-3948.

3.3.2 Reconocimiento de patrones

Dentro del reconocimiento de patrones se identifican 2 etapas distintas: Entrenamiento y comparación. Ambas etapas no se dan únicamente para los modelos ocultos de Markov, sino que también, por ejemplo, se dan cuando se usan redes neuronales (Clemente et al., 2001).

3.3.2.1 Entrenamiento

Etapa que consiste en la construcción de modelos que representen las palabras y/o subunidades (sílabas, fonemas, entre otros) que se quieren reconocer al momento del habla (Antón, 2015, p. 35).

Estos patrones de referencia y asociaciones se realizan sobre modelos ocultos de Markov. Básicamente, lo que se hace es buscar que modelo representa mejor la entrada (audio) que se ingresa. Para realizar esta tarea, se utiliza el algoritmo Baum–Welch, como ya se mencionó en el punto 3.3. Además, se utiliza Vector Quantization para definir un espacio con sus regiones con las que comparar los vectores de entrada obtenidos en el preprocesamiento.

Cuenta con algunos pasos previos y posteriores que complementan esta fase y son necesarios para garantizar un entrenamiento adecuado. Las fases del entrenamiento se muestran en la figura 3.14:

Figura 3.14

Fases del entrenamiento

Nota. Adaptado de *Entrenamiento y Evaluación de reconocedores de Voz de Propósito General basados en Redes Neuronales feed-forward y Modelos Ocultos de Markov*, por Clemente et al., 2001, [tesis de licenciatura, Universidad de las Américas]. Puebl, Dept. Computer Systems Engineering, México.

3.3.2.2 Preparación de datos

Paso necesario antes de realizar el entrenamiento en sí. Se requieren archivos de voz asociados con transcripciones de palabras o fonemas (corpus es el conjunto de varios de estos). Luego, estos archivos se juntan y se crea un Master Label File (MLF), que, en resumen, es un archivo que contiene la información de todos los otros archivos juntos (Clemente et al., 2001).

Una vez se tiene el MLF, el siguiente paso es reconocer la arquitectura del reconocedor. La arquitectura define las unidades básicas usadas durante el reconocimiento y pueden ser: palabras, sílabas, fonemas o subfonemas. Esto se puede definir usando el lenguaje de configuración del CSLU-HMM (Clemente et al., 2001).

3.3.2.3 Evaluación

Una vez se terminen de procesar todos los archivos, se comparan las respuestas obtenidas del reconocedor contra las transcripciones de salida con el objetivo de evaluar el desempeño del reconocedor (Clemente et al., 2001).

3.3.3 Modelo acústico, diccionario fonético y modelo de lenguaje

3.3.3.1 Modelo acústico

Los modelos ocultos de Markov no pueden trabajar con la señal continua, por lo que es necesaria una discretización de la señal en dos niveles: los sucesos deben ocurrir en intervalos discretos de tiempo y las manifestaciones de dichos sucesos deben estar dentro de un conjunto finito de símbolos. Estos símbolos son los fonemas, es decir, los sonidos que se utilizan para formar las palabras (Milone, 2004).

3.3.3.2 Diccionario Fonético

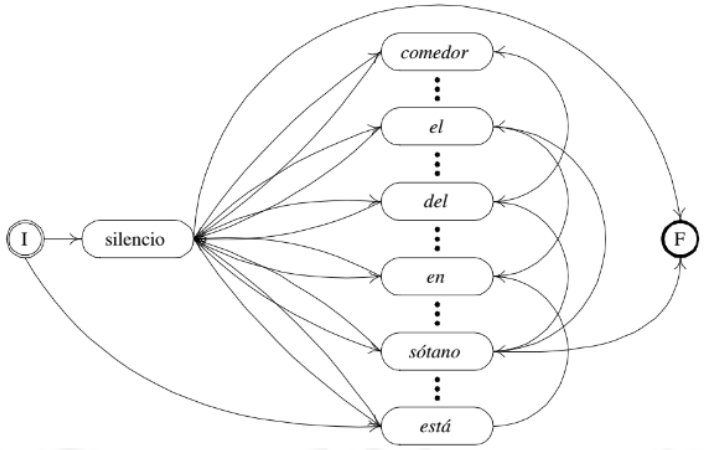
Gracias al modelo acústico se puede modelar un nuevo modelo a nivel fonético. Las descripciones fonéticas de cada palabra son conocidas como diccionario fonético. El modelo generado está a un nivel más alto que el modelo acústico (Milone, 2004).

3.3.3.3 Modelo de Lenguaje

Es el modelo que se ubica en el nivel más alto. Se enfoca en las palabras y la forma en que se combinan para formar frases (Milone, 2004).

Figura 3.15

Modelo de lenguaje



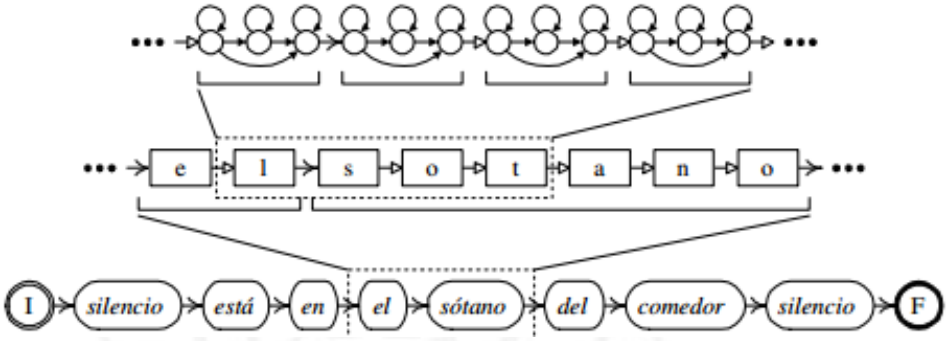
Nota. Los estados de inicio y finalización se indican con las letras I y F, respectivamente. De *Modelos ocultos de Markov para el reconocimiento automático del habla*, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

3.3.3.4 Modelo compuesto

El modelo compuesto es la combinación del modelo acústico, el diccionario fonético y el modelo de lenguaje. Gracias a este modelo se pueden formar modelos para diferentes frases. Además, con una extensión del algoritmo de Viterbi, se pueden obtener las probabilidades para cada frase obtenida. El proceso de reconocimiento termina con la elección del modelo de la frase que mayor probabilidad posea y devuelve como resultado el texto con el que se formó la frase (Milone, 2004). Una representación gráfica para el modelo compuesto de la frase: 'está en el sótano del comedor' puede observarse en la figura 3.16. Se pueden observar los tres niveles de la composición: los estados del modelo acústico en la parte de arriba, el diccionario fonético al medio y el modelo de lenguaje en la parte inferior del diagrama. Se eliminaron los estados no emisores en los modelos acústicos para simplificar el esquema.

Figura 3.16

Modelo compuesto para la frase: está en el sótano del comedor



Nota. De Modelos ocultos de Markov para el reconocimiento automático del habla, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

3.3.3.5 Comparación

En la comparación se toma el modelo con mejor desempeño de la etapa de entrenamiento y se prueba con datos no usados en ninguna de las etapas anteriores (Clemente et al., 2001).

Esto da como resultado un porcentaje de error conocido como Word Error Rate (WER). Este concepto se detalla en el capítulo 6: Pruebas y Resultados.

3.4 Domótica

3.4.1 Concepto

La domótica se describe como una combinación de sistemas basados en sensores y actuadores con el propósito de mejorar las condiciones de vida y comodidad de las personas ajustando automáticamente las condiciones ambientales o asistiendo a los usuarios en sus tareas diarias (Pham-Huu et al., 2015).

3.4.2 Protocolos usados en domótica

En la domótica, los protocolos son los encargados de hacer posible la comunicación entre los distintos componentes de hardware utilizado. Estos establecen las reglas para

transmitir información entre dispositivos que se comunican de manera cableada o inalámbrica (Miori, Tarrini, Manca y Tolomei, 2006). Los protocolos más utilizados son:

3.4.2.1 X10

X10 es un protocolo de comunicación basado en el uso de la línea eléctrica. Fue el primer protocolo usado en domótica. Su desarrollo se remonta al año 1975 y, hoy en día, sigue siendo el protocolo más usado. Esto se debe a que cuenta con una serie de ventajas, entre las que destacan la sencillez y la accesibilidad al protocolo al poder utilizar la línea eléctrica del hogar y no necesitar cableado extra. Debido a esto, se han llegado a crear una considerable cantidad de aplicaciones de software y hardware (Yunaningsih, 2009).

3.4.2.2 EIB (European Installation Bus)

El EIB (European Installation Bus) es un protocolo desarrollado bajo el amparo de la Unión Europea con el objetivo de crear un estándar europeo y de disminuir la cantidad de productos domóticos importados provenientes de Japón y Norteamérica (Rodríguez, 2014).

El EIB es un sistema descentralizado, esto quiere decir, que si un elemento del sistema falla, el sistema puede seguir funcionando, aún si lo tiene que hacer de manera parcial. Dentro de este sistema existen varios medios para la interconexión de dispositivos (Herrera, 2005):

- Par trenzado
- Red eléctrica de baja tensión
- Radiofrecuencia
- Infrarrojo

La elección de un medio de transmisión sobre otro dependerá del tipo de edificación o vivienda y de las instalaciones con las que cuente. Actualmente, ha sido reemplazado por KNX.

3.4.2.3 BatiBUS

BatiBUS fue un protocolo de comunicación totalmente abierto desarrollado en Francia. Dentro de sus ventajas, destacaban su bajo coste, facilidad de instalación y capacidad de adaptación. El medio físico que utilizaba era el cable de par trenzado. Actualmente, es un protocolo obsoleto y ha sido reemplazado por KNX (Rodríguez, 2014).

3.4.2.4 EHS

El EHS (European Home System) fue un estándar más de creación europea que tuvo como objetivo la automatización de las viviendas europeas de manera masiva. Utilizaba como medio de comunicación la red eléctrica mediante el uso de la tecnología Power-line communication. A día de hoy, es un protocolo obsoleto y ha sido reemplazado por KNX (Rodríguez, 2014).

3.4.2.5 KONNEX- KNX

KONNEX- KNX surgió como una iniciativa de tres asociaciones europeas: EIBA (European Instalation Bus Association), BCI (Batibus Club Internacional), y EHSA (European Home Systems Association). KNX es el sucesor y la convergencia de los estándares previos representados por estas entidades: EIB, BatiBUS y EHS. Contempla 3 modos de trabajo dependiendo del nivel de competencia del instalador (Herrera, 2005):

- Modo S (sistema): Sigue el mismo formato que en EIB. Los dispositivos deben ser instalados y configurados por profesionales con el apoyo de un software específico para tal propósito.
- Modo E (fácil): Los dispositivos vienen configurados de fábrica. Solo se necesitan realizar pequeños ajustes para la instalación y configuración.
- Modo A (automático): Funciona como plug and play. El dispositivo solo se conecta y no se necesita ninguna configuración.

KNX puede utilizarse sobre distintos medios físicos, dentro de los que se encuentran:

- Par trenzado

- Ondas portadoras
- Ethernet
- Radiofrecuencia

3.4.2.6 Lonworks

Es un protocolo propietario desarrollado por la empresa Echelon en 1992. Tiene un alto costo y por ello no ha sido ampliamente adoptado. Destaca sobre todo su presencia en Estados Unidos (Herrera, 2005).

Se caracteriza por el hecho de que los dispositivos que quieran utilizar este protocolo deben contar con un microcontrolador especial propietario conocido como neuron chip. La principal ventaja de este protocolo es que implementa el modelo de referencia Open System Interconnection (OSI) (Rodríguez, 2014).

3.4.2.7 Zigbee

Zigbee es un conjunto de protocolos de comunicación inalámbrica basado en el estándar IEEE 802.14.5 de redes inalámbricas de área personal. Fue diseñado con el objetivo de lograr comunicaciones seguras con una baja tasa de envío de datos, maximizando la vida útil de los dispositivos (Pham-Huu et al., 2015).

3.4.3 Tecnologías de interconexión

Las tecnologías de interconexión utilizadas para la domótica se dividen en dos grandes grupos:

3.4.3.1 Cableadas

Las tecnologías cableadas son utilizadas para realizar las conexiones de forma alámbrica. Las tecnologías usadas son:

- Ethernet: Es un estándar de redes de área local creado por la unión de varios dispositivos a través de un cable. Se ha ido actualizando a través del tiempo

añadiendo mejoras continuas para adaptarse a las necesidades actuales. El estándar más avanzado, 10 Gigabit Ethernet, soporta velocidades de hasta 10 Gbit/s.

- USB: Es un estándar que define los cables, conectores y protocolos usados para la comunicación y alimentación eléctrica de dispositivos a través de un bus serial. Este protocolo se ha ido actualizando hasta el actual USB 3.2 con una velocidad de 20 Gbit/s.
- FireWire: Es un estándar para un bus serial similar al USB. Actualmente, su uso es residual y en su lugar es usado el estándar USB. Ofrece velocidades de hasta 3.2 Gbit/s en el estándar Firewire s3200.
- Thunderbolt: Es una interfaz de alta velocidad que se utiliza sobre una conexión, brindando mejoras como velocidades más altas o la posibilidad de conectar nuevos periféricos. Es una tecnología propietaria. La última versión del protocolo, Thunderbolt 3, ofrece velocidades de hasta 40 Gbit/s.

3.4.3.2 Inalámbricas

- Wi-Fi: Nombre que engloba varios protocolos que permiten la interconexión inalámbrica entre dispositivos. Estos dispositivos pueden conectarse entre sí o a internet mediante un punto de acceso.
- Bluetooth: Es un estándar industrial utilizado en redes inalámbricas de área personal (WPAN). Tiene como objetivo facilitar la comunicación entre dispositivos móviles.

3.4.4 Productos

Son los dispositivos que forman parte de un entorno domótico. Se listan algunos a continuación:

- Bombillas inteligentes
- Cámaras
- Timbres

- Sensores
- Alarmas
- Termostatos
- Altavoces
- Enchufes

3.4.5 Plataformas

Permiten el uso de los dispositivos mencionados en el punto 3.4.4 mediante la interacción con una pantalla o con el uso de la voz.

- Apple Homekit: Es la plataforma desarrollada por Apple para el manejo de un sistema domótico instalado en una vivienda. Permite controlar distintos dispositivos inteligentes desde un teléfono inteligente, Tablet, mediante control de voz, etc.
- Google Home: Es la plataforma de Google para crear un hogar inteligente. Incluye una amplia gama de productos los cuales son controlados mediante software desarrollado por Google a través de una pantalla o mediante el uso de la voz.
- Alexa: Es la plataforma de Amazon para el control de un entorno domótico. Destaca por tener una gran compatibilidad con diversos productos y por su potente asistente de voz.
- Neocontrol: Es una empresa brasileña dedicada a la domótica con presencia en el Perú. Ofrece diversos productos que pueden ser controlados mediante su aplicación Minibox Wi-fi.

Además, es posible crear una plataforma propia mediante el uso conjunto de PCs, tablets, Raspberry Pi o Arduino (Dennis, 2013; Sanclemente, 2016).

3.4.6 Domótica para personas con discapacidad

Un sistema domótico para personas con discapacidad presenta como principal diferencia, respecto a un sistema domótico convencional, un entorno especialmente adaptado a las dificultades que tengan los usuarios (López et al., 2016).

Estos sistemas especialmente adaptados brindan soluciones para facilitar la interacción humano-interfaz. En general, la incorporación de comandos de voz para controlar la interfaz facilita el uso del sistema para personas con dificultades de movilidad, pero con plena capacidad de habla (Prieto y Jara, 2013).

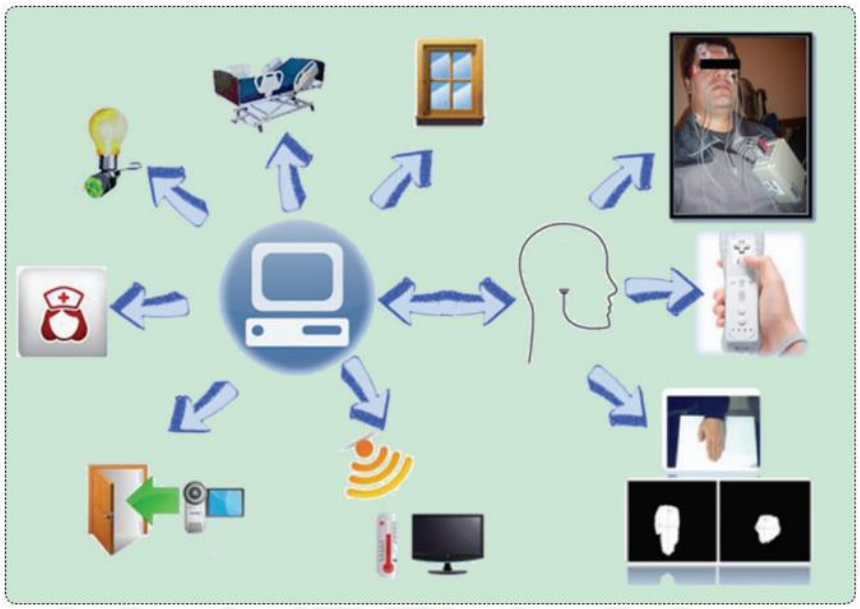
En el caso particular de que el usuario presente dificultades para mover el brazo (lo que dificulta el uso de un interfaz convencional, como una que utilice la combinación de teclado y ratón), también se puede optar por una interfaz en la que la navegación se realice mediante la detección de la posición de la mano, reemplazando el clic por la acción de cerrar la mano. Otra posible solución a este inconveniente es el uso de consolas con sensores de movimiento como la Nintendo Wii o la HP Swing. El uso de estas soluciones garantiza robustez, ergonomía, bajo costo y una instalación sencilla (López et al, 2016).

Para las personas que presenten discapacidad severa (como los cuadripléjicos) es posible controlar el puntero de un mouse mediante el movimiento ocular y la contracción voluntaria de cualquier músculo facial (López et al, 2016).

Los sistemas mencionados anteriormente pueden ser aplicados para controlar el aire acondicionado, ventanas, cama ortopédica, luces, botón de llamada a enfermera, intercomunicador, aire acondicionado, etc. En la figura 3.17 se puede observar como el usuario puede interactuar con estos objetos mediante el control por movimiento ocular, Nintendo Wii o por detección de mano.

Figura 3.17

Interacción Humano-Computadora



Nota. De *From hospital to home care: Creating a domotic environment for elderly and disabled people*, por López et al., IEEE pulse, 7(3), 38-41.

3.5 Arduino

3.5.1 Características

Arduino es una plataforma de creación de electrónica de código abierto que abarca hardware y software creada por la empresa del mismo nombre. Arduino cuenta con distintas placas basadas en un microcontrolador ATMEL en el cual se pueden guardar instrucciones mediante el lenguaje de programación Arduino (Torrente, 2013).

Arduino cuenta con una serie de ventajas que hacen que destaque sobre otras soluciones: la curva de aprendizaje es rápida, cuenta con una gran comunidad y accesorios y tiene un costo reducido. Estas ventajas se mantienen a través de los distintos modelos de placas que existen. Dentro de estos modelos destacan:

- Arduino Uno: Arduino Uno es la primera placa basada en el estándar USB. Esta se ha ido actualizando con distintas revisiones hasta la versión actual en

la que se cuenta con un procesador ATmega328P, 14 pines digitales y 6 pines analógicos.

- Arduino Nano: Se trata de una versión reducida del Arduino Uno en la que se respetan las salidas, pero el procesador cambia a un ATmega328. Este procesador es más pequeño, lo que permite que el Arduino Nano sea considerablemente más pequeño.
- Arduino Mega: Es la versión más potente y completa de la familia Arduino. Cuenta con un procesador ATmega2560, 54 pines digitales y 16 pines analógicos. Sus dimensiones son las mayores para lograr estas características.

Tabla 3.2

Comparativa entre distintos modelos de Arduino

Modelo	Arduino Uno	Arduino Nano	Arduino Mega
Procesador	ATmega328P	ATmega328	ATmega2560
Dimensiones	68.6 mm x 53.3 mm	43.18 mm x 18.54 mm	101.6 mm x 53.3 mm
Salidas digitales I/O	14	14	54

Nota. Adaptado de *Casa domótica con Arduino*, por Sanclemente, C., 2016, Universidad Politécnica de Valencia, España.

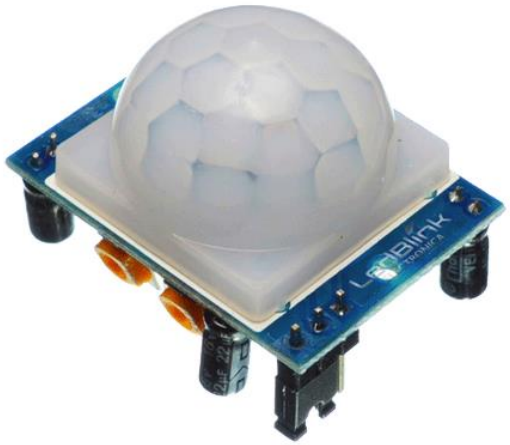
3.5.2 Sensores Arduino para domótica

Existe una gran variedad de sensores que pueden ser conectados a las placas Arduino. Debido a su popularidad, estos son fáciles de conseguir y existe una vasta comunidad que comparte conocimiento acerca de cómo utilizarlos y sobre nuevos posibles usos de los sensores (Sanclemente, 2016). Entre los sensores más destacados se encuentran:

- Sensor de presencia: Este sensor detecta la presencia mediante luz infrarroja. Este sensor puede ser utilizado para encender luces al ingresar un ambiente, abrir puertas al estar cerca, entre otros.

Figura 3.18

Sensor de presencia HC-SR501



Nota. De Hi-Fi S.A.C, 2020, https://www.hifisac.com/shop

- Sensor de temperatura: El sensor de temperatura mide la temperatura de un objeto o ambiente. El sensor de la figura 3.19 es capaz de medir la temperatura de un objeto a distancia en función de la emisión de infrarrojos que este objeto produce.

Figura 3.19

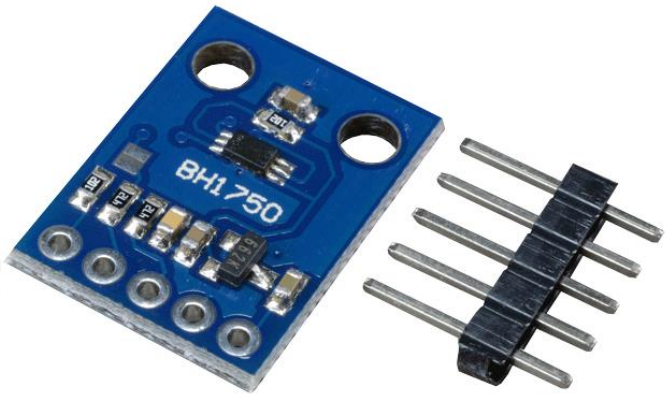
Sensor de temperatura MLX90614

Nota. De Hi-Fi S.A.C, 2020, https://www.hifisac.com/shop

- Sensor de luminosidad: Este sensor mide la luminosidad en el ambiente. El valor obtenido se muestra en Lux (lumen /m²).

  Figura 3.20

  Sensor de luminosidad BH1750



  Nota. De Hi-Fi S.A.C, 2020, https://www.hifisac.com/shop

- Sensor de flujo de agua: Este sensor sirve para medir el caudal de un fluido. Esta medida se representa como la cantidad de líquido que circula a través de una tubería por unidad de tiempo.

  Figura 3.21

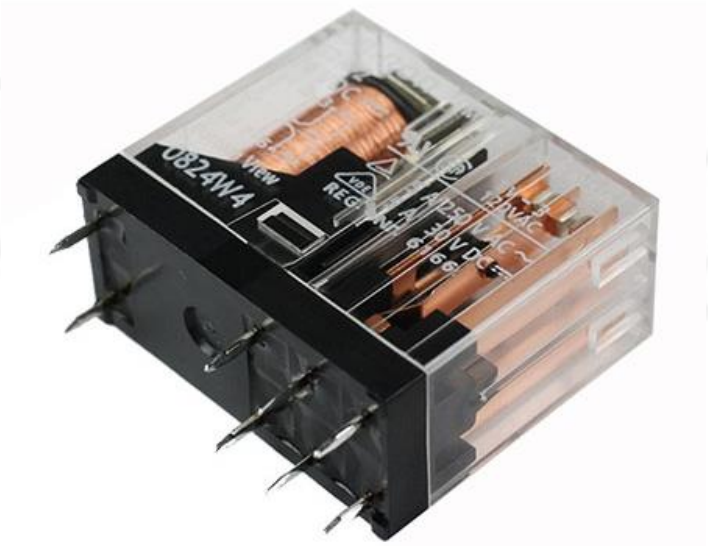
  Sensor de flujo de agua YF-S201

  Nota. De Hi-Fi S.A.C, 2020, https://www.hifisac.com/shop

- Relé de potencia: Este dispositivo actúa como un interruptor que puede controlarse mediante el Arduino. Permite el paso de corriente eléctrica cuando está cerrado y la interrumpe cuando está abierto. Gracias a esto, se puede controlar el encendido de diversos dispositivos.

Figura 3.22

Relé OMRON OCB



Nota. De Hi-Fi S.A.C, 2020, https://www.hifisac.com/shop

3.5.3 Ejemplos de control domótico usando Arduino

La plataforma Arduino tiene diversos usos dentro de un entorno domótico, pudiendo controlar y automatizar distintos aspectos de la vivienda. Algunos posibles usos son:

- Control de iluminación: Gracias a la plataforma Arduino, es posible indicar el encendido y apagado de luces mediante sensores y actuadores. La luz puede ser encendida al ingresar a un ambiente, apagada si se detecta que no hay nadie, encendida mediante comandos de voz, entre otros.
- Automatización de persianas: Las persianas pueden abrirse o cerrarse según las condiciones medioambientales.

- Automatización de electrodomésticos: La plataforma Arduino permite encender o apagar los electrodomésticos en base al suministro de energía. Esto se puede programar por horarios o realizarse de forma manual mediante comandos de voz.
- Control de temperatura: Se regula la temperatura de la vivienda de forma automática o mediante el uso de comandos.
- Seguridad: Arduino permite controlar distintos aspectos de la seguridad como: presencia de personas, detección de incendios mediante el uso de detectores de humo, control de cámaras, etc.

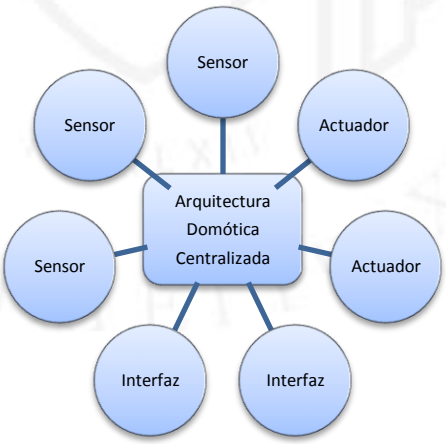
3.5.4 Arquitecturas de control domótico

3.5.4.1 Centralizada

Todos los elementos están conectados y se controlan desde un único sistema de control. Este puede ser una PC, un Raspberry Pi o un microcontrolador embebido como un Arduino (Sanclemente, 2016).

Figura 3.23

Arquitectura Domótica Centralizada



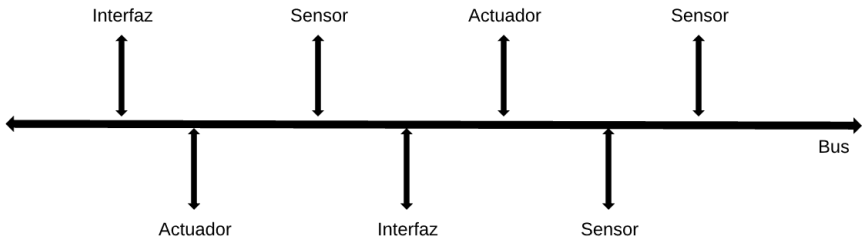
Nota. Adaptado de *Casa domótica con Arduino*, por Sanclemente, C., 2016, Universidad Politécnica de Valencia, España.

3.5.4.2 Distribuida

La arquitectura distribuida tiene como característica principal que cada uno de los dispositivos tiene un procesador propio, es decir son independientes. Gracias a este procesador, el dispositivo puede llevar a cabo ciertas tareas que vengan programadas por el fabricante. Estas son llevadas a cabo mediante la información que recibe por un bus de datos que interconecta a todos los dispositivos. Esta arquitectura es muy utilizada en sistemas domóticos inalámbricos (Wimsatt, 2004).

Figura 3.24

Arquitectura Domótica Distribuida



Nota. Adaptado de *Casa domótica con Arduino*, por Sanclemente, C., 2016, Universidad Politécnica de Valencia, España.

CAPÍTULO IV: PROPUESTA DE SOLUCIÓN

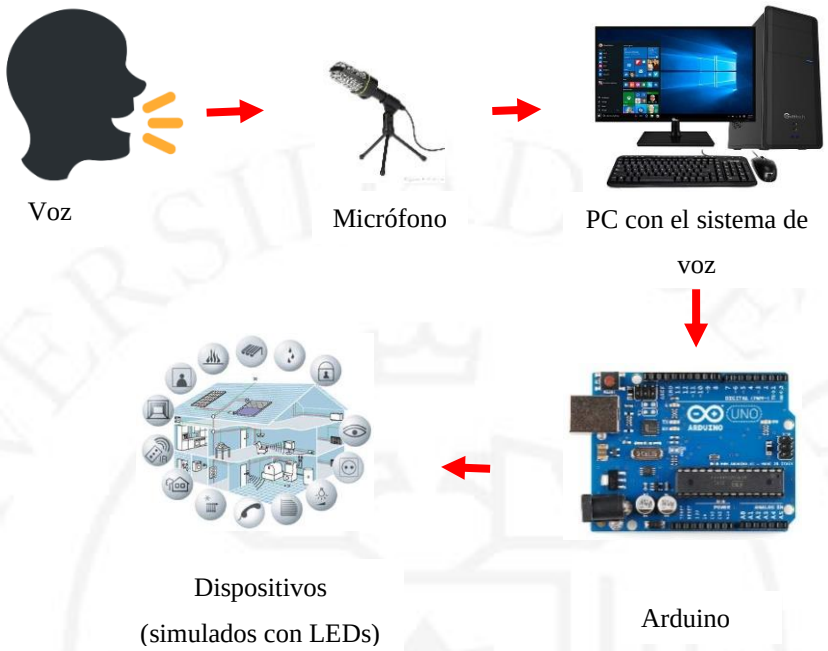
4.1 Método de investigación

El presente trabajo está dividido en 3 grandes fases:

1. La primera fase está dedicada al estudio de los modelos ocultos de Markov y a cómo estos son aplicados al reconocimiento de voz.
2. La segunda fase consiste en el desarrollo del sistema de reconocimiento de comandos de voz.
3. La tercera fase consiste en la implementación del sistema de control por voz en un sistema domótico mínimo para simular acciones automatizadas en el hogar. Para esto se hizo uso de la plataforma Arduino como interfaz entre la PC y los dispositivos actuadores de control domótico. En la figura 4.1 se puede ver el proceso completo. Este comienza cuando el usuario dice un comando. Este es recibido por el micrófono que envía la señal al PC donde se encuentra el sistema que procesa la señal y la transforma en texto. Este texto se envía al Arduino que se encuentra programado para realizar una acción distinta dependiendo del input que recibe. Finalmente, el Arduino envía la indicación al dispositivo de control domótico que realiza la acción y termina el proceso.

Figura 4.1

Proceso de reconocimiento de comandos de voz del sistema aplicado sobre un sistema domótico mínimo



4.2 Alcance

El alcance del trabajo respecto al sistema de reconocimiento del habla es el siguiente:

- El vocabulario contemplado en el reconocedor estará limitado a un conjunto de palabras básicas de un diccionario, pues su uso solo tiene como objetivo la activación/desactivación de ciertas funcionalidades del hogar.
- Las diferentes funcionalidades del sistema domótico mínimo son representadas mediante LEDs de diferentes colores, que cambian su estado ON/OFF según el comando de voz reconocido.

4.3 Supuestos

- Los usuarios objetivos del presente trabajo son las personas que presenten alguna discapacidad física, pero que conserven pleno uso del habla.

- Los usuarios son hispanohablantes.
- El suministro de energía de la vivienda es desde la red principal y de forma ininterrumpida.
- No existe ruido de fondo de alta intensidad en los ambientes del hogar.
- El usuario del sistema solo será una persona, con la cual el sistema será entrenado.

4.4 Riesgos

Existe el riesgo de que el reconocedor no reconozca la palabra que se haya indicado y se realice otra acción. Este riesgo se cuantifica mediante el porcentaje de error (WER). Debido a que este riesgo es conocido, se han tomado medidas para reducir su impacto. La principal medida tomada es la inclusión de comandos para cancelar las acciones realizadas previamente, con lo cual se logra corregir la acción incorrecta.

4.5 Entregables

- Cuadro de pruebas en escenarios distintos
- Simulación de funcionamiento del reconocedor en un entorno domótico mínimo.

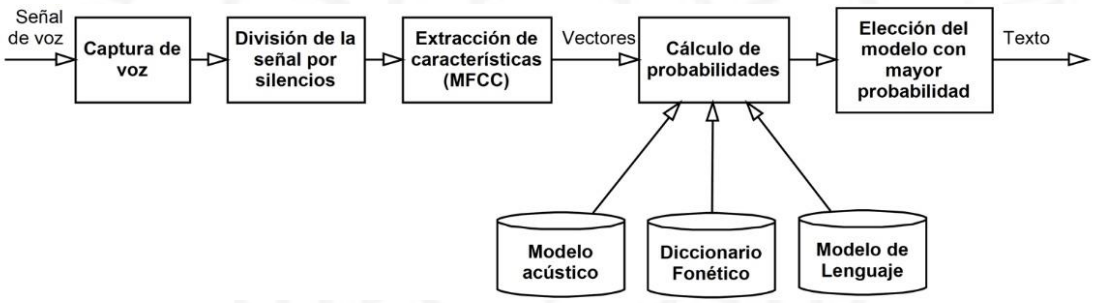
CAPÍTULO V: DESARROLLO DE LA SOLUCIÓN PROPUESTA

5.1 Arquitectura de la solución

La solución propuesta se divide en 2 grandes partes: El reconocedor de voz y el sistema domótico utilizando Arduino. El reconocedor propuesto presenta la estructura de la figura 5.1:

Figura 5.1

Diagrama del reconocedor de voz desarrollado



Nota. Adaptado de *Desarrollo de un sistema de reconocimiento de habla natural independiente del locutor*, por Antón, J., 2015, [tesis de licenciatura, Universidad Autónoma de Madrid]. Departamento de Tecnología Electrónica y de las Comunicaciones, España. https://repositorio.uam.es/handle/10486/667850

Para poder entender la estructura, es necesario conocer previamente cómo actúan en la solución el Modelo Acústico, el Diccionario Fonético y el Modelo de Lenguaje y cómo interactúan entre sí.

5.1.1 Modelo Acústico

El modelo acústico es la representación estadística de los distintos sonidos que forman las palabras. En otras palabras, es un conjunto de modelos ocultos de Markov que representan los fonemas. Los fonemas no son unidades aisladas, sino que dependen del

contexto (fonema previo y fonema posterior). Los fonemas que consideran el contexto son llamados trifonemas (Carnegie Mellon University, 2008). Por ejemplo, la palabra roca es representada por los siguientes fonemas:

<sil> R O K A <sil>

Donde <sil> representa que la palabra está precedida y continuada por silencio.

En trifonemas, los fonemas se representarían de la siguiente manera:

R(sil, O); O(R, K); K(O, A); A(K, sil)

Los cuatro fonemas se siguen mostrando; pero, además, se añaden los fonemas previos y siguientes dentro del paréntesis por cada fonema. Cada uno de estos trifonemas es representado por un HMM, donde se tienen 3 estados: fonema de entrada, fonema principal y fonema de salida. Si la combinación de 2 estados es similar o igual, es posible reducir la cantidad de estados totales para el programa, debido a que la información para ambos es la misma (Singh, 2003). Por ejemplo, para O(R, K) y O(R,A), se tiene el mismo inicio (O precedido de R). Por lo tanto, para el Sphinx, el estado perteneciente a R es considerado como un solo estado, debido a que comparten las mismas probabilidades de transición y emisión.

Para la creación del modelo acústico es necesario un proceso de entrenamiento de una base de datos de voces, llamada Corpus. Este entrenamiento se apoya en una serie de algoritmos que serán detallados más adelante. El modelo acústico del presente trabajo se creó utilizando la data del Corpus CIEMPIESS obtenido de la Universidad Nacional Autónoma de México (UNAM). El detalle del corpus utilizado puede verse en el siguiente link: http://www.ciempiess.org/CIEMPIESS_Statistics.html

El entrenamiento requiere de una serie de archivos detallados a continuación:

- Diccionario fonético.
- Lista de fonemas.
- Modelo de lenguaje
- Diccionario con sonidos no procedentes del habla.
- Lista de los archivos de voz usados para el entrenamiento.
- Transcripciones de los archivos de audio a texto.

- Archivos con las voces de uno o más hablantes.

Estos archivos se encuentran dentro del Corpus CIEMPIESS, por lo que una vez ordenados correctamente y, habiendo validado el formato solicitado, se procede a realizar el entrenamiento.

El entrenamiento es realizado mediante la herramienta Sphinxtrain. Esta se apoya en dos algoritmos para lograr su cometido: El algoritmo Forward-Backward y el algoritmo Baum-Welch. Este último es el que se encarga del entrenamiento en sí, es decir, se encarga de encontrar el HMM que maximice las posibilidades de las observaciones recibidas. Baum-Welch se apoya en el algoritmo Forward-Backward para su funcionamiento (Oropeza y Suárez, 2006).

5.1.2 Diccionario Fonético

El diccionario fonético, como ya se mencionó, es la lista de las palabras con los respectivos fonemas que la componen en el lenguaje deseado (Milone, 2004). El diccionario fonético tiene la estructura de la figura 5.2, donde las palabras se muestran a la izquierda y los fonemas que las representan a la derecha.

Figura 5.2

Extracto del diccionario fonético utilizado.

```
salome s a l o m e
salomon s a l o m o n
salon s a l o n
salones s a l o n e s
salou s a l o u
salpapicos s a l p a p i k o s
salpicada s a l p i k a d a
salpicado s a l p i k a d o
salpicados s a l p i k a d o s
salpicaduras s a l p i k a d u r( a s
salpicar s a l p i k a r(
```

Nota. Adaptado de *A New Open-Sourced Mexican Spanish Radio Corpus,* por Hernández-Mena, C. y Herrera-Camacho, J., 2014, In LREC. European Language Resources Association.

Para el presente trabajo, se ha utilizado el Diccionario Fonético incluido en el Corpus CIEMPIESS y se ha adaptado a las necesidades del trabajo, reduciendo la cantidad de palabras que el reconocedor identificará.

Este archivo actúa como intermediario entre el Modelo de Lenguaje y el Modelo Acústico y su función es esencial para el correcto funcionamiento del sistema. Este enlace se da al momento de llamar iniciar el descifrador, software encargado de enlazar las partes que forman el lenguaje.

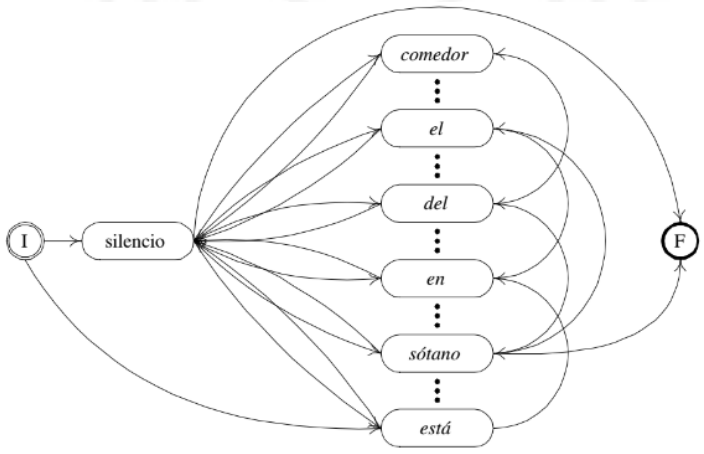
5.1.3 Modelo de Lenguaje

El Corpus obtenido de la UNAM también contiene un modelo de lenguaje; pero, a diferencia del Diccionario Fonético, este archivo no solamente será tomado y usado en el sistema.

El modelo de lenguaje contiene las probabilidades de transición entre palabras para formar frases. Dicho de otro modo, es un Modelo Oculto de Markov de palabras. Gracias al modelo de lenguaje, el reconocedor aumenta su velocidad y precisión (Milone, 2004).

Figura 5.3

Modelo de lenguaje



Nota. Los estados de inicio y finalización se indican con las letras I y F, respectivamente. De *Modelos ocultos de Markov para el reconocimiento automático del habla*, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

Existen distintos tipos de modelos de lenguaje. Para el presente trabajo se utiliza un modelo de lenguaje trigrama, es decir, un modelo de lenguaje en el que cada HMM tiene 3 estados, cada uno representando una palabra. De esta forma, se puede obtener la palabra más probable en base con la anterior. Además, también se tienen modelos de dos estados (bigrama) y un estado (unigrama). Si no se encuentra un trigrama adecuado, se utiliza un bigrama y, en caso no se encuentre el bigrama, se pasa a utilizar el unigrama (Gales y Young, 2008). Una representación del modelo de lenguaje trigrama puede observarse en la figura 5.4:

Figura 5.4

Extracto del modelo de lenguaje utilizado.

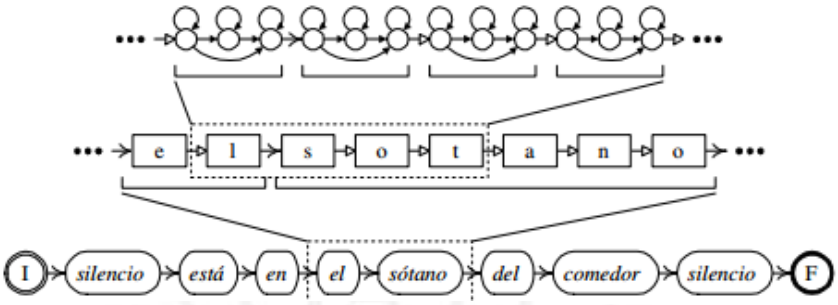
```
-0.9024 dEE platafOOrmas pAAra
-0.6014 dEE platicAAr cOOn
-0.2752 dEE pluralIIsmo </s>
-0.6014 dEE pluralidAAd </s>
-0.6014 dEE poblAAdos </s>
-2.3174 dEE poblaciOOn <UNK>
-0.1845 dEE poblaciOOn </s>
-1.6902 dEE poblaciOOn EEn
-2.3174 dEE poblaciOOn contribUUyen
-1.6902 dEE poblaciOOn dEE
-2.3174 dEE poblaciOOn hispAAna
-2.3174 dEE poblaciOOn mundiAAl
-2.3174 dEE poblaciOOn pAAra
-2.3174 dEE poblaciOOn tendrAAn
```

Nota. Adaptado de *A New Open-Sourced Mexican Spanish Radio Corpus,* por Hernández-Mena, C. y Herrera-Camacho, J., 2014, In LREC. European Language Resources Association.

Para calcular la salida más probable, el Modelo de Lenguaje utiliza el algoritmo de Viterbi siguiendo el funcionamiento que se explicó en el marco teórico.

Figura 5.5

Modelo compuesto para la frase: está en el sótano del comedor.



Nota. De Modelos ocultos de Markov para el reconocimiento automático del habla, por Milone, D., 2004, http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf

En la figura 5.5 se pueden observar los tres niveles de la composición: los estados del modelo acústico, el diccionario fonético y el modelo de lenguaje. Cada uno de estos niveles está compuesto por varios HMM. En los modelos acústicos se han eliminado los estados no emisores para simplificar el esquema.

El funcionamiento del reconocedor es el siguiente: Los sonidos son discretizados y transformados a vectores. Por cada vector, el descifrador (decoder) busca el HMM del modelo acústico que representa el vector que representa un sonido. Mediante este proceso se pueden obtener los fonemas de lo que ha querido decir el usuario, pues los HMMs del modelo acústico representan fonemas del idioma que se esté utilizando, en este caso, español latinoamericano. Este proceso continúa y el descifrador sigue asociando sonidos a HMM de los fonemas en el modelo acústico hasta que se llegue a una pausa en el habla (Gales y Young, 2008).

Luego de la pausa, el descifrador obtiene todos los fonemas escuchados y separa los conjuntos de fonemas en base a los silencios. En el caso de la figura 5.5, se obtienen dos conjuntos de fonemas: uno para la palabra ‘el’ y otro para la palabra ‘sótano’.

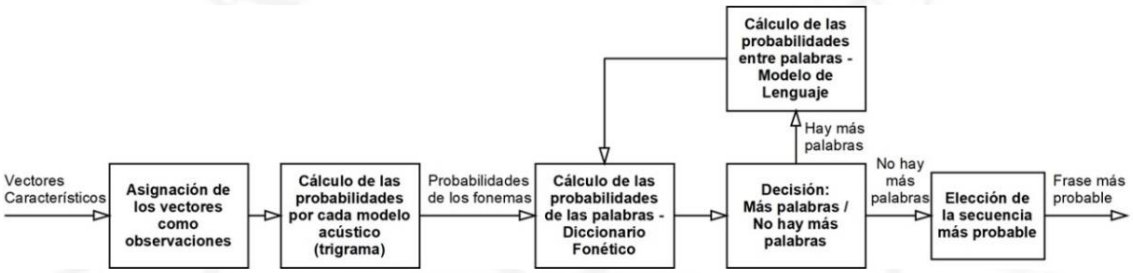
Para la palabra ‘el’, se obtienen los fonemas correspondientes a las letras ‘e’ y ‘l’. El diccionario fonético, que guarda la relación entre fonemas y palabras, busca entre todas las posibles palabras teniendo en cuenta las probabilidades previas del modelo acústico y decide que la palabra más probable es ‘el’.

Finalmente, en el caso del habla continua como en este caso, el descifrador busca dentro del modelo de lenguaje las combinaciones de frases y sus posibilidades, teniendo en cuenta las probabilidades previas calculadas para el modelo acústico y el diccionario fonético, y determina cuál es la secuencia de palabras más probable. En este caso devuelve la frase ‘el sótano’.

Para el cálculo de las probabilidades en todos los modelos se utiliza el algoritmo de Viterbi explicado en el marco teórico.

Figura 5.6

Viterbi en el reconocimiento de voz.



Como se muestra en la figura 5.6, el algoritmo de Viterbi funciona de la siguiente manera en el reconocimiento de voz:

- Asignación de los vectores como observaciones: Se recibe como input los vectores característicos. Estos son tomados como observaciones para los HMM de los fonemas.
- Cálculo de las probabilidades por cada modelo acústico: Con las observaciones del paso anterior, se calculan las probabilidades para cada HMM del modelo acústico. Con esto se tienen las probabilidades de cada fonema.
- Cálculo de las probabilidades – Diccionario Fonético: Teniendo las probabilidades de los fonemas, en este paso se calculan las probabilidades de las palabras en base a su estructura fonética.

- Cálculo de las probabilidades – Modelo de Lenguaje: Si se reconoce más de una palabra, se calculan las probabilidades entre palabras (las secuencias de palabras) utilizando el modelo de lenguaje.
- Elección de los modelos con mayor probabilidad: Con las probabilidades de los tres niveles (fonemas, palabras y secuencias de palabras) se elige la secuencia de palabras más probable, siendo el texto de la secuencia más probable la salida del algoritmo.

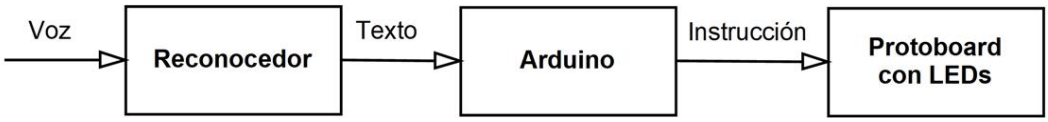
Con el conocimiento de estos conceptos previos, se procede a detallar los pasos mencionados en la figura 5.1:

- Captura de voz: Se captura la señal de la voz mediante un micrófono.
- División de la señal por silencios: Se divide la señal de voz de acuerdo con los silencios para diferenciar las palabras.
- Extracción de características: La señal continua de voz se discretiza obteniendo los coeficientes cepstrales en escala Mel (MFCC). Con estos coeficientes se forman vectores que son los que se utilizan para buscar las palabras.
- Cálculo de las probabilidades: Con los vectores obtenidos en el paso anterior y los HMMs del modelo acústico, diccionario fonético y modelo de lenguaje obteniendo las probabilidades de cada vector en cada modelo con el uso del algoritmo de Viterbi.
- Elección del modelo con mayor probabilidad: Luego de obtener todas las probabilidades, se eligen los modelos con mayor probabilidad. Hay modelos para los fonemas, las palabras y las secuencias de palabras. La combinación de todos los modelos más probables da como resultado la palabra o frase dicha.

La arquitectura del reconocedor se une con la domótica de la forma que se muestra en la figura 5.7:

Figura 5.7

Arquitectura de la solución.

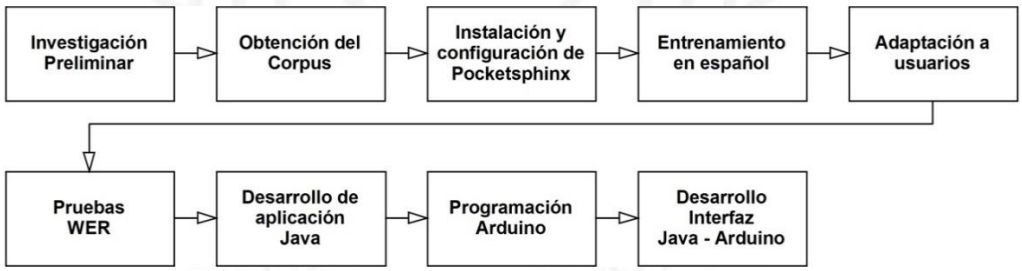


- Reconocedor: Este bloque representa todo lo mostrado en la figura 5.1.
- Arduino: El Arduino recibe un texto enviado por el reconocedor, lo interpreta y envía una instrucción en base al texto recibido. Este reconocimiento es posible debido a que se programó el Arduino para que maneje distintos casos en base al texto recibido.
- Protoboard con LEDs: El Arduino envía la instrucción de encender o apagar la luz o luces LED conectadas al Protoboard.

5.2 Proceso de desarrollo de la solución

Figura 5.8

Diagrama del desarrollo.



Tal como se muestra en la Figura 5.8, el desarrollo consiste en los siguientes pasos:

- Investigación Preliminar: Esta etapa consiste en la investigación sobre la solución y sobre los requerimientos y pasos que se necesitaron para desarrollarla.

- Obtención del Corpus: Se obtuvieron los archivos necesarios para realizar el entrenamiento del sistema.
- Instalación de PocketSphinx: Se instaló y configuró PocketSphinx en el sistema operativo Ubuntu. En este paso se dejó todo configurado para realizar el entrenamiento y posteriormente la adaptación. Se utilizó PocketSphinx sobre Sphinx4 (alternativa exclusiva para computadoras personales y servidores) por su facilidad de uso para esta tarea. Debido a su mayor complejidad y versatilidad, Sphinx4 se utilizó al momento de desarrollar el aplicativo en Java.
- Entrenamiento al español: En este paso se usó parte de la información del Corpus para entrenar un sistema de reconocimiento de voz en español. Al finalizar, se obtiene el modelo acústico para el español. Este modelo, en conjunto con el modelo de lenguaje y el diccionario fonético forman el sistema de reconocimiento de voz. A partir de este momento, ya se pueden reconocer palabras.
- Adaptación a usuarios: Para mejorar la precisión, se necesitaron realizar dos ajustes: reducir las palabras a reconocer en el diccionario fonético y adaptar el modelo acústico al usuario que vaya a utilizar el sistema.

  El detalle de estos pasos se encuentra en el paso 5.3.
- Pruebas WER: Se realizaron pruebas para medir el porcentaje de error en distintos escenarios y analizar la evolución a lo largo de los cambios. Esto se realiza para 20 personas distintas.
- Desarrollo de aplicación Java: Se creó una aplicación en la que se utilizan los archivos generados previamente para reconocer la voz del usuario. Esta aplicación utiliza los archivos generados en las etapas de entrenamiento y adaptación. Estas son interpretadas por Java gracias a las librerías provistas por Sphinx4. El detalle de estos pasos se encuentra en el paso 5.4.
- Programación Arduino: Se programó el Arduino para que realice distintas acciones basadas en la entrada que recibe. Se consideraron casos para el encendido y apagado de distintas luces.

- Desarrollo Interfaz Java – Arduino: Se crea la conexión entre la aplicación Java y el Arduino. Con esto, la voz reconocida en la aplicación Java se convierte en el input que recibe el Arduino para realizar una acción.

5.3 d Entrenamiento en español

El entrenamiento del reconocedor es una parte fundamental del desarrollo de la solución. El objetivo de este paso es conseguir un modelo acústico en español latino. Con esto, se puede tener un reconocedor de voz funcional en español latino, el cual luego será modificado para adaptarlo a las características requeridas por la solución.

Para obtener este modelo acústico, lo primero es conseguir los archivos necesarios del corpus CIEMPIESS:

- Archivos de audio: Se incluyen todos los archivos de audio de palabras o frases. Deben estar en formato WAV.
- Identificación de archivos de audio: Estos archivos contienen la ruta de los archivos de audio.
- Transcripciones: Archivos que contienen la transcripción a texto por cada uno de los archivos de audio.
- Diccionario Fonético: Asociación de las palabras con sus fonemas correspondientes.
- Listado de fonemas: Incluye el listado de todos los fonemas incluidos en los archivos de audio.
- Modelo de lenguaje: Contiene las secuencias de palabras posibles y sus probabilidades.
- Diccionario de relleno: Contiene fonemas no presentes en el modelo de lenguaje. En este caso, solo son fonemas que representan el silencio.

Luego de tener todos los archivos listos, hay que indicar dos parámetros antes de iniciar el proceso de entrenamiento: indicar las cantidades de ‘senones’ y las densidades. En el toolkit provisto por CMU, los senones equivalen a los trifonemas, mientras que las densidades equivalen a las probabilidades de salida.

CMU provee recomendaciones referenciales para definir estos 2 valores en base a la cantidad de palabras a reconocer y a la cantidad de horas que se tenga en el corpus.

Tabla 5.1

Valores recomendados para el entrenamiento.

Vocabulario	Horas de audio en en la base de datos	Senones	Densidades
20	5	200	8
100	20	2000	8
5000	30	4000	16
20000	80	4000	32
60000	200	6000	16
60000	2000	12000	64

Nota. Adaptado de *Learning to use the CMU SPHINX Automatic Speech Recognition system,* por Carnegie Mellon University, 2008, http://www.speech.cs.cmu.edu/sphinx/tutorial.html#app3

El corpus CIEMPIESS tiene 17.22 horas de data de entrenamiento y 53169 palabras a reconocer, por lo que decidió usar 4000 senones y 16 densidades. En otras palabras, el reconocedor tendrá 4000 trifonemas y cada uno de estos trifonemas tendrá asignadas 16 salidas posibles. Estas salidas pueden ser compartidas entre senones. Como se puede observar en la tabla 5.1, mientras más data se tenga, se necesita aumentar el número de senones y densidades. Por lo tanto, se necesita una mayor diversidad de trifonemas y salidas para representar una mayor cantidad de data.

Finalmente, se procede a realizar el entrenamiento. Este consiste en correr una serie de scripts que realizan los siguientes pasos:

1. El primer paso consiste en realizar la etapa de preprocesamiento. El resultado es la extracción de los vectores característicos en el proceso conocido como Vector Quantization. Esto se realiza para convertir la señal continua en discreta, debido a que el reconocedor no trabaja directamente con la señal de voz, sino que debe ser discretizada primero.
2. En este paso se generarán los 4000 senones (trifonemas) indicados en la configuración.

3. Se crean las conexiones entre estados de trifonemas, así como sus probabilidades de transición.
4. Finalmente, se asignan las probabilidades de emisión por cada estado. Como se indicó, cada estado tendrá 16 probabilidades distintas de salida. Este paso se hace de forma secuencial. Es decir, se asigna la primera probabilidad, luego la segunda y así hasta completar las 16 probabilidades.

5.4 Adaptación al usuario

La adaptación al usuario permite personalizar la herramienta para el usuario final. La solución tiene como finalidad implementar comandos para mejorar el desempeño en la vida cotidiana de las personas con discapacidad. Este tipo de uso presenta dos importantes características que no se toman en cuenta en las otras alternativas de propósito general:

- El listado de palabras utilizadas en los comandos es reducido.
- El reconocedor será usado únicamente por un usuario, la persona con discapacidad.

Estas dos características son tomadas en cuenta por la solución para ofrecer una solución con un mejor desempeño y personalizada para el usuario final. Esto se logra mediante la reducción del vocabulario y la adaptación del modelo acústico.

5.4.1 Reducción del vocabulario a reconocer

Debido a que el uso del reconocedor estará limitado a únicamente comandos para realizar acciones en el hogar, es posible limitar la cantidad de palabras consideradas por el sistema. Esto conlleva mejoras en la precisión del reconocedor, simplifica su uso y personaliza la solución al usuario. Al reducir el vocabulario, también se reduce el porcentaje de error (Qiao, Sherwani, Rosenfeld, 2010). Por ejemplo, se pueden evitar el uso de palabras con pronunciación similar, mejorando la precisión (Noyes y Frankish, 1992). Además, se pueden adecuar la solución a las palabras de preferencia del usuario.

El entrenamiento realizado en el paso anterior da como resultado un sistema de reconocimiento de voz capaz de reconocer 53169 palabras distintas. Esta cantidad de

palabras son las que se encuentran dentro del corpus CIEMPIESS y son las que aparecen en el diccionario fonético.

CMUSphinx contiene las probabilidades de las palabras y sus secuencias dentro del modelo de lenguaje y la lista de palabras a reconocer en el diccionario fonético. Las palabras que se encuentren en ambos archivos son las que el sistema reconoce. Por lo tanto, para la reducción del vocabulario a reconocer se procede a modificar el diccionario para que refleje solo las palabras que se quieren reconocer. Esta modificación se realiza de acuerdo con los objetos y/o actividades que se quieran manejar mediante la voz.

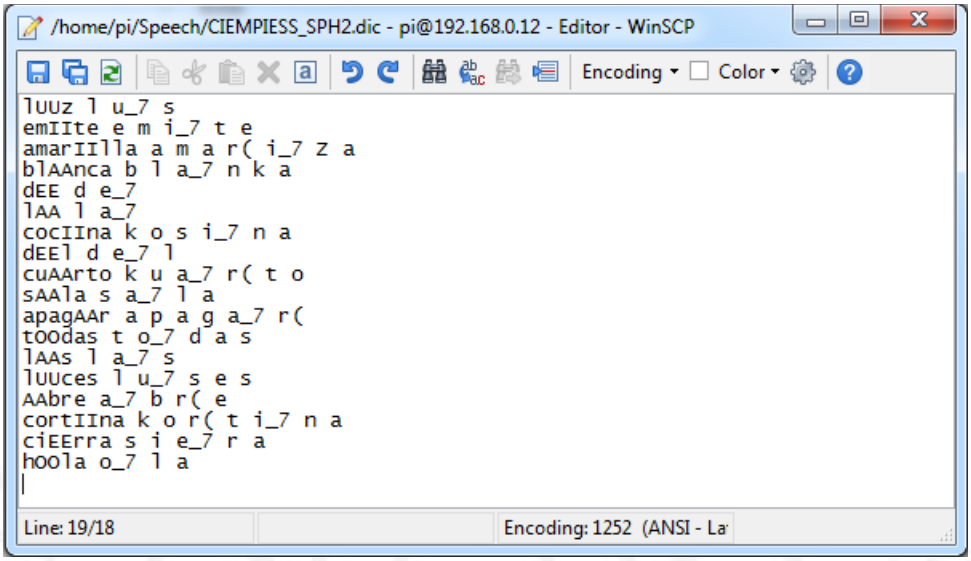
En el presente trabajo y para realizar las pruebas se redujo el diccionario dejando solo las siguientes palabras:

- luz
- emite
- amarilla
- blanca
- de
- la
- cocina
- sala
- del
- cuarto
- apagar
- todas
- las
- luces
- abre
- cortina
- cierra

- hola

Figura 5.9

Diccionario reducido para pruebas.



5.4.2 Adaptación del modelo acústico

La adaptación del modelo acústico es una modificación mediante la cual se puede mejorar el rendimiento del reconocedor de voz sin la necesidad de realizar un reentrenamiento (Wang, Schult y Waibel, 2003).

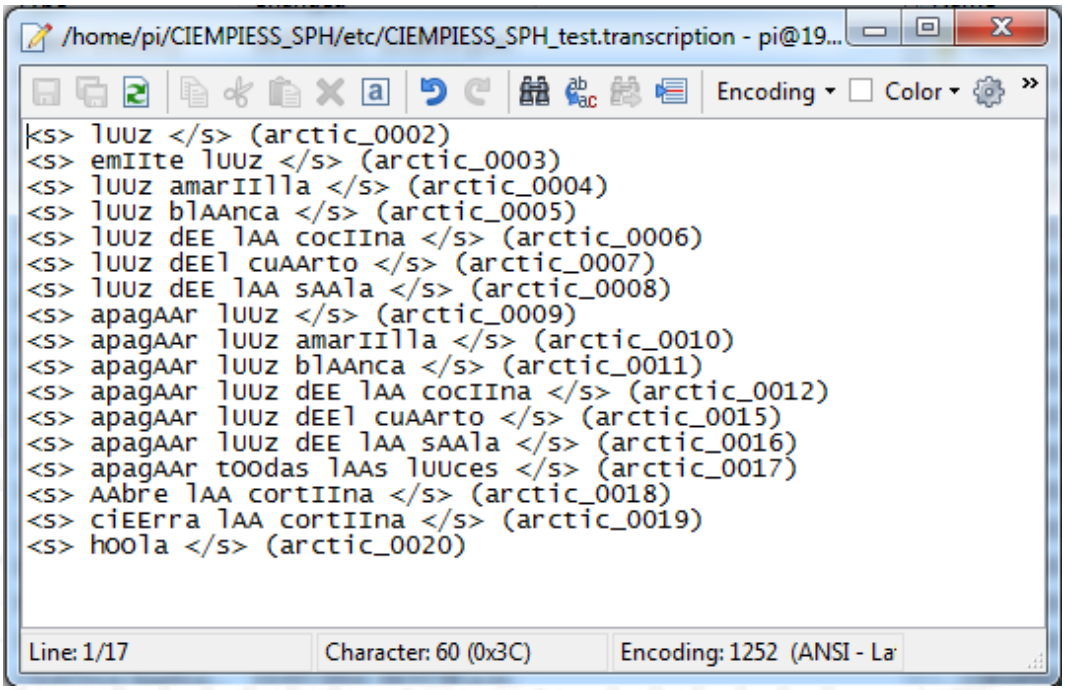
Este proceso consigue ajustar el modelo acústico para que represente mejor la data (corpus) de adaptación. Es posible adaptar el modelo a un hablante en particular, al entorno en que se va a hablar o a un acento distinto del mismo idioma. Es inclusive posible realizar una adaptación entre distintos lenguajes, siempre que se manejen los mismos fonemas (Carnegie Mellon University, 2016).

En este caso, el modelo acústico se adaptará a un hablante en particular, siempre tomando en cuenta el ruido de fondo que pueda existir. Gracias a esto, se logra reducir el porcentaje de error para el usuario que utilizará el reconocedor de voz, en este caso, la persona con discapacidad. Para esta adaptación, es necesario grabar las voces del hablante diciendo las palabras/frases que utilizará en el sistema. Estas deben ser grabadas lo más limpias posible, a una frecuencia de 16 KHz y en mono con un solo canal (Carnegie Mellon University, 2016). Se grabaron 17 distintos comandos que se desean reconocer.

Una vez se tengan las voces grabadas, es necesario crear dos archivos: uno que contiene la transcripción de lo que se ha dicho y otro enumerando las transcripciones.

Figura 5.10

Transcripción con los comandos a reconocer.



Las grabaciones se utilizan para la creación de los Coeficientes Cepstrales en las Frecuencias de Mel (MFCC). Se genera un MFCC por cada archivo de audio mediante la configuración y uso de la herramienta sphinx_fe.

La adaptación consiste en generar transformaciones en la media y varianza de las emisiones de los estados de los HMM para que se asemejen más a la data de adaptación (Young et al., 2002).

Finalmente, con los archivos generados, se procede a la creación del nuevo modelo acústico adaptado al hablante.

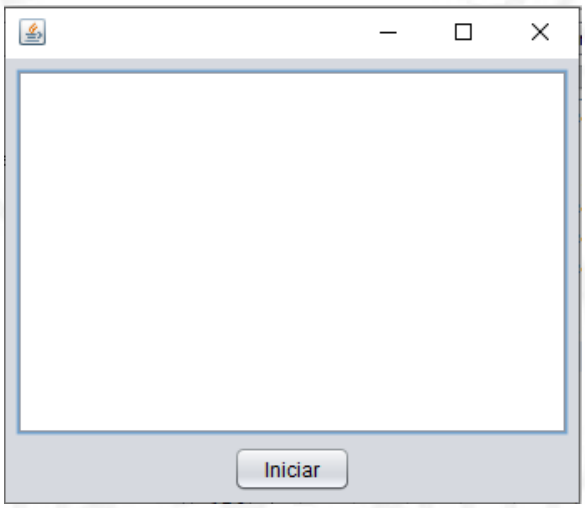
5.5 Prototipo desarrollado

Se desarrolló una aplicación en Java para demostrar la funcionalidad del modelo entrenado. La aplicación utiliza las tres partes del lenguaje mencionadas previamente:

modelo acústico, modelo de lenguaje y diccionario fonético. Las tres partes son las que se utilizaron durante el proceso de desarrollo de la solución. El modelo de lenguaje es el que se obtuvo del corpus CIEMPIESS y que se utilizó durante el entrenamiento del modelo acústico. El diccionario fonético es el diccionario reducido de 18 palabras mencionado en el punto 5.4.2. Finalmente, el modelo acústico utilizado es uno propio luego de realizar el entrenamiento y la adaptación del modelo acústico mencionados en los puntos 5.3 y 5.4.3, respectivamente.

Figura 5.11

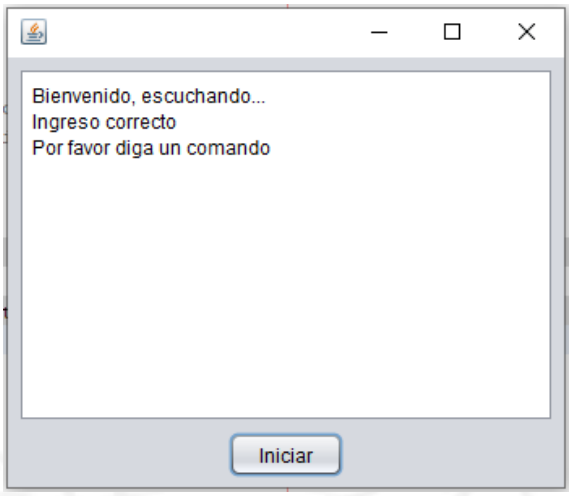
Pantalla inicial de la aplicación de demostración.



La aplicación empieza el proceso de reconocimiento al hacer clic en el botón Iniciar como se muestra en la figura 5.11. La aplicación está programada para que esta esté a la espera a una palabra clave antes de empezar a reconocer el resto de los comandos. En este caso, la palabra clave utilizada es ‘hola’.

Figura 5.12

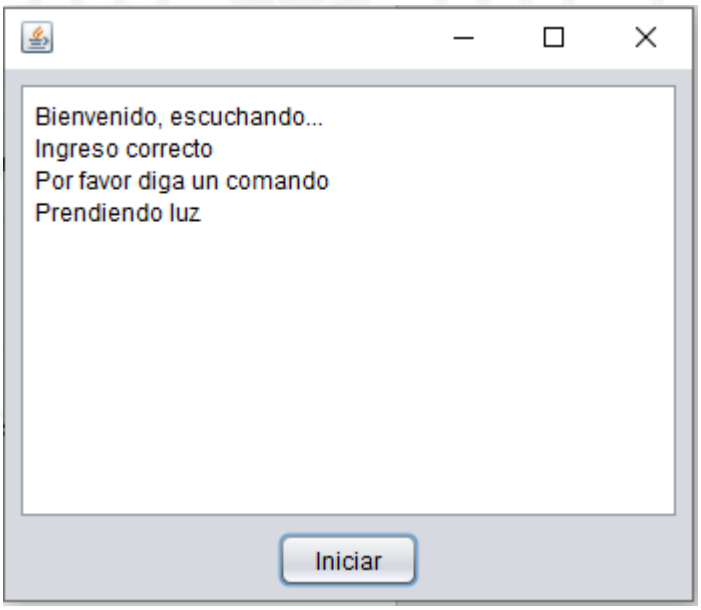
Aplicación solicitando un comando luego de validar que se haya dicho la palabra clave.



La aplicación muestra una respuesta distinta de acuerdo al comando reconocido. Por ejemplo, al decir el comando ‘Luz amarilla’, la aplicación devuelve el mensaje: ‘Prendiendo luz’.

Figura 5.13

Aplicación luego de identificar el comando ‘Luz amarilla’.



Además, al decir las palabras ‘pausa’ o ‘reanudar’, se interrumpe el reconocimiento o se reactiva, respectivamente.

Se creó un escenario para demostrar la funcionalidad en internet de las cosas conectando la app a un Arduino. Para esto se usaron los siguientes equipos:

- Laptop
- Micrófono Blue Snowball
- Arduino Uno
- 3 resistencias de 330Ohm
- Cables Jumper Macho – Macho
- 3 leds de distintos colores (rojo, amarillo y verde).

En la simulación realizada, cada color de led simulaba un aparato. El led rojo simulaba unas cortinas, el amarillo una luz amarilla y el verde una computadora. Se utilizaron los siguientes comandos:

- Hola: Activa el reconocimiento.
- Abrir cortinas: Enciende el led rojo.
- Luz amarilla: Enciende el led amarillo.
- Cerrar cortinas: Apaga el led rojo.
- Computadora: Enciende el led verde.
- Pausa: Pausa el reconocimiento.
- Abrir cortinas: No realiza nada, debido a que el reconocimiento se encuentra pausado.
- Reanudar: Reanuda el reconocimiento.
- Abrir cortinas: Enciende el led rojo.
- Apagar todo: Apaga todos los leds.

Un video demostrativo del funcionamiento completo del prototipo y las pruebas realizadas se puede acceder en el siguiente link: https://www.youtube.com/watch?v=vKyXNt0e1Uw.

CAPÍTULO VI: PRUEBAS Y RESULTADOS

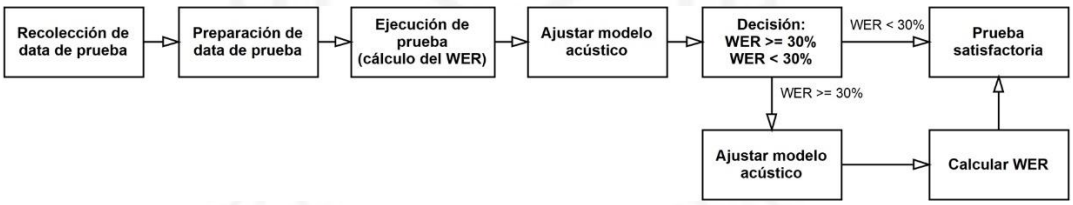
6.1 Introducción

La validación del trabajo consiste en obtener el porcentaje de error por cada persona de prueba en distintas condiciones. Este porcentaje de error es conocido como Word Error Rate (Carnegie Mellon University, 2008).

Para calcularlo fue necesaria la obtención de data de prueba (voces). El Corpus CIEMPIESS contiene data de prueba de reconocimiento general, mientras que para las pruebas de comandos específicos se generó data propia mediante la grabación de 20 personas distintas (10 hombres y 10 mujeres) que no presentan dificultad para hablar. El trabajo está orientado a personas con discapacidad que dificulte o impida la movilidad, mas no el habla. Por lo tanto, las pruebas pueden realizarse con cualquier persona, siempre y cuando, no presente dificultades con el habla. En caso el resultado no sea el esperado, se adaptó el modelo acústico según lo explicado en el punto 5.2 hasta obtener el resultado deseado. Gráficamente se representa el flujo en la figura 6.1:

Figura 6.1

Proceso de validación.



6.2 Conceptos Previos

6.2.1 Word Error Rate

Para medir la exactitud de un reconocedor de voz se utiliza un factor conocido como Word Error Rate (WER). El WER indica el porcentaje de error que se ha conseguido basado en el modelo acústico entrenado y al modelo de lenguaje (Clemente et al., 2001).

Este porcentaje considera tres tipos de errores distintos (Clemente et al., 2001):

- Error de sustitución. - Se da cuando la palabra reconocida se sustituye por otra que no es correcta.
- Error de inserción. - Se da cuando al momento del reconocimiento, se inserta una palabra no dicha por el hablante.
- Error de omisión. - Se da cuando el reconocedor omite una palabra mencionada por el hablante.

Para calcular el WER, se sigue la siguiente fórmula:

WER = ((sustituciones + inserciones + omisiones) / número total de palabras evaluadas) * 100%

El WER puede ser superior al 100%, debido a los errores de inserción. Una aplicación de reconocimiento de voz puede inclusive funcionar con un WER de 66%; sin embargo, a partir del 30% la precisión del reconocedor empieza a decaer bastante (Johnson et al., 1999).

6.3 Objetivo

El objetivo de la validación es verificar que la aplicación funcione correctamente para los comandos de voz previamente entrenados. La aplicación debe reconocer el comando dado por cualquier persona. Para esto, se pondrá a prueba el modelo y, en base a los resultados obtenidos, se determinará su viabilidad.

6.4 Experimento

6.4.1 Requisitos

Para llevar a cabo la validación, fue necesario reunir dos bases de datos de voces de pruebas la cuales deben cumplir con dos condiciones: el material grabado debe ser distinto al usado en el entrenamiento y el tamaño debe ser de aproximadamente el 10% del Corpus de entrenamiento. Las dos bases de datos reunidas son: Data de prueba general y data de prueba específica. Ambas serán usadas en distintos escenarios de prueba detallados más adelante.

La data de prueba general fue obtenida del Corpus CIEMPIESS de la UNAM y se le han realizado ajustes menores para que funcione sin problemas sobre el CMUSphinx. Esta base de datos se utilizó para la prueba general que incluye a distintos hablantes.

En el caso de la prueba de los comandos específicos, se generó data propia de 20 hablantes distintos, 10 hombres y 10 mujeres. El número de personas se eligió en base a trabajos previos, donde se utiliza una cantidad de hablantes similar (Leggetter y Woodland, 1995a) o menor (Leggetter y Woodland, 1995b). Esta data no cumple con ser el 10% del entrenamiento, sino que cubre toda la data entrenada. Esto fue posible, debido a que la data usada para adaptar el modelo acústico es reducida y se puede crear data de prueba que abarque todos los comandos. Esta base de datos se utilizó para las pruebas específicas.

6.4.2 Pruebas

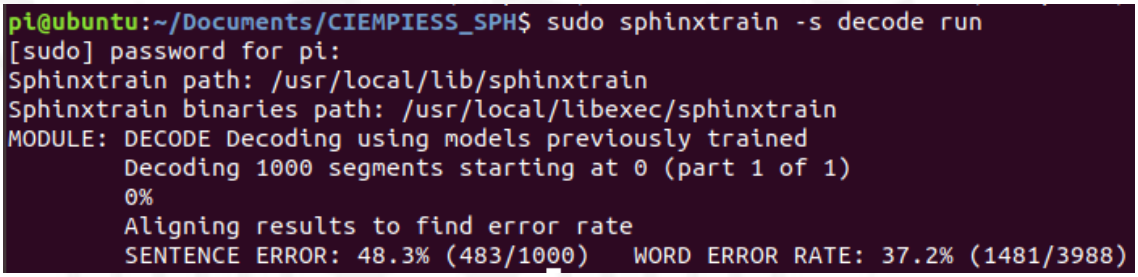
Sphinxtrain, una parte del CMUSphinx, provee una herramienta para calcular el WER y observar el grado de error/acierto que se ha logrado. Se definieron 5 escenarios de prueba en los que se puede observar la evolución del WER a lo largo del desarrollo del sistema:

- Escenario 1: Diccionario original, modelo acústico original y data de prueba general del corpus CIEMPIESS (varios hablantes).
- Escenario 2: Diccionario original, modelo acústico original y data de prueba específica (20 hablantes).
- Escenario 3: Diccionario original, modelo acústico adaptado y data de prueba específica (20 hablantes).
- Escenario 4: Diccionario reducido, modelo acústico original y data de prueba específica (20 hablantes).
- Escenario 5: Diccionario reducido, modelo acústico adaptado y data de prueba específica (20 hablantes).

El primer escenario es un caso único, en el que se utiliza valida utilizando la data de prueba brindada por el corpus CIEMPIESS que consiste en 1000 frases dichas por 170 hablantes (97 hombres y 73 mujeres). Este escenario sirve como punto de referencia para observar como la reducción del vocabulario y la adaptación al usuario otorgan menores porcentajes de error. En el resto de los escenarios se utiliza la data propia con los 20 hablantes mencionados en la introducción. Para el escenario 1, solo se presenta el resultado general debido a que solo se utiliza como punto de referencia. En los escenarios 2, 3, 4 y 5 se presentan los resultados promedio y el detalle, ya que en estos escenarios se están probando distintas situaciones para la data propia. Los resultados brindados por esta data han sido analizados y son los que determinan la viabilidad del sistema.

Figura 6.2

Proceso de validación del escenario 1.



Para los escenarios 2, 3, 4 y 5 las palabras y frases probadas por los 20 hablantes fueron:

- Luz
- Emite luz
- Luz amarilla
- Luz blanca
- Luz de la cocina
- Luz del cuarto
- Luz de la sala
- Apagar luz

- Apagar luz amarilla
- Apagar luz blanca
- Apagar luz de la cocina
- Apagar luz del cuarto
- Apagar luz de la sala
- Apagar todas las luces
- Abre la cortina
- Cierra la cortina
- Hola

En todos los escenarios se usaron las mismas frases y palabras. Cada uno de los 20 hablantes realizó un único intento.

Los resultados promedio se muestran en la tabla 6.1:

Tabla 6.1

Escenarios con sus respectivos WER.

Escenario	WER (%)
Escenario 1	37.2%
Escenario 2	78.7%
Escenario 3	74.7%
Escenario 4	27.0%
Escenario 5	19.0%

Los resultados a detalle de los 4 escenarios y los 20 hablantes se muestran en la tabla 6.2:

Tabla 6.2

Detalle de las pruebas por hablante.

Hablante	Escenario 2	Escenario 3	Escenario 4	Escenario 5
Hablante 01	100%	98%	70.60%	35.30%
Hablante 02	64.7%	56.7%	19.6%	15.7%
Hablante 03	100.0%	86.3%	35.3%	31.4%
Hablante 04	102.0%	98.0%	27.4%	23.5%
Hablante 05	47.1%	58.8%	25.5%	13.7%
Hablante 06	82.3%	76.5%	45.1%	39.2%
Hablante 07	84.3%	84.3%	27.4%	19.6%
Hablante 08	78.4%	70.6%	33.3%	25.5%
Hablante 09	49.0%	47.1%	15.7%	23.5%
Hablante 10	78.4%	72.5%	15.7%	7.8%
Hablante 11	72.5%	68.6%	21.6%	11.8%
Hablante 12	52.9%	47.1%	17.6%	9.8%
Hablante 13	98.0%	96.1%	35.3%	27.5%
Hablante 14	94.1%	90.2%	19.6%	9.8%
Hablante 15	90.2%	86.3%	17.6%	7.8%
Hablante 16	74.5%	68.6%	17.6%	9.8%
Hablante 17	70.6%	72.5%	7.8%	3.9%
Hablante 18	82.3%	76.5%	35.3%	27.5%
Hablante 19	74.5%	68.6%	9.8%	3.9%
Hablante 20	78.4%	70.6%	41.2%	33.3%
Promedio	78.7%	74.7%	27.0%	19.0%

6.5 Discusión de los resultados

En la tabla 6.1 se pueden observar los resultados promedio arrojados en cada escenario. El escenario 1, como se mencionó previamente, sirve como punto referencia para el análisis del resto de escenarios. En este, se valida si el reconocedor sin ninguna modificación sirve como un reconocedor de propósito general, es decir, como un reconocedor de un gran número de palabras y que funcione independientemente del hablante.

Los resultados obtenidos en este escenario demuestran que es posible usar el sistema como un reconocedor de propósito general, contando con un WER de 37%, una cifra suficiente para el buen funcionamiento de un reconocedor de voz (Johnson et al., 1999).

Con este punto de referencia, se proceden a analizar los resultados del resto de escenarios. El objetivo es que en el escenario 5, luego de aplicar la reducción del vocabulario y la adaptación al usuario, se tenga el menor WER de todos los escenarios.

En el escenario 2 se prueban las 20 frases en todo el universo de palabras del escenario 1 (53169) y sin ninguna clase de adaptación concluyendo que el reconocedor sin ninguna clase de adaptación es prácticamente inútil, pues cuenta con un porcentaje demasiado alto de error (78.7%). La reducción de la cantidad de palabras a reconocer disminuyó enormemente el porcentaje de error, logrando un 27% como se puede observar en el escenario 4.

Finalmente, la adaptación del modelo acústico disminuyó el WER en 4% en el escenario 3 respecto al escenario 2 y 8% en el escenario 5 respecto al escenario 4. La disminución luego de este proceso se situó en el rango recomendado de alrededor del 10%, por lo que estos valores son adecuados (Carnegie Mellon University, 2016). Gracias a la reducción del diccionario y a la adaptación del modelo acústico, se logra conseguir el menor WER de todos en el escenario 5 (19%), siendo este menor al 30%, consiguiendo así un reconocedor con un buen funcionamiento (Johnson et al., 1999). Analizando el detalle de tabla 6.2, se observa que 17 de 20 casos cumplieron con un WER menor a 30%, llegando inclusive a tener un caso con un WER de 3.9%. Con estos resultados, se demostró que el reconocedor es perfectamente viable para su uso.

6.6 Comparativa

6.6.1 Comparativa de costes

Uno de los objetivos específicos del presente trabajo es obtener el menor costo posible. El bajo costo, logrado gracias al uso de software que no requiere licencia como Sphinx, Java o Arduino, y la posibilidad de uso del reconocedor sin necesidad de internet son las principales diferencias respecto a otros reconocedores en el mercado.

La comparativa de costos se enfocó en el reconocedor en sí, ya sea fuera de línea u online. Esto quiere decir que el costo analizado es el del dispositivo en el que funcionará el sistema más el precio del uso por el reconocedor (en caso aplique). Siempre se eligió el dispositivo más barato sobre el que puede funcionar el reconocedor. Todos los costos presentados utilizan como referencia el precio oficial de Estados Unidos, pues el costo

en Perú es variable de acuerdo con el proveedor y al momento. En el caso del WER, se utilizaron los últimos datos oficiales publicados, en caso este dato se encuentre disponible.

Los aparatos extras que conforman el entorno domótico (luces, termostato, entre otros) no fueron considerados para la comparativa de precios, debido a que el uso de estos dependerá completamente del ambiente en el que será instalada la solución. Sí se consideró como un apartado independiente la compatibilidad con otros dispositivos, ya que, a mayor compatibilidad, mayor cantidad de opciones por elegir y, por lo tanto, mayor libertad al momento de elegir entre distintos precios (Pham-Huu et al., 2015).

Tabla 6.3

Comparativa de costes de reconocedores.

Herramienta	Solución Propuesta	CMUSphinx	Google Cloud Speech API	WATSON (IBM)	AT&T	Alexa (Amazon)	Siri (Apple)	Julius	Cortana (Microsoft) - Versión para Winwos 10 IoT
Dispositivo más económico sobre el que funciona	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Echo Dot	HomePod	Raspberry Pi	Raspberry Pi
Precio del Dispositivo	$35	$35	$35	$35	$35	$50	$299	$35	$35
Precio del software	Gratuito	Gratuito	60 minutos gratis. $0.004 - $0.006 por cada 15 segundos después de eso.	$0.02 por minuto. El precio disminuye después de 250000 minutos	$250 por el SDK que incluye una prueba de 90 días. El costo de la licencia final es variable.	Gratuito	Gratuito	Gratuito	Gratuito
Compatibilidad con dispositivos inteligentes	Libre	Libre	Libre	Libre	Libre	Solo con dispositivos certificados como compatibles	Solo con dispositivos certificados como compatibles	Libre	Libre

Nota. Información obtenida de distintas páginas oficiales.

Fuentes: Carnegie Mellon University (2020); Kawahara et al., 2000; AT&T Natural Voices SDK, 2020; AVS Frequently Asked Questions, 2020; Cloud Text-to-Speech, 2020; Developer Cortana, 2020; Products: Raspberry Pi, 2020; Watson Speech to Text, 2020; HomePod: Apple, 2020

En la comparativa se observan los distintos rangos de precio, desde los $35 hasta los $299 de la alternativa de Apple. La solución desarrollada se encuentra entre las más baratas; tanto por el precio del dispositivo como por la posibilidad de conectarse con cualquier dispositivo sin la necesidad de que este esté específicamente diseñado para la solución. Con esto, se abre un amplio abanico de posibilidades para la elección de los dispositivos, pudiendo elegir opciones más económicas. Este bajo precio solo es igualado por Julius y por Cortana, la alternativa de Microsoft.

6.6.2 Comparativa de funcionalidad

En esta sección se han comparado las mismas alternativas analizadas en la comparativa anterior, pero a nivel de funcionalidad. Se evalúan los siguientes aspectos:

- Compatibilidad con el español latino: Se menciona si es compatible o no con el español nativo y si este soporte es nativo o se requiere algún proceso de adaptación.
- Funcionamiento: Se menciona el modelo estadístico sobre el cual está construido el reconocedor.
- Domótica: Se menciona si puede ser usado para domótica y qué se necesita para esa compatibilidad.
- Dependiente de internet: Se menciona si requiere o no internet para su funcionamiento.
- Otros: Se mencionan otros aspectos a considerar.
- WER: Se menciona el porcentaje de error del reconocedor si este está disponible.

Tabla 6.4

Comparativa funcional de reconocedores.

Herramienta	Solución Propuesta	CMUSphinx	Google Cloud Speech API	WATSON (IBM)	AT&T	Alexa (Amazon)	Siri (Apple)	Julius	Cortana (Microsoft) - Versión para Winwos 10 IoT
Soporte español latino	Sí	Mediante adaptación	Nativo	Nativo	Nativo	No	Nativo	Mediante adaptación	No
Funcionamiento	Modelos Ocultos de Markov	Modelos Ocultos de Markov	Redes Neuronales Profundas	Redes Neuronales Profundas	Redes Neuronales Profundas	Redes Neuronales Profundas	Redes Neuronales Profundas	Modelos Ocultos de Markov	Redes Neuronales Profundas
Domótica	Sí, mediante aplicación propia y el uso de Arduino	Sí, mediante la creación o uso de aplicaciones compatibles.	Sí, mediante la creación o uso de aplicaciones compatibles.	Sí, mediante la creación o uso de aplicaciones compatibles.	Sí, mediante la creación o uso de aplicaciones compatibles.	Soporte nativo	Soporte nativo	Sí, mediante la creación o uso de aplicaciones compatibles.	Sí, mediante la creación o uso de aplicaciones compatibles.
Dependiente de Internet	No	No	Sí	Sí	Sí	Sí	Sí	No	Sí
Otros	-	-	-	-	-	-	-	Julius funciona nativamente en japonés. Existe muy poca información y herramientas para su uso en español.	Windows 10 IoT es una versión especial para Internet de las cosas. Es más limitada, pero gratuita.
WER	19%	De acuerdo a entrenamiento	4.90%	5.10%	No disponible	No disponible	No disponible	De acuerdo a entrenamiento	6.30%

Nota. Información obtenida de distintas páginas oficiales.

Fuentes: Carnegie Mellon University (2016), Kawahara et al (2000), AT&T Natural Voices SDK (2020), AVS Frequently Asked Questions (2020), Protalinski (2017), Saon (2017), Siri Team (2017)

Lo primero a analizar en la comparativa es si la alternativa presenta soporte en español latino. La gran mayoría de propuestas soportan el idioma, con las excepciones de Alexa de Amazon y Cortana de Microsoft.

En el funcionamiento solo hay dos alternativas: modelos ocultos de Markov y redes neuronales profundas. El impacto que tiene esta elección está en la dependencia de internet, notándose que cuando se usan HMM el uso de internet no es necesario.

Respecto a la posibilidad de uso en domótica, todas las alternativas son compatibles, ya sea de manera nativa o mediante adaptación.

En el análisis del WER se observa que la solución propuesta tiene el porcentaje de error más alto dentro de las alternativas que mencionan este dato; sin embargo, el porcentaje logrado es válido para el correcto funcionamiento del reconocedor.

Finalmente, se puede observar que ninguna alternativa, salvo Julius, presenta una propuesta parecida. La desventaja de Julius está en que la adaptación al español es una tarea mucho más complicada que en CMUSphinx debido a la falta de recursos y documentación. Además, las alternativas que cuentan con ciertas ventajas como un WER menor, presentan otras desventajas como la obligatoriedad de internet para su uso o un mayor costo como se desprende de la comparativa anterior.

En conclusión, la solución desarrollada presenta unas características únicas, siendo Julius la única alternativa viable, pero que cuenta con graves desventajas como se mencionó en el párrafo anterior. Esto no quiere decir que el reconocedor desarrollado sea superior al resto de propuestas actuales, sino que cubre un espacio único que la competencia no contempla.

CONCLUSIONES Y RECOMENDACIONES

Se demostró que es posible crear un reconocedor de comandos de voz basado en modelos ocultos de Markov, el cual logró un porcentaje de error de 19% en el escenario 5 luego de aplicar todas las mejoras (reducción del vocabulario y adaptación al usuario). Esta cifra es aceptable para que el reconocedor pueda ser utilizado en aplicaciones reales según Johnson et al. (1999). Además, con el objetivo de apoyar a personas con discapacidad física, se simuló su uso en un sistema domótico mínimo que simulaba la realización de acciones automatizadas en el hogar.

Otro aspecto importante es el cumplimiento del objetivo de implementar un sistema de reconocimiento de voz sobre plataformas de libre acceso como Java y Arduino, logrando así un bajo costo, lo que permite tener un sistema de control por voz accesible.

Así mismo, la solución propuesta abarca un segmento al que no se le presta mucha atención, el uso de los sistemas de reconocimiento de voz para apoyar a personas con discapacidad. Con el valor agregado de la no necesidad de internet y el uso de español latino como idioma de reconocimiento.

El porcentaje de error obtenido en las pruebas evidenció la viabilidad del sistema, a pesar de ser mayor a la de otros sistemas como se pudo observar en la tabla 6.4. Sin embargo, continúa estando dentro del rango aceptable para los sistemas de reconocimiento de voz. Además, el sistema cuenta con otras ventajas como el bajo costo y mayor seguridad anteriormente mencionados.

Por último, a pesar de que la propuesta está diseñada para apoyar a personas con discapacidad física; con los ajustes necesarios, podría utilizarse para otros propósitos como dictado automático de textos, aplicaciones de seguridad, entre otros. Para generar una nueva solución que use lo aplicado en este trabajo, es necesario primero analizar cuál es el objetivo y en base a ello determinar los requisitos. Estos requisitos pueden variar de acuerdo con la complejidad de la solución y a qué tanto se deba modificar la solución propuesta en este trabajo. Por ejemplo, para crear un software para el dictado automático

de textos se requiere mucha más data para entrenar al modelo y ampliar las capacidades del reconocedor. En conclusión, la solución puede adaptarse para otros usos con una complejidad que depende del objetivo.

REFERENCIAS

Antón, J. (2015). Desarrollo de un sistema de reconocimiento de habla natural independiente del locutor [tesis de licenciatura, Universidad Autónoma de Madrid]. Departamento de Tecnología Electrónica y de las Comunicaciones, España. https://repositorio.uam.es/handle/10486/667850

Apple. (2019). Usar Siri en todos tus dispositivos de Apple. https://support.apple.com/es-es/HT204389

Aqeel-ur-Rehman, R. A. y Khursheed, H. (2014). Voice controlled home automation system for the elderly or disabled people. J. *Appl. Environ. Biol. Sci*, *4*(8S), 55-64.

Asociación Española de Domótica e Inmótica. (s. f.). *Qué es Domótica*. http://www.cedom.es/sobre-domotica/que-es-domotica.

AT&T Natural Voices SDK. (2020). Wizzard. http://www.wizzardsoftware.com/text-to-speech-sdk.php

AVS Frequently Asked Questions. (2020). Amazon Alexa. https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/faqs.html

Bernal, J., Gómez, P., y Bobadilla, J. (1999). *Una visión práctica en el uso de la Transformada de Fourier como herramienta para el análisis espectral de la voz*. Estudios de fonética experimental, *10*, 75-105.

Biewald, L. (2017). *Build a super fast deep learning machine for under $1,000*. https://www.oreilly.com/learning/build-a-super-fast-deep-learning-machine-for-under-1000

Bongaarts, J. (2009). Human population growth and the demographic transition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1532), 2985-2990.

Brush, A. J, Lee, B., Mahajan, R., Agarwal, S., Saroiu, S. y Dixon, C. (2011, mayo). Home automation in the wild: challenges and opportunities. In proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2115-2124). ACM.

Carbureanu, M., Mihalache, S. F. y Gheorghe, C. G. (2016). A Hardware ANN-based Controller for pH Neutralization in Industrial Plants. *Revista de Chimie*, *67*(7), 1309-1313.

Carnegie Mellon University. (2008). Learning to use the CMU SPHINX Automatic Speech Recognition system. http://www.speech.cs.cmu.edu/sphinx/tutorial.html#app3

Carnegie Mellon University. (2016). OPEN SOURCE SPEECH RECOGNITION TOOLKIT. https://cmusphinx.github.io/

Carnegie Mellon University. (2020). Adaptation of the acoustic model. https://cmusphinx.github.io/wiki/tutorialadapt/

Clemente, E., Vargas, A., Olivier, A., Kirschning, I., y Cervantes, O. (2001). *Entrenamiento y Evaluación de reconocedores de Voz de Propósito General basados en Redes Neuronales feed-forward y Modelos Ocultos de Markov* [tesis de licenciatura, Universidad de las Américas]. Puebl, Dept. Computer Systems Engineering, México.

Cloud Text-to-Speech. (2020). Google Clou. https://cloud.google.com/text-to-speech

Dennis, A. (2013). *Raspberry Pi home automation with Arduino*. Packt Publishing Ltd.

Developer Cortana. (2020). Microsoft. https://developer.microsoft.com/es-es/cortana/

Gales, M. y Young, S. (2008). *The application of hidden Markov models in speech recognition*. *Foundations and trends in signal processing, 1*(3), 195-304.

Hernández-Mena, C. y Herrera-Camacho, J., *CIEMPIESS: A New Open-Sourced Mexican Spanish Radio Corpus*, In LREC. European Language Resources Association, 2014.

Hernando, J. (1993). *Técnicas de procesado y representación de la señal de voz para el reconocimiento del habla en ambientes ruidosos*. Universidad Politécnica de Cataluña, España.

Herrera, L. (2005). Viviendas inteligentes (domótica). *Ingeniería e investigación, 25*(2), 47-53.

Hi-Fi S.A.C. (2020). Hi-Fi S.A.C. https://www.hifisac.com/shop

HomePod: Apple. (2020). Apple. https://www.apple.com/shop/buy-homepod/homepod

Huang, X., Baker, J. y Reddy, R. (2014). A historical perspective of speech recognition. *Communications of the ACM, 57*(1), 94-103.

Instituto Nacional de Estadística e Informática. (2014). *Primera Encuesta Nacional Especializada Sobre Discapacidad 2012*. https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1171/ENEDIS%202012%20-%20COMPLETO.pdf

Jabbar, A. S., Abbas, E. I. y Safi, M. E. (2018). *Wireless Home Automation Control Using Voice Recognition Based on HM2007*. *University of Thi-Qar Journal, 13*(2), 138-150.

Johnson, S. E., Jourlin, P., Moore, G. L., Jones, K. S. y Woodland, P. C. (1999). The Cambridge University spoken document retrieval system. In *Acoustics, Speech, and Signal Processing, Proceedings, IEEE International Conference on* (vol. 1, pp. 49-52). IEEE.

Juang, B. H. y Rabiner, L. R. (2005). *Automatic speech recognition–a brief history of the technology development*. Georgia Institute of Technology, Atlanta Rutgers University and the University of California. *Santa Barbara*, *1*, 67.

Kawahara, T., Lee, A., Kobayashi, T., Takeda, K., Minematsu, N., Sagayama, S., Itou, K. Itou, A., Yamamoto, M. Yamada, A. Utsuro, T. y Shikano, K. (2000). *Free software toolkit for Japanese large vocabulary continuous speech recognition*.

Kraus, L. (2015). *Disability Statistics Annual Report*. https://disabilitycompendium.org/sites/default/files/user-uploads/2015%20Annual%20Report%20%28PDF%29.pdf.

Leggett, J. y Williams, G. (1984). An empirical investigation of voice as an input modality for computer programming. *International Journal of Man-Machine Studies, 21*(6), 493-520.

Leggetter, C. J., y Woodland, P. C. (1995b). *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*. Computer speech y language, 9(2), 171-185.

Leggetter, C. y Woodland, P. (1995a). *Flexible speaker adaptation using maximum likelihood linear regression*. In Proc. ARPA spoken language technology workshop (Vol. 9, pp. 110-115).

López, N., Ponce, S., Piccinini, D., Perez, E., y Roberti, M. (2016). *From hospital to home care: Creating a domotic environment for elderly and disabled people*. IEEE pulse, 7(3), 38-41.

Markets and Markets. (2016). *Home Automation System Market worth 78.27 Billion USD by 2022*. Recuperado de: http://www.marketsandmarkets.com/

Martínez, F., Portale, G., Klein, H., y Olmos, O (s. f.). *Reconocimiento de voz, apuntes de cátedra para Introducción a la Inteligencia Artificial*. Recuperado de: http://www.secyt.frba.utn.edu.ar/giar/IA1_IntroReconocimientoVoz.pdf

Milone, D. (2004). *Modelos ocultos de Markov para el reconocimiento automático del habla*. Recuperado de: http://tsam-fich.wdfiles.com/local--files/apuntes/hmmdiet.pdf.

Miori, V., Tarrini, L., Manca, M. y Tolomei, G. (2006). An open standard solution for domotic interoperability. *IEEE Transactions on Consumer Electronics, 52*(1), 97-103.

Mosbah, B. B. (2006, abril). Speech recognition for disabilities people. In *2nd International Conference on Information & Communication Technologies* (vol. 1, pp. 864-869). IEEE.

Nilsson, M. y Ejnarsson, M. (2002). *Speech Recognition using Hidden Markov Model*. http://urn.kb.se/resolve?urn=urn:nbn:se:bth-3946

Noyes, J. M., Haigh, R. y Starr, A. F. (1989). Automatic speech recognition for disabled people. *Applied Ergonomics, 20*(4), 293-298.

Noyes, J. y Frankish, C. (1992). Speech recognition technology for individuals with disabilities. *Augmentative and Alternative Communication, 8*(4), 297-303.

Obaid, T., Rashed, H., El Nour, A. A., Rehan, M., Saleh, M. M., y Tarique, M. (2014). *ZigBee based voice controlled wireless smart home system*. *International Journal of Wireless y Mobile Networks, 6*(1), 47.

Organización Mundial de la Salud. (2018). *Discapacidad y Salud*. http://www.who.int/mediacentre/factsheets/fs352/es/

Oropeza Rodríguez, J. L. y Suárez Guerra, S. (2006). Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. *Computación y sistemas, 9*(3), 270-286.

Pham-Huu, D. N., Nguyen, V. H., Trinh, V. A., Bui, V. H. y Pham, H. A. (2015, November). Towards an open framework for home automation development. In *Advanced Computing and Applications* (ACOMP), *International Conference on* (pp. 75-81). IEEE.

Prasanna, G. y Ramadass, N. (2014). Low Cost Home Automation Using Offline Speech Recognition. *International Journal of Signal Processing Systems, 2*(2), 96-101.

Prieto, F. y Jara, E. A. M. (2013). Control de luces por voz, una aplicación orientada a personas con capacidades especiales. *FPUNE Scientific, 8*(8).

Products: Raspberry Pi. (2020). Raspberry Pi. https://www.raspberrypi.org/products/

Protalinski, E. (2017). Venture Beat. https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/

Putthapipat, P., Woralert, C. y Sirinimnuankul, P. (2018, enero). Speech recognition gateway for home automation on open platform. *In International Conference on Electronics, Information, and Communication* (ICEIC) (pp. 1-4). IEEE.

Qiao, F., Sherwani, J. y Rosenfeld, R. (2010). Small-vocabulary speech recognition for resource-scarce languages. In *Proceedings of the First ACM Symposium on Computing for Development* (p. 3). ACM.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Rodríguez, A. (2014). *Diseño de un sistema domótico centralizado*. Universidad de Valladolid, España.

Sanclemente, C. (2016). *Casa domótica con Arduino*. Universidad Politécnica de Valencia, España.

Saon, G. (2017). *Reaching new records in speech recognition*. IBM. https://www.ibm.com/blogs/watson/2017/03/reaching-new-records-in-speech-recognition/

Singh, R. (2003). *Designing HMM-based ASR systems*. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/lecture-notes/lecture15.pdf

Siri Team. (2017). *Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant*. https://machinelearning.apple.com/2017/10/01/hey-siri.html

Srinivasan, A. (2011). Speech recognition using Hidden Markov model. *Applied Mathematical Sciences, 5*(79), 3943-3948.

Torrente, O. (2013). *ARDUINO. Curso práctico de formación*. RC Libros.

Wang, Z., Schultz, T., y Waibel, A. (2003). *Comparison of acoustic model adaptation techniques on non-native speech*. In Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on (Vol. 1, pp. I-I). IEEE.

Watson Speech to Text. (2020). IBM. https://www.ibm.com/pe-es/cloud/watson-speech-to-text/pricing

Wildstrom, S. (2011). *Nuance Exec on iPhone 4S, Siri, and the Future of Speech*. https://techpinions.com/nuance-exec-on-iphone-4s-siri-and-the-future-ofspeech/3307.

Wimsatt, W. (2004). *U.S. Patent Application No. 10/820,434*.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, Odell, Ollason, Povey, Ragni, Valtchev, Woodland y Valtchev, V. (2002). The HTK book. *Cambridge university engineering department, 3*, 175.

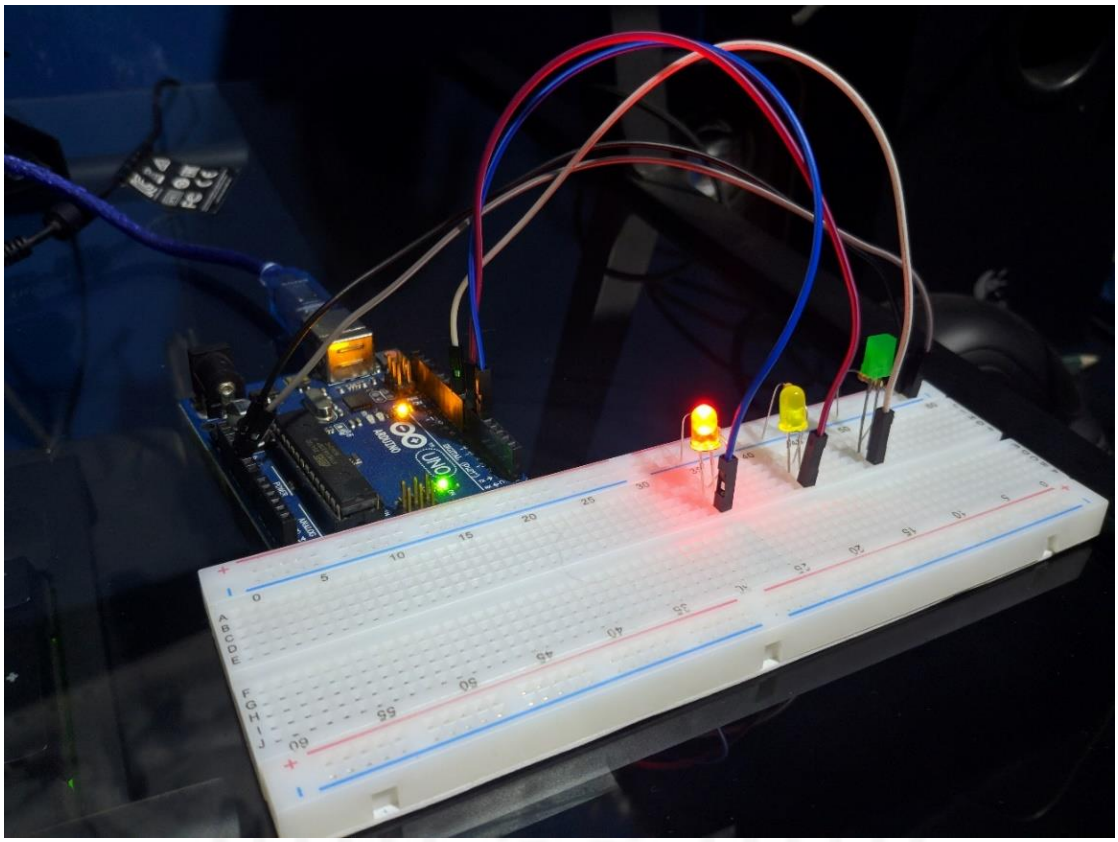
Yuksekkaya, B., Kayalar, A. A., Tosun, M. B., Ozcan, M. K. y Alkar, A. Z. (2006). A GSM, internet and speech controlled wireless interactive home automation system. *IEEE Transactions on Consumer Electronics, 52*(3), 837-843.

Yunaningsih, R. (2009). X10 Protocol Man Machine Interface Implementation using Labview. *Journal of Applied Science, 9*(19), 3562-3568.

ANEXOS

ANEXO 1: Materiales utilizados para la demostración

ANEXO 2: Led rojo encendido luego del comando ‘Abrir cortinas’



ANEXO 3: Led amarillo encendido luego del comando ‘Luz amarilla’