

Uso del asistente virtual Alexa como herramienta de interacción para el monitoreo de clima en hogares inteligentes por medio de Raspberry Pi y DarkSky API

Adrián Calvopiña¹, Freddy Tapia¹, Luis Tello-Oquendo²

ascalvopina@espe.edu.ec, fmtapia@espe.edu.ec, luis.tello@unach.edu.ec

¹ Computer Sciences Department, Universidad de las Fuerzas Armadas ESPE, Quito, Ecuador

² College of Engineering, Universidad Nacional de Chimborazo, Riobamba, Ecuador

DOI: 10.17013/risti.36.102–115

Resumen: En los últimos años, el desarrollo de la domótica ha aumentado gracias al uso de las capacidades de la Internet de las Cosas (IoT, por sus siglas en inglés) que proveen tecnologías emergentes como Raspberry Pi. Esto permite automatizar tareas en casa o monitorear parámetros específicos. Este artículo propone una mejora en la interacción entre el sistema domótico inteligente y el usuario final utilizando un asistente de voz inteligente en lugar de aplicaciones móviles o una plataforma web local. Además, se exploran otras fuentes de información para pronósticos meteorológicos, como las interfaces de programación de aplicaciones (API) que proporcionan este servicio.

Palabras-clave: Raspberry Pi; Asistente de voz inteligente; Clima; Alexa.

Using the Alexa virtual assistant as an interaction tool for climate monitoring in smart homes by Raspberry Pi and DarkSky API

Abstract: During the last few years, the development of home automation has increased thanks to the use of Internet of Things (IoT) capabilities provided by emerging technologies such as Raspberry Pi. This allows us to automate tasks at home or monitor specific parameters. This article proposes an improvement in the interaction between the smart home automation system and the end-user using an intelligent voice assistant instead of mobile applications or a local web platform. Furthermore, other sources of information for weather forecasts are explored, such as application programming interfaces (APIs) that provide this service.

Keywords: Raspberry Pi; Intelligent Voice Assistant; Weather; Alexa.

1. Introducción

La Internet de las cosas (IoT, por sus siglas en inglés) es un paradigma que se está volviendo cada vez más importante en nuestras actividades diarias, pero que aún no

ha sido explotado plenamente en los hogares. Es un concepto que escuchamos todo el tiempo. La idea que IoT intenta representar está bastante bien ilustrada por su nombre: cosas cotidianas que se conectan a Internet para realizar varias tareas; sin embargo, se trata de mucho más que eso. Para comprender de qué se trata IoT, también debemos entender que sus fundamentos no son nada nuevos.

Desde hace 30 años, se han realizado varias investigaciones con el objetivo de hacer que todos los objetos cotidianos sean un poco más interactivos. Conceptos como el hogar inteligente (Ricquebourg, et al., 2006) (Bugeja, Jacobsson, & Davidsson, 2018) también conocido como el hogar del mañana, han evolucionado antes de que nos demos cuenta de un hogar conectado para entrar en la era de IoT. Si se definiera IoT, sería mejor decir que es una red que interconecta objetos físicos usando Internet (Ma & Liu, 2017). Los objetos mencionados anteriormente utilizan sistemas embebidos o, en otras palabras, hardware especializado que no solo permite la conectividad a Internet, sino que también permite establecer eventos específicos de acuerdo con las tareas que se le dictan de forma remota (González García, Meana Llorián, Pelayo García-Bustelo, & Cueva Lovelle, 2017).

Las cosas se pueden clasificar como objetos que funcionan como sensores y objetos que realizan acciones activas. Cada uno de los objetos conectados a Internet tiene una dirección específica de protocolo de Internet (IP), y a través de esa dirección IP se puede acceder para recibir instrucciones. También puede contactar a un servidor externo y enviar los datos que recopila (Hipertextual, 2018). En un hogar inteligente, aparatos y dispositivos como luces, cerraduras, alarmas, refrigeradores, aspiradoras, televisores e incluso lavadoras se pueden manejar desde un smartphone, tablet o por voz y ser configurados para funcionar de forma automatizada.

En este estudio se propone una solución de hogar inteligente en la que la voz realiza la interacción entre el sistema y el usuario. Usamos la voz ya que es un medio de comunicación más natural y económico; específicamente, se propone el uso de Amazon Alexa (Hayes, 2017) como asistente inteligente. Siempre que se trata el tema de domótica o hogares inteligentes, existe la idea de que la interfaz final debe o puede ser una aplicación para un teléfono inteligente o una página web a la que se accede e interactúa con el sistema. En este estudio, proponemos cambiar este paradigma mediante el uso de Alexa Voice Service (AVS) (Whatis, 2018) y Raspberry Pi (Upton & Halfacree, 2012) utilizando comandos de voz para la interacción hombre-máquina.

El resto del documento está estructurado de la siguiente manera: La Sección II presenta el trabajo relacionado y analiza la evolución del uso de Amazon Alexa y Raspberry Pi. La Sección III presenta las áreas de conocimiento que abordamos y utilizamos en esta investigación. La Sección IV presenta la arquitectura propuesta, detalla sus componentes y cómo interactúan dentro de la arquitectura. La Sección V detalla la implementación de la solución y la configuración de los componentes. Finalmente, la Sección VI muestra las conclusiones y presenta el trabajo futuro planificado.

2. Trabajos relacionados

En (Jindarat & Wuttidittachotti, 2015), los autores proponen una “granja inteligente” con el uso de IoT implementado con Raspberry Pi, Arduino y varios sensores, para monitorear entornos, automatización de dispositivos electrónicos y recopilación de

datos. Por el contrario, el sistema que se propone en esta investigación elimina el soporte de Arduino para que Raspberry Pi por sí mismo realice la recopilación y el análisis de datos. Además, nuestra propuesta simplifica el sistema al no usar una base de datos; esto se debe a que no habrá informes o análisis posteriores con la información recopilada.

En otros estudios, el monitoreo inteligente de entornos se desarrolla mediante el uso de varios sensores, para vigilar la calidad del aire, la temperatura, la humedad, la intensidad de la luz y los movimientos sísmicos (Ibrahim, Elgamri, Babiker, & Mohamed, 2015). Aunque más datos aumentan el alcance del sistema, no se considera necesario centrarse tanto en los datos debido a que se presentan al usuario final de este sistema. En (Ibrahim, Elgamri, Babiker, & Mohamed, 2015), los autores utilizan el protocolo SSH para conectarse remotamente, lo que resulta en una interfaz para interactuar con el sistema bastante versátil pero rústico y primitivo. En (Rao & Uma, 2015) se plantean entornos más tradicionales, monitoreando el consumo de energía y la temperatura con el uso de IoT y el control de luces y puertas a través de domótica. Esto se realiza a través de una aplicación móvil desarrollada para Android, que accede al sistema a través de Internet a todos los servicios montados en una Raspberry Pi que sirve como unidad central de procesamiento. En el caso de un entorno inteligente, siempre es preferible una interacción más natural y fácil de, por lo tanto, en este estudio se considera el uso de Alexa como la interfaz final del sistema y se enfoca en la temperatura y el clima como campo de monitoreo.

En otros trabajos, diferentes asistentes de voz como Siri y Bixby también se plantean (Hoy, 2018), pero dentro de todas las opciones Alexa sigue siendo la mejor decisión porque Amazon permite dar más acciones a su lista de habilidades que permiten plantear proyectos interesantes como se menciona en (Kim, Row, & Nam, 2018) y otros más tradicionales como en (Aravindan & James, 2017). En estas investigaciones, Alexa es el software intermedio para el sistema de domótica/IoT; aunque también hay propuestas como en (Putthapipat, Woralert, & Sirinimnuankul, 2018) donde Amazon y Alexa se dejan de lado para comenzar a usar la API de Google para el discurso a texto que será procesado por Raspberry Pi. Google Translate también propone el uso de Wit. AI (Mitrevski, 2018) como la interfaz final del sistema que se conecta con la API de voz de Google para poder comunicarse con el usuario.

3. Marco conceptual

En esta sección se presentan diferentes áreas de conocimiento las cuales han sido utilizadas para el desarrollo de esta propuesta. Primero, asistente de voz inteligente, la cual se presenta como el enlace entre el usuario final y el sistema; principalmente a través del Amazon Voice Service (AVS, por sus siglas en inglés), el cual contribuye significativamente al estudio y tiene rol principal dentro de la arquitectura del sistema. Por otro lado, Raspberry Pi, se presenta como la estructura física que actúa como punto final para servicios AVS y permitir el monitoreo de los servicios.

3.1. Asistentes de voz inteligentes

Cada compañía que construye un asistente aplica métodos y acercamientos al desarrollo lo cual afecta al resultado final. Un asistente puede sintetizar palabras

más cualitativamente, otro puede hacerlo con más precisión y sin explicaciones ni correcciones adicionales, otros pueden realizar un rango más estrecho de tareas, pero con mayor precisión y como el usuario quiera. El conjunto de características que tiene un asistente depende completamente de las áreas a las que un desarrollador ha prestado mayor atención, dado que todos los sistemas se basan en el aprendizaje automatizado (Polyakov, et al., 2018).

Un asistente de voz inteligente se relaciona por medio de software, asociado con el sistema operativo de dispositivos específicos, el cual pueden interactuar con el ser humano a través de comandos de voz. Para esto, los asistentes de voz solo pueden responder preguntas simples, búsquedas de información, así como el pronóstico del tiempo o realizar acciones telefónicas básicas como “llamar a Jordi”.

Es por esto por lo que las versiones móviles de las páginas web comenzaron a priorizar y adaptar sus interfaces a los comandos de voz. Por ejemplo, estos asistentes están disponibles en muchos de los servicios de Google, desde la búsqueda del navegador hasta Google Maps y, por supuesto, en teléfonos Android, donde el usuario tiene solo que indicar lo que necesita saber y el dispositivo responderá de inmediato. Amazon promueve “echo”, una serie de altavoces inteligentes, para tener un asistente digital en casa. Esto es un cambio tan trascendental que significa olvidar la pantalla tradicional y empezar a interactuar de una manera radicalmente diferente por medio de la voz (Oleoshop, 2018).

Estos dispositivos introdujeron una interfaz conversacional dentro del hogar que permite a los usuarios el pedir y guardar información, controlar casas inteligentes y una serie de acciones dentro del internet. Por ejemplo, una persona con escasa movilidad podría controlar las luces dentro de la casa o las cerraduras usando la voz, también una persona con discapacidad visual puede preguntar la hora o información sobre el clima. Varios estudios encontraron que este nuevo paradigma de las interfaces visuales avanzadas (AVI, por sus siglas en inglés) tiene un tremendo potencial para inclusividad y accesibilidad (Pradhan, Mehta, & Findlater, 2018).

3.2. Servicio de voz ALEXA

Alexa Voice Service (AVS) es el conjunto de servicios de Amazon basado en su asistente de inteligencia artificial controlado por voz para el hogar y otros ambientes. AVS y Alexa se presentaron por primera vez con “echo”, el altavoz inteligente permite la interacción por medio de la voz con varios sistemas del medio y en línea. Alexa está disponible para un número creciente de otros dispositivos, incluidos teléfonos inteligentes, tabletas y controles remotos.

Amazon Alexa Voice Service (AAVS) proporciona servicios de control de voz para dispositivos de hardware integrados como Alexa, entre otros, la línea de productos de la familia Echo de Amazon y dispositivos domésticos inteligentes como termostatos y cámaras de seguridad.

En 2017, Amazon anunció un refrigerador LG compatible con Alexa y Volkswagen anunció que incrustaría Alexa directamente en sus sistemas de automóviles. Alexa Voice servicios no se limitan a productos de consumo. En noviembre 2017, Amazon anunció Alexa for Business, que es una extensión de Alexa Voice Services en el calendario

corporativo, reuniones de mercado, llamadas de conferencia y gestión de tareas. Se predice que 60 millones de dispositivos integrados de Alexa serán vendidos al 2020 y, con la integración actual de Alexa con más de 50 productos individuales de la empresa, parece que esta predicción se alinea con la visión estratégica de Amazon de Alexa para estar en todas partes.

Alexa es un programa de software basado en la nube que realiza los deberes de un asistente personal inteligente mientras la Línea de productos Echo de Amazon (Echo Dot, Echo Tap o Echo Show) es el hardware conectado a Internet (altavoz, micrófono y cámara) siempre escuchando la frase de activación del usuario conocida como la palabra de activación, cuando un usuario dice “Alexa, ¿qué hora es?”, la palabra de activación Alexa le indica al Echo que grabe y transmita el siguiente comando de voz a Alexa Voice Services. Usando una interfaz de programa de aplicación (API), AVS luego pasa los comandos de voz a Alexa, que utiliza reconocimiento de voz y del lenguaje natural para el procesamiento y formulación de las respuestas de Alexa.

Alexa, entonces, responde a los comandos del usuario con los altavoces integrados de Echo,” La hora es las 3:15 pm.” y envía una copia del comando de voz y la respuesta de Alexa a la aplicación Alexa ubicada en la nube de Amazon. Las copias de los comandos de audio del usuario y las respuestas de Alexa se conocen como tarjetas de respuesta. Se puede acceder a las tarjetas de respuesta a través de la aplicación Alexa del teléfono inteligente (Ford & Palmer, 2019).

AVS está integrado con el entorno de comercio electrónico de Amazon, lo que significa que hace que las compras sean rápidas y simples. El sistema puede funcionar como un centro de automatización del hogar, lo que permite al usuario poder controlar los sistemas de calefacción e iluminación. Por ejemplo, Alexa también se conecta con servicios de transmisión en línea y admite If This Then That (IFTTT), un servicio en línea que automatizar las tareas basadas en la web para que cuando los eventos ocurridos, las tareas de seguimiento se activan y gestionan.

En el entorno AVS, los servicios se conocen como habilidades. El Centro de Huracanes, por ejemplo, es una habilidad de Alexa que proporciona información constantemente actualizada sobre sistemas de tormentas, basado en datos de agencias gubernamentales. Una habilidad llamada virtual el bibliotecario es un motor de recomendaciones que sugiere libros, basado en nominaciones a premios, listas de bestsellers y reseñas de usuarios. Alexa Skills Kit, un kit de desarrollo de software (SDK, por sus siglas en inglés), está disponible gratuitamente para desarrolladores y las habilidades están disponibles para descarga instantánea desde Amazon.

3.3. Raspberry Pi

Raspberry Pi es una placa base de computadora de bajo costo; es una computadora pequeña, del tamaño de una tarjeta de crédito, desarrollada en el Reino Unido por la Fundación Raspberry Pi en 2011. Fue ideado para estimular la enseñanza de la informática en escuelas, aunque no comenzó su comercialización hasta 2012.

Raspberry Pi es un pequeño, potente, barato, hackeable y computadora de tablero orientado a la educación introducido en 2012. Funciona de la misma manera que una PC estándar, lo que requiere un teclado para entrada de comandos, una unidad de

visualización y una fuente de alimentación. Esta computadora del tamaño de una tarjeta de crédito con muchas actuaciones y asequible por \$25-\$35 es la plataforma perfecta para interfaz de muchos dispositivos. La gran mayoría de los componentes del sistema: su procesamiento central y unidad gráfica, hardware de audio y comunicaciones junto con 256 MB (Modelo A): se construyen chips de memoria de 512 MB (Modelo B) en un solo componente. La placa Raspberry Pi contiene lo esencial (procesador, chip de gráficos, memoria de programa - RAM) y otros dispositivos opcionales (varios interfaces y conectores para periféricos). El procesador de Raspberry Pi es un sistema en un chip de 32 bits y 700 MHz, que es construido en la arquitectura ARM11 y se puede overclockear para más poder. La memoria flash SD sirve como disco duro para Raspberry Pi.

La Raspberry Pi, como cualquier otra computadora, usa un sistema operativo. La opción de Linux llamada Raspbian es excelente compatible con Raspberry Pi porque es gratis y de código abierto, manteniendo el precio de la plataforma bajo y haciéndolo más hackeable. También hay algunas opciones de sistema operativo que no son de Linux disponible. Una de las mejores cosas de Raspberry Pi es que tiene una amplia gama de usos (Jain, Vaibhav, & Goyal, 2014).

Consiste en una placa que soporta varios componentes que necesita una computadora convencional y es capaz de comportarse como tal. En mayo de 2009, la Fundación Raspberry Pi fue fundada en Caldecote, South Cambridgeshire en Inglaterra como asociación caritativa regulada por la Comisión de caridad de Inglaterra y Gales. La idea es conseguir computadoras portátiles baratas que permiten que los niños las usen sin miedo, abriendo sus mentes y ayudándoles en la ética de “abrir y ver cómo funciona”.

El cofundador de la fundación es Eben Upton, un ex empleado de Broadcom, responsable del software y arquitectura de hardware de la Raspberry Pi. Eben Upton, contactó a un grupo de profesores, académicos e informáticos entusiastas de crear una computadora para alentar a los niños a aprender informática como lo hizo el microordenador Acorn BBC en 1981 (Historia de la Informática, 2018).

4. Arquitectura propuesta

A continuación, se presenta el esquema de arquitectura propuesto. La Figura 1 ilustra los elementos de software y hardware utilizados en este estudio. Amazon “echo” (Colon & Greenwald, 2015) es la plataforma física que admite el uso de Alexa y, por ende, AVS y Alexa Skill, los cuales registran las habilidades personalizadas del usuario. Esto permite la interacción con Raspberry Pi; la información de Alexa Skill se transporta de forma directa y segura a través de un ngrok o túnel virtual (VPN, por sus siglas en inglés), tan pronto como la información llega al Raspberry PI.

Ngrok es un software de proxy inverso de túnel multiplataforma que establece túneles seguros desde un punto final público como Internet a un servicio de red que se ejecuta localmente mientras captura todo el tráfico para una inspección y reproducción detalladas como se muestra en la Figura 2.

Antes de usar ngrok, cuando necesitábamos exponer una aplicación localhost a la web (internet), todo lo que estábamos haciendo era implementar la aplicación en un servidor que ejecutaba una DMZ o solíamos reubicar el host en DMZ y configurar NATing en el

formulado por cada usuario. Para esto se utiliza comandos de voz y la respuesta se procesa leyendo en ese momento los datos de los sensores y haciendo un requerimiento a la API (DarkSky), uno de estos puede ser los sensores de clima y temperatura. Con estos datos, una respuesta se procesa y se envía a través de un servidor JSON, el cual interpreta la respuesta con los datos necesarios para ser aceptado por el AVS según su estándar. Esta respuesta viaja de la misma manera que llegó, pasando a través de ngrok a AVS y se muestra a través de Alexa en Amazon Echo.

Vale la pena enfatizar que el uso del lenguaje natural permite interpretar los servicios a realizar y responder las solicitudes que el usuario formula disminuyendo la barrera entre los sistemas domóticos y entornos del usuario.

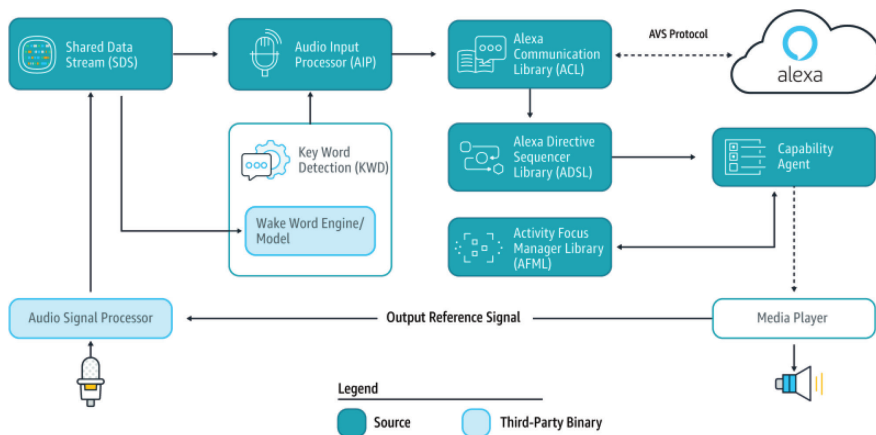


Figura 3 – Arquitectura AVS.

Alexa reside en la nube. El acceso es a través de la API de AVS. Alexa recibe solicitudes como secuencias de voz. Alexa analiza el discurso e identifica la habilidad solicitada. Se crea una representación estructurada de la habilidad y la intención.

Existen dos maneras, si el código de habilidad se implementa en AWS Lambda, ese servicio se activará dentro de AWS. Si se trata de un servicio web al que se accede a través de HTTPS, se realiza la llamada a la API REST (ver Figura 3). En cualquier caso, Alexa recibirá la respuesta como texto. Opcionalmente, la respuesta puede tener imágenes. Alexa convertirá el texto en voz, que se transmite al usuario a través de su dispositivo actual. Las imágenes se mostrarán donde el dispositivo tenga una pantalla.

Una habilidad se identifica mediante un nombre de invocación y se asigna a un servicio de fondo. Una habilidad puede ser diseñada para hacer múltiples cosas, cada una de las cuales se llama una intención. Cada intento puede identificarse mediante una o más palabras o frases, cada una de las cuales se denomina enunciado. Por ejemplo, puede crear una habilidad relacionada con el viaje que tenga tres propósitos: alquilar un automóvil, reservar un vuelo o reservar un hotel (Devopedia, 2018).

5. Desarrollo

El desarrollo de software comienza con la configuración de los componentes dentro de la Raspberry Pi; para esto, el uso de Alexa simplifica el montaje de parte de la infraestructura, por lo que tiene prioridad preparar el punto de llegada en base de habilidades personalizadas que vendrían hacer órdenes que luego serán respondidas por el Raspberry Pi. Para comenzar con el proceso de desarrollo se requiere una cuenta de Amazon para acceder al sitio de desarrolladores de Amazon donde lo primero que debes hacer es crear son las habilidades para Alexa.

5.1. Creando el modelo básico de Alexa Skill

Dentro de la sección de creación de habilidades (skills), se le da un nombre a dicha habilidad, esto debe crearse con una serie de palabras con las que Alexa podría invocar las habilidades creadas. Es importante recordar el nombre de la intención, ya que se utilizará para reconocer las diferentes tareas que Raspberry Pi puede reconocer.

El modelo de interacción es una parte muy importante de nuestra habilidad. Contiene el nombre de invocación de la habilidad, sus intents y expresiones, entre otros aspectos que son cruciales para la interacción con los servicios de habilidades. Los Intents incorporados tienen enunciados incorporados. Es posible extender el número de enunciados existentes agregando nuevos enunciados. Cuando se crea una habilidad, se agregan cuatro intents incorporados al modelo de interacción de forma predeterminada. Estos son: AMAZON.HelpIntent, AMAZON.CancelIntent, AMAZON.FallbackIntent y AMAZON.StopIntent.

El “AMAZON.NoIntent”, “AMAZON.CancelIntent” y “AMAZON.StopIntent” manejarán la solicitud de voz del usuario cuando el usuario decida interactuar con la habilidad, o quiera detener la interacción. El “Amazon.FallbackIntent” se encargará de la solicitud de voz del usuario que no coincide con ninguna de las declaraciones. El “AMAZON.HelpIntent” se encargará de la solicitud de ayuda de voz del usuario. El “AMAZON.NoIntent” no se incluye de manera predeterminada, como los otros intentos integrados, por lo que tendremos que agregarlo (Ralevic, 2018).

```
{
  "name": "raspTemperatureIntent",
  "slots": [],
  "samples": [
    "how hot it is",
    "how cold it is",
    "how is the weather",
    "what is the temperature",
    "look out for the temperature",
    "find me the temperature",
    "get me the temperature",
    "tell me the temperature",
    "give me the temperature"
  ]
}
```

Figura 4 – Intent en format JSON.

La Figura 4 muestra las diferentes formas de invocar las habilidades, esto con el objetivo de que sea más cómodo y flexible trabajar por parte del usuario. Amazon recomienda el uso de frases cortas para esto, porque el usuario debe invocar la habilidad de la manera más natural posible.

Las tragamonedas son un accesorio muy poderoso para construir una habilidad personalizada de Alexa. Por ejemplo, la declaración “how hot it is”. Además, las ranuras múltiples brindan la posibilidad de usar cuadros de diálogo en los que la habilidad Alexa solicita al usuario que complete todos los valores de las ranuras para cumplir con la intención.

5.2. Configuración y mapeo del servidor Ngrok

Ngrok es un túnel VPN que conecta dos puntos, uno de esos actúa como un servidor. Los datos van de punto a punto de forma segura, manteniendo la integridad de los datos a través del sistema propuesto. Esto es crucial porque los estándares y reglas que Amazon aplica para enviar y recibir datos de los AVS. La Figura 4 muestra el servidor ngrok configurado y corriendo; usa el puerto 5000, y debe ser el mismo puerto que el servidor JSON está escuchando.

Puede usar cualquier framework del lado del servidor para trabajar con ngrok. El único requisito aquí es asegurarse de que el servidor local se esté ejecutando correctamente. Para ello se debe colocar “http://6fd1bdc1.ngrok.io” o “https://6fd1bdc1.ngrok.io” (SSL) en la barra de direcciones del navegador y debería mostrar el sitio local. Estas son URL únicas que se crean cada vez que reinicia ngrok, lo que facilita compartirlas para sesiones de prueba únicas en un equipo (SitePoint, 2015).

5.3. Crear el script de Python para JSON

Un servidor que interpreta los datos en formato JSON desde Amazon debería estar configurado. Para este propósito, se ha creado un script en lenguaje Python que escuchó el mismo puerto que se asignó al ngrok para responder a las solicitudes hechas por Alexa. La Figura 5 muestra el código del servidor JSON. Como puede verse, está escuchando el puerto 5000, que es lo mismo que el servidor *ngrok*. Además, podemos resaltar el uso de la biblioteca de Amazon para el desarrollo de este tipo de sistemas.

5.4. Configurando la API de Dark Sky

El código vinculado a la cuenta y la ubicación desde la cual se quiere recopilar los datos deben especificarse para usar la API. La API podría devolver los datos sobre la velocidad del viento y la temperatura, así como los niveles de humedad, pero para el sistema propuesto, solo se usará la temperatura ambiente y un pronóstico por día. La Figura 6 ilustra cómo los pedidos de clima y temperatura podrían ser realizados por la API Dark Sky para ser usado en la respuesta enviada para el usuario.

5.5. Implementación de Infraestructura Física

Hasta ahora se han detallado las configuraciones relacionadas con el software del sistema, pero el hardware específico necesita ser habilitado para poder completar las

solicitudes hechas por el usuario y definidas por las habilidades personalizadas. Para esto se necesitan sensores de temperatura y humedad DHT22, que fueron adquiridos y probados con modelos de la misma serie para asegurar su correcto funcionamiento. Los sensores de humedad y temperatura deben estar conectados para leer los datos y enviarlos al AVS.

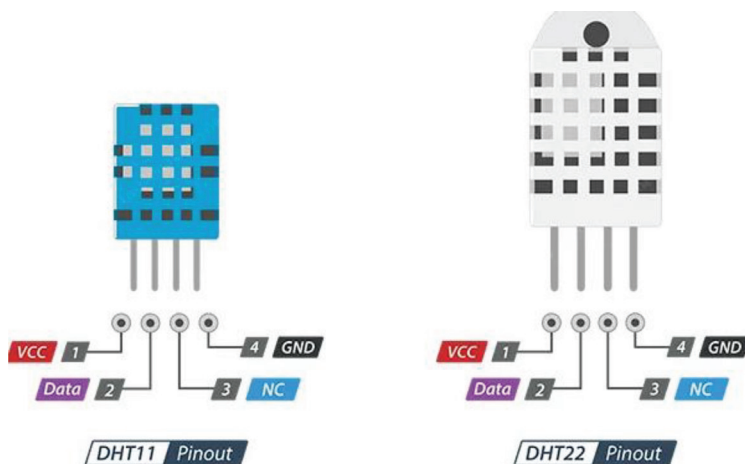


Figura 5 – Sensores DHT11 y DHT22.

En el mercado hay varios sensores de este tipo como se observa en la Figura 5. El sensor DHT11 trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos. Por otro lado, en el sensor DHT22 el rango de medición de temperatura es de -40°C a 80 °C con precisión de ± 0.5 °C y rango de humedad de 0 a 100% RH con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos. Ambas unidades usan un sensor de humedad capacitivo y un termistor para medir el aire circundante, y emiten una señal digital en el pin de datos (no se necesitan pines de entrada analógicos), para que un microcontrolador pueda leerlos fácilmente. Son básicos y lentos, pero excelentes para proyectos a pequeña escala que requieren un registro de datos básico (Wia Community, 2018).

Para garantizar la precisión de los datos recibidos, se probaron los sensores DHT11 y DHT22 por cinco días comparando la lectura de datos de cada sensor. Teniendo en cuenta los resultados obtenidos, el sensor DHT22 fue elegido para realizar el trabajo ya que ofrece un 99% de precisión. El sensor está conectado a la Raspberry Pi a través de sus pines de entrada y salida (GPIO), más específicamente a los pines 22 y 2 también especificados en el código que se observa en la Figura 6.

```

logging.getLogger("flask_ask").setLevel(logging.DEBUG)
@ask.intent(' raspTemperatureIntent')
def gpio_control():

    try:
        humidity, temperatureSensor = Adafruit_DHT.read(22, 2)
        print(temperatureSensor)
        location = forecast(key, "house,units="si")
        temperatureOutside = location.temperature
        forecaste = location['currently']['summary']
    except Exception as e:
        return statement('something went wrong with temperature')
    return statement('the temperature of the room is {:.2f} degrees, outside is {} degrees and {}'.format(temperatureSensor, temperatureOutside, forecaste))

@ask.intent(' forecastIntent', mapping={'day': 'day'})
def gpio_control(day):
    print(day)
    days = ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"]
    format_date='%Y-%m-%d'
    try:
        location = forecast(key, "house,units="si")
        datetime_obj = datetime.strptime(day, format_date)
        dayWeek=datetime_obj.weekday()
        dayName = days[dayWeek]
        sum = location.daily[dayWeek].summary
        tempMin = location.daily[dayWeek].temperatureMin
        tempMax = location.daily[dayWeek].temperatureMax
    except Exception as e:
        return statement('something went wrong with forecast')
    return statement('{}: {} Temperature range: {} - {}'.format(dayName, sum, tempMin, tempMax))

app.run(debug=False)

```

Figura 6 – Lectura e interpretación del código del servidor.

6. Conclusiones

El presente trabajo de investigación pone de manifiesto las bondades de las nuevas tecnologías aplicadas a temas de interés colectivos, tal es el caso del medio ambiente. Para esto se hace necesario la orquestación de varios servicios a través de una placa base como Raspberry Pi con otros elementos externos como DarkSky API y las habilidades (skills) que Alexa incorpora, siendo estos últimos los que mayor problema presentan ya que dependen del versionamiento, código fuente, entre otras características que impiden o permiten su integración con otro hardware.

La configuración de eventos (intentos, peticiones), son los que permiten una fácil interacción hombre – máquina, para esto se hace necesario combinar ciertas instrucciones (Python y DarkSkyAPI), he aquí donde se llegaron a evidenciar ciertas necesidades que garanticen la confiabilidad del prototipo, una de estas es la implementación de túneles (Ngrok).

El uso de un servicio de intérprete de lenguaje natural para realizar y responder a las solicitudes del usuario naturaliza y disminuye la barrera entre el sistema de domótica y el entorno del usuario. Como trabajo futuro, se plantea reemplazar el AVS por una herramienta más flexible que admita una variedad de idiomas.

Referencias

- Aravindan, V., & James, D. (2017). Smart Homes using Internet of Things. *International Research Journal of Engineering and Technology*, (pp. 1725-1729).
- Bugeja, J., Jacobsson, A., & Davidsson, P. (2018). Smart Connected Homes. In J. Bugeja, A. Jacobsson, & P. Davidsson, *Internet of Things A to Z: Technologies and Applications* (pp. 359-384). Wiley.
- Colon, A., & Greenwald, M. (2015). Amazon echo. *PC Magazine*.
- Devopedia. (2018). *Amazon Alexa*. Retrieved from <https://devopedia.org/amazon-alexa>
- Ford, M., & Palmer, W. (2019). Alexa, are you listening to me? An analysis of Alexa voice service network traffic. *Personal and Ubiquitous Computing*, 23(1), 67-79.
- González García, C., Meana Llorián, D., Pelayo García-Bustelo, B., & Cueva Lovelle, J. (2017). A review about smart objects, sensors, and actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- González García, C., Meana Llorián, D., Pelayo García-Bustelo, B., & Cueva Lovelle, J. (2017). A review about smart objects, sensors, and actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- Hayes, A. (2017). *Amazon Alexa: A Quick-Start Beginner's Guide*. Andy Hayes.
- Hipertextual. (2018). *Qué es y cómo funciona el Internet de las cosas?* Retrieved from <https://hipertextual.com/archivo/2014/10/internet-cosas>
- Historia de la Informática. (2018). *RASPBERRY PI*. Retrieved from <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi>
- Hoy, M. (2018). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical reference services quarterly*, 37(1), 81-88.
- Ibrahim, M., Elgamri, A., Babiker, S., & Mohamed, A. (2015). Internet of things based smart environmental monitoring using the Raspberry-Pi computer. *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, (pp. 159-164).
- Jain, S., Vaibhav, A., & Goyal, L. (2014). Raspberry Pi based interactive home automation system through E-mail. *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, (pp. 277-280).
- Jindarat, S., & Wuttidittachotti, P. (2015). Smart farm monitoring using Raspberry Pi and Arduino. *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, (pp. 284-288).
- Kim, S., Row, Y.-k., & Nam, T.-J. (2018). Thermal Interaction with a Voice-based Intelligent Agent. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, (p. LBW631).
- Ma, H., & Liu, W. (2017). A progressive search paradigm for the internet of things. *IEEE MultiMedia*, 25(1), 76-86.

- Mitrevski, M. (2018). Getting started with wit. ai. In M. Mitrevski, *Developing Conversational Interfaces for iOS* (pp. 143-164). Springer.
- Oleoshop. (2018). *Principales asistentes de voz y en qué se diferencian*. Retrieved from <https://www.oleoshop.com/blog/asistentes-de-voz>
- Polyakov, E., Mazhanov, M., Rolich, A., Voskov, L., Kachalova, M., & Polyakov, S. (2018). Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning. *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, (pp. 1-5).
- Pradhan, A., Mehta, K., & Findlater, L. (2018). “Accessibility Came by Accident” Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, (pp. 1-13).
- Putthapipat, P., Woralert, C., & Sirinimnuankul, P. (2018). Speech recognition gateway for home automation on open platform. *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, (pp. 1-4).
- Ralevic, U. (2018). *How To Build A Custom Amazon Alexa Skill, Step-By-Step: My Favorite Chess Player*. Retrieved from <https://medium.com/crowdbotics/how-to-build-a-custom-amazon-alexa-skill-step-by-step-my-favorite-chess-player-dccoedae53fb>
- Rao, P., & Uma, S. (2015). Raspberry Pi home automation with wireless sensors using smart phone. *International Journal of Computer Science and Mobile Computing*, 4(5), 797-803.
- Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delahoche, L., & Loge, C. (2006). The smart home concept: our immediate future. *2006 1st IEEE international conference on e-learning in industrial electronics*, (pp. 23-28).
- SitePoint. (2015). *How to Use ngrok to Test a Local Site*. Retrieved from <https://www.sitepoint.com/use-ngrok-test-local-site/>
- Upton, E., & Halfacree, G. (2012). *Meet the Raspberry Pi*. John Wiley & Sons.
- Vmoksha. (2020). *Open source technology - What is Ngrok?* Retrieved from <https://vmokshagroup.com/blog/expose-your-localhost-to-web-in-50-seconds-using-ngrok>
- Whatis. (2018). *Alexa Voice Services (AVS)*. Retrieved from <https://whatis.techtarget.com/definition/Alexa-Voice-Services-AVS>
- Wia Community. (2018). *DHT11 vs DHT22: What is the difference?* Retrieved from <https://community.wia.io/d/54-dht11-vs-dht22-what-s-the-difference>