

# Proyecto Kavita: Programación por voz para personas con discapacidades motoras

Christopher L. Ayala-Griffin, subgraduado<sup>1</sup>, Dayanlee de León, subgraduada<sup>1</sup> y Patricia Ordoñez-Franco, PhD<sup>1</sup>

<sup>1</sup>Universidad de Puerto de Puerto Rico Recinto de Río Piedras,

[christopher.ayala@upr.edu](mailto:christopher.ayala@upr.edu), [dayanlee.deleon@upr.edu](mailto:dayanlee.deleon@upr.edu), [patricia.ordonez@upr.edu](mailto:patricia.ordonez@upr.edu)

**Resumen**— La mayoría de los programas de computadora están pensados de una manera que requiere interacción basada en movimientos de dedos y gestos con las manos. Este tipo de entorno, que asume la destreza de las manos humanas, presenta limitaciones para quienes tienen discapacidades motoras. Esta limitación excluye a dicha población de aprender a codificar y, para aquellos que desarrollan algún trastorno musculoesquelético en etapas más tardías, podría poner en peligro sus carreras como programadores. El objetivo de esta investigación es diseñar un Voice User Interface (o VUI por sus siglas en inglés) que permita el uso de la voz del usuario para programar en un Integrated Development Environment (o IDE por sus siglas en inglés). Para ello, se definió una estructura de programación básica utilizando Alexa Skills, que permite al usuario declarar variables, imprimir valores, resolver expresiones matemáticas básicas, insertar expresiones condicionales, crear bucles y definir funciones. Se creó un editor de texto en línea usando CodeMirror para correr la entrada del usuario utilizando el lenguaje de programación Python. Aunque todavía no se pudo evaluar resultados puesto que la aplicación no tiene integrado un compilador. Para futuros trabajos se contempla añadir el compilador, y así poder ejecutar el programa del usuario en el editor en línea. Además, añadir la capacidad de poder editar, depurar y mover el cursor utilizando el Alexa Skill.

**Palabras Claves**— programación por voz, accesibilidad, diseño universal, lenguajes de programación, open-source

## I. INTRODUCCIÓN

Parte del proceso de diseño de interfaz de un programa, un sistema operativo o una computadora, es tomar en consideración al usuario durante todo el proceso de desarrollo del producto. En los últimos años han habido avances significativos en el desarrollo del diseño universal tomando en consideración a una base de usuarios más diversa y asegurando que los programas puedan ser más accesibles a diferentes usuarios sin discriminar por edad, habilidad o estatus social [1]. Sin embargo, a pesar de estos esfuerzos todavía existe una brecha que limita a usuarios discapacitados o con problemas motores. La programación requiere el uso continuo de las manos y se necesita mayor esfuerzo para que personas que no puedan utilizar sus manos puedan aprender a programar. Así permitiendo mayor acceso a las personas discapacitadas para integrarse en carreras dentro de las

ciencias de cómputos o como programadores. Una solución alternativa a este problema es utilizar programas de reconocimiento de voz (Speech Recognition System, SRS por sus siglas en inglés) para codificar. Como parte de los esfuerzos de aumentar la diversidad en las ciencias de cómputos, proponemos el desarrollo de herramientas de trabajo que utilicen el modelo de diseño universal, en este caso un sistema que permita programar por voz. El sistema de SRS utilizado en este proyecto fue Alexa, mediante el desarrollo de un Alexa Skill, que reconoce los comandos del usuario, guarda la información y luego de ser procesados los datos, le devuelve los resultados al usuario.

Existe una comunidad creciente de programadores interesados en el desarrollo de métodos para escribir código utilizando comandos de voz en lugar de escribir con el teclado. Por ejemplo, Travis Rudd, un programador profesional, explicó cómo escribe código dictando en lugar de usar el teclado. Usó Dragon Naturally Speaking y una extensión de voz de Python, y afirmó que era más eficiente que con la forma tradicional de escribir en el teclado [2]. Implementó una gramática que usaba sonidos cortos y palabras mundanas como entrada, pero esta gramática representaría una lucha para las personas que no son programadores y necesitan o quieren aprender a codificar inicialmente con la voz. Spoken Java es un sistema de lenguaje diseñado para hablar Java de forma natural, basado en el documento de investigación de Andrew Begel, "Una evaluación de un entorno de programación basado en el habla" [3]. Se han realizado distintos enfoques, como por ejemplo, en el escrito "Programación por voz: un enfoque de manos libres para niños con problemas de motor" cuando se apunta a un grupo demográfico más joven a través de la plataforma Scratch que combina la programación de voz con su interfaz gráfica de usuario [4]. En "Programación de asistencia para discapacitados: uso de entrada de voz para escribir código", también se creó una gramática personalizada para el software de reconocimiento de voz Sphinx 4, junto con un analizador

para mejorar la precisión del sistema y producir código en Java [5].

## II. TRASFONDO

La programación por voz cumple con la tarea de diseño universal, y aunque requiere de entrenamiento para dominar su uso efectivamente, tiene el potencial de facilitar el acceso a personas con discapacidad motora. Vale considerar que resulta más natural para los humanos la comunicación oral que la escrita, por lo que, esta alternativa de programar por voz pudiera ser adoptada por otros programadores como alternativa a tener que escribir código. Sin embargo, como cualquier otro lenguaje, la programación por voz requiere del diseño de una sintaxis que resulte natural y simple para el programador. Uno de los mayores retos es el hecho que cada lenguaje de programación tiene sus propias reglas de sintaxis y no siguen la misma estructura que el lenguaje hablado. Esto hace que sea difícil de comprender por los SRS actuales cuando se intenta su aplicación en un IDE. Algunas preguntas que intentamos contestar son, ¿cómo se define un lenguaje para codificar por voz?, ¿cómo se diseña de manera que pueda obedecer las reglas de otros lenguajes?, ¿cómo hago para mover el cursor?, entre otras interrogantes que al final pudieran tener un efecto importante en la experiencia del usuario y la aceptación de este modelo alternativo para programar. En este proyecto se propone el uso de gerundios para escribir palabras clave y diferenciar claramente los comandos de voz destinados a generar una estructura de programación estándar de los destinados al dictado. Además, después de la primera etapa de investigación, llegamos a la conclusión de que los gerundios son distinguidos con mayor efectividad por el programa de reconocimiento de voz.

Luego del desarrollo de un lenguaje para programar por voz que puede ser utilizado por programadores principiantes hasta los más avanzados, nos proponemos integrarlo a un IDE. En este proyecto de investigación se desarrollará un Alexa Skill para implementar el vocabulario, puesto que Alexa cuenta ya con capacidades para entender el lenguaje natural. El sistema se desarrollará usando las herramientas provistas por Amazon Web Services (AWS), ya que permiten mayor flexibilidad para ser integrado a otros servicios. El IDE será en línea e integrado en un webapp. Se utilizará el editor de CodeMirror puesto que su código fuente corre en JavaScript y puede ser integrado al webapp. Este editor es muy versátil, tiene la posibilidad de ser utilizado con diferentes lenguajes y sus componentes y funcionalidades pueden ser editadas. Sin embargo, en esta etapa del proyecto,

el usuario podrá solo hacer programas en el lenguaje de programación de Python. CodeMirror es un programa abierto, compartido bajo la licencia de MIT. Esto implica que podremos contar con un proyecto de código abierto que será accesible para todos, permitir que sea compartido y promover la colaboración de la comunidad.

## III. METODOLOGÍA

La implementación del vocabulario se desarrolla con el Alexa Skill, así se saca ventaja de sus capacidades de reconocimiento de voz y entender el lenguaje natural. La implementación del Alexa Skill se realiza usando los recursos que provee AWS, en este caso el código que corre dentro de la función Lambda. Esto permite definir estructura de programación básica que permite al usuario declarar variables, imprimir valores, resolver expresiones matemáticas básicas, insertar expresiones condicionales, crear bucles y definir funciones. El Alexa Skill se divide en habilidades, intenciones y expresiones. Dentro de las habilidades se implementan las intenciones y las expresiones. Para invocar las habilidades, se le asigna un nombre de invocación. Las intenciones son acciones específicas que ocurrirán en función de las declaraciones. Las declaraciones son la forma en que el vocabulario se registra en la habilidad. Cada intento tiene su propio conjunto de enunciados y son únicos para cada uno, esto permite que Alexa encuentre la acción correcta que el usuario desea realizar.

Dentro de las expresiones hay valores específicos requeridos que Alexa debe completar. Esto da flexibilidad a los usuarios en la creación de tipos de variables, nombres de variables, asignar o cambiar valores, entre otras posibilidades donde cada usuario se le permita proveer una entrada distinta. Estas variables están fuera del vocabulario estándar. Un ejemplo de esto se puede ver en la figura 1.

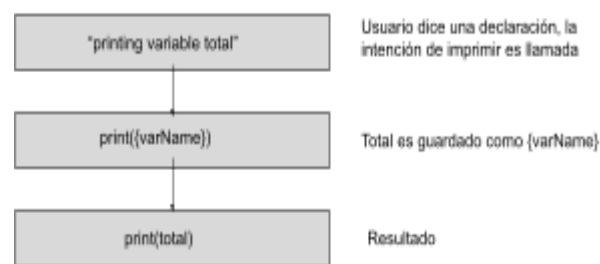


Fig. 1 Diagrama que demuestra como el nombre de la variable es captado y acomodado en la sintaxis del lenguaje de programación.

Para generar estructuras de decisión y repetición, se crearon dos habilidades independientes. El propósito de esto

es dividir las condiciones en enunciados más pequeños que se unirán al final, lo cual se puede observar en la figura 2. Una de las habilidades se ocupa de la inserción de símbolos matemáticos, mientras que la otra cubre los operadores lógicos y relacionales. Para comenzar una condición el usuario dice "inserting condition", posteriormente debe especificar si la comparación es con un número o una variable seguida del valor o el nombre. Después de especificar la forma en que se comparará el valor, al decir "ending condition" se colocarán los dos puntos necesarios al final y se preparará para el cuerpo del if o el ciclo.

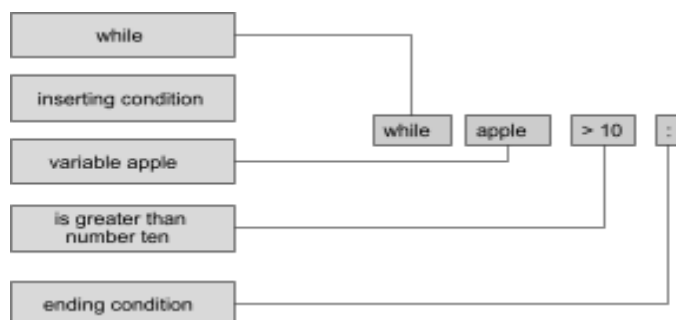


Fig. 2 Diagrama demostrando que parte del código es generado con respecto a cada parte de la expresión

En la figura 3 se muestra el diagrama de flujo de datos del sistema desarrollado. Esto incluye el Alexa Skill principal que procesa la información recibida por el usuario, guarda los resultados en la base de datos y le devuelve una respuesta al usuario. Por otro lado, está el webapp que contiene el editor en línea de CodeMirror y, además, recibe la información guardada en la bases de datos. En el JavaScript que realiza el GET se hace la solicitud al API de Alexa y la información solicitada se procesa en un segundo Alexa Skill que busca la información en la base de datos y devuelve la contestación al webapp.

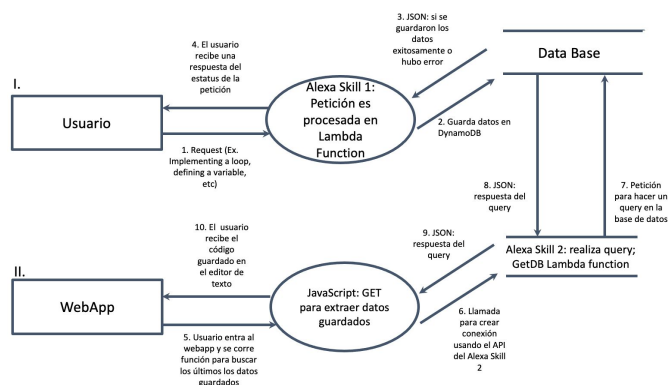


Fig. 3 Diagrama de Flujo de Datos. Hay dos sistemas principales, el sistema I representa la interacción del usuario con un dispositivo de Alexa y el sistema

II representa la interacción con el editor en línea integrado en el webapp. El flujo de los procesos están numerados, indicando el orden desde la petición inicial hasta cuando el usuario recibe los datos en el editor en línea.

#### IV. RESULTADOS

Al hacer las pruebas en el Amazon Developer Console se logró declarar variables, imprimir valores, resolver expresiones matemáticas básicas, insertar expresiones condicionales y crear ciclos exitosamente utilizando el Alexa Skill. Esto se realizó dentro del ambiente de pruebas de la consola de Alexa Skill, donde además permitió ver los resultados de la acción solicitada y corroborar si el resultado fue correcto. También se logró la conexión del editor en línea usando CodeMirror con el Alexa Skill, permitiendo que se recibiera la información guardada en la base de datos.

#### V. DISCUSIÓN

La consola para desarrolladores de Alexa Skill de Amazon resultó ser de gran utilidad para el desarrollo del VUI debido a sus capacidades de procesamiento de lenguaje natural. Además, el ambiente de desarrollo que ofrece la consola de Alexa permite suficiente flexibilidad para desarrollar un idioma de programación más natural para el usuario y así le resulte más fácil de aprender. Este idioma permite al usuario realizar acciones como declarar variables, imprimir valores, resolver expresiones matemáticas básicas, insertar expresiones condicionales y crear ciclos. Encontramos que Alexa fue capaz de elegir la acción deseada en función del tipo de solicitud y el orden en que se escriben los comandos, dando prioridad a la primera que se ha encontrado en los casos en que dos o más comandos son capaces de completar la misma tarea. Sin embargo, el sistema de Alexa no está diseñado para entender idiomas de programación. Esto obligó al diseño de comandos que puedan ser entendidos en el idioma inglés. Por un lado esto simplifica en gran medida el idioma de programación por voz para el usuario, pero por otro lado crea retos adicionales para interpretar los comandos y sean extraídos al editor usando la sintaxis de Python.

Hemos logrado definir un vocabulario y realizar operaciones básicas que devuelven resultado al usuario. Sin embargo, todavía queda trabajo por completar para tener un IDE donde se pueda programar por voz. La conexión de los componentes más importantes del sistema se logró con éxito, con excepción del compilador que no ha sido integrado. El sistema actual permite que el Alexa Skill reciba comandos del usuario los procese y le devuelva los resultados al usuario, y por otro lado, el editor en línea es capaz de obtener dichos resultados. Queda trabajo por realizar para tener un IDE que

se pueda controlar completamente por voz, pero los resultados que hemos obtenido del sistema actual han sido positivos para probar los conceptos aquí descritos. Futuros trabajos incluyen añadir el compilador, para así poder ejecutar el programa del usuario en el editor en línea, y añadir la capacidad de poder editar, depurar y mover el cursor utilizando el Alexa Skill. La meta es lograr un IDE que pueda correr Python usando solo la voz y ayudar en los esfuerzos para hacer la programación accesible a más personas.

#### RECONOCIMIENTOS

Este proyecto está inspirado en Kavita Krishnaswamy, estudiante graduada de la Universidad de Maryland, Baltimore. Kavita tiene lo que se conoce como atrofia muscular espinal, una condición genética poco común que la mantiene paralizada del cuello hasta las extremidades inferiores. A pesar que el diagnóstico es poco alentador, Kavita se niega a darse por vencida y con un solo dedo que puede mover para codificar, ha dedicado su tiempo y esfuerzo en diseñar tecnología que pueda ayudar a otros como ella a ser más independiente. Personas como Kavita son una inspiración y nos enseñan las grandes cosas que podemos lograr cuando perseveramos. Agradecemos también al Dr. Keith Vertanen de Michigan Technological University en Houghton, Michigan, por su mentoría y brindar apoyo a este proyecto.

#### REFERENCES

- [1] "Ronald L. Mace on NC State University, College of Design". Design.ncsu.edu  
[https://www.ncsu.edu/ncsu/design/cud/about\\_us/usronmace.htm](https://www.ncsu.edu/ncsu/design/cud/about_us/usronmace.htm)

- Retrieved 2020-02-16.
- [2] Rudd, T "Using Python to Code by Voice". Pycon US 2013. Retrieved April 25, 2015 from  
<http://pyvideo.org/video/1735/using-python-to-code-by-voice>
- [3] Begel, A., Graham, S.L. "An Assessment of a Speech-Based Programming Environment," Visual Languages and Human-Centric Computing, 2006. VL/HCC 2006, pp.116,120, 4-8 Sept. 2006.
- [4] 3. Wagner, A., Rudraraju, R., Datla, S., Banerjee, A., Sudame, M., & Gray, J. (2012). Programming by voice: A hands-free approach for motorically challenged children. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*(pp. 2087-2092).
- [5] Corbly, James E. "The Free Software Alternative: Freeware, Open-Source Software, And Libraries." Information Technology & Libraries 33.3 (2014): 65-75. Academic Search Complete. Web. 28 Feb. 2015.
- [6] Rosenblatt, L., Carrington, P., Hara, K., & Bigham, J. P. (2018). Vocal programming for people with upper-body motor impairments. In *Proceedings of the Internet of Accessible Things* (pp. 1-10).
- [7] Lee, H., Fenwick Jr, J. B., Klima, R. E., McRae, A. A., & Vahlbusch, J. (2019). *Disability Assistive Programming: Using Voice Input to Write Code* (Doctoral dissertation, Appalachian State University).
- [8] Pitts, J. (2017, September 15). UMBC robotics student invents help for those with disabilities like hers. Retrieved March 5, 2021, from <https://www.baltimoresun.com/health/bs-hs-kavita-robotics-20170829-story.html>