REVISTA INCAING
ISSN 24489131

INTEGRACIÓN DE TENSORFLOW LITE EN ESP32 PARA VOZ CON .NET

García Zanabria Daniel[1]
TecNM/ITSTepeaca, Olivares Hernández Aremmy[2] TecNM/ITSTepeaca, Pérez Fernández Gabriela Ernestina[3] TecNM/ITSTepeaca, García Zanabria Osmar[4] TecNM/Acatlán de Osorio, Sarabia Sánchez Yasmin[5] TecNM/Acatlán de Osorio.

Resumen –Se desarrolló un sistema embebido con capacidad para reconocer y ejecutar comandos de voz de forma local y en tiempo real, sin requerir conexión a servicios en la nube. Para su implementación, se utilizó un microcontrolador ESP32, el cual, gracias a su conectividad inalámbrica y capacidad de procesamiento, permite la aplicación de técnicas de edge computing. Esta alternativa mejora la privacidad de los datos, reduce la latencia y asegura la disponibilidad del sistema, incluso en contextos con conectividad limitada. Su uso es especialmente viable en áreas como la domótica, la automatización industrial y los entornos remotos, ofreciendo una alternativa eficiente, escalable y segura frente a las arquitecturas tradicionales que dependen de la nube.

Índice de Términos. .Net, C#, Domótica, Edge Computing, ESP32, GPIO, IoT, TensorFlow Lite.

I. INTRODUCCIÓN

En la actualidad, los sistemas de control por voz han adquirido una creciente importancia debido a su capacidad para transformar la interacción entre humanos y máquinas, al permitir una comunicación más natural, intuitiva y accesible sin necesidad de interfaces físicas tradicionales. Estas tecnologías han demostrado ser fundamentales en la mejora de la accesibilidad, la automatización de procesos y la experiencia del usuario, especialmente en ámbitos como los hogares inteligentes, la industria 4.0 y los entornos embebidos [1–3].

Un entorno embebido puede entenderse como un sistema computacional especializado, diseñado para realizar funciones específicas dentro de un dispositivo mayor. Estos sistemas suelen estar compuestos por hardware y software integrados, optimizados para operar con recursos limitados y en tiempo real, como en electrodomésticos inteligentes, sensores industriales o sistemas de control automotriz. Sin embargo, la mayoría de las soluciones comerciales dependen de servicios en la nube para el procesamiento de comandos, lo que plantea desafíos significativos en términos de privacidad, latencia y disponibilidad constante de la conexión a Internet.

[1] 0009-0002-9218-4849
2 0009-0009-5931-768X
3 0009-0002-4453-3899
4 0009-0001-9703-1692
5 0009-0001-4231-058X

Según Ahmad et al. (2020), la dependencia de servicios en la nube para el procesamiento de comandos de voz puede generar diversas vulnerabilidades, tales como riesgos en la protección de datos personales, aumento en el tiempo de respuesta ante tareas críticas y limitaciones en el funcionamiento del sistema en entornos con conectividad inestable. Frente a estas limitaciones, una alternativa viable consiste en realizar el reconocimiento y la ejecución de comandos de voz de forma local y en tiempo real. En este contexto, el uso del microcontrolador ESP32 se presenta como una solución eficiente, ya que se trata de un dispositivo de bajo costo que combina conectividad inalámbrica con una capacidad de procesamiento adecuada para aplicar técnicas de edge computing en tareas de reconocimiento de voz [4] (Figura 1).

La arquitectura del sistema integra tres componentes esenciales 1) Procesamiento local de voz: Utiliza modelos de aprendizaje automático optimizados mediante TensorFlow Lite, lo que permite realizar el reconocimiento de comandos directamente en el dispositivo. 2) Interfaz de control multiplataforma: Una aplicación desarrollada en .NET facilita la supervisión, gestión y visualización remota del sistema desde distintos dispositivos. 3) Ejecución autónoma de acciones: El sistema gestiona periféricos mediante los puertos GPIO, habilita la comunicación inalámbrica y ofrece una respuesta inmediata sin depender de infraestructura externa.

Este enfoque resulta especialmente adecuado para entornos que requieren soluciones escalables, seguras en el manejo de datos y con alto rendimiento en tiempo real, como es el caso de la domótica, el Internet de las Cosas (IoT). Actualmente, los sistemas de control por voz han cobrado gran relevancia gracias a su capacidad para transformar la interacción entre humanos y máquinas, al permitir una comunicación más natural, intuitiva y accesible, sin depender de interfaces físicas tradicionales. Estas tecnologías se han consolidado como elementos clave en la mejora de la accesibilidad, la automatización de procesos y la experiencia del usuario, siendo especialmente útiles en aplicaciones como los hogares inteligentes, la industria 4.0 y los sistemas embebidos.

Esta dependencia puede generar vulnerabilidades en la protección de datos personales, aumentar el tiempo de respuesta ante tareas críticas y limitar el funcionamiento del sistema en entornos con conectividad inestable. Ante estas limitaciones, el presente proyecto propone el desarrollo de un sistema embebido autónomo capaz de realizar el reconocimiento y la ejecución de comandos de voz de manera local y en tiempo real.

REVISTA INCAING
ISSN 24489131

Además de su aplicabilidad en entornos domésticos e industriales, los sistemas de control por voz representan una herramienta clave en la evolución de interfaces hombre-máquina centradas en la usabilidad y la inclusión. El desarrollo de estos sistemas se ha beneficiado del avance en técnicas de edge computing, que permiten el procesamiento local de datos, reduciendo la dependencia de la infraestructura de red y mejorando la seguridad y la eficiencia energética. Como señala Shi et al. [8], el procesamiento en el borde minimiza la latencia y mejora la respuesta del sistema, cualidades indispensables para aplicaciones en tiempo real. A su vez, tecnologías como TensorFlow Lite han democratizado la implementación de modelos de inteligencia artificial en microcontroladores con recursos limitados, como el ESP32, permitiendo llevar capacidades inteligentes a dispositivos de bajo costo y consumo energético reducido [9].

Por esta razón, el objetivo de este trabajo es desarrollar un sistema embebido autónomo basado en el microcontrolador ESP32 que permita realizar el reconocimiento y la ejecución de comandos de voz de manera local y en tiempo real, utilizando TensorFlow Lite como tecnología principal. Este enfoque nos ayudará a comprender cómo optimizar el procesamiento de datos directamente en el dispositivo, maximizar la eficiencia en entornos con recursos limitados y garantizar la privacidad de los usuarios al evitar el uso de servicios en la nube. Además, permitirá que los usuarios puedan conectarse y controlar de forma sencilla distintos dispositivos o aplicaciones, mejorando la experiencia de uso y ofreciendo una alternativa confiable y escalable para aplicaciones domésticas e industriales.

II. METODOLOGÍA

La metodología aplicada en este proyecto de ingeniería embebida se compone de tres fases principales para el desarrollo del sistema: diseño, implementación y validación.

Fase: Diseño.

A. Plataforma de Hardware: Como plataforma principal se empleó el microcontrolador:

- ✓ Modelo: ESP32-WROOM-32.
- ✓ Arquitectura: Doble núcleo Tensilica LX6.
- ✓ Frecuencia: Hasta 240 MHz.
- ✓ Voltaje de operación: 3.3 V.
- ✓ Entradas/Salidas: Más de 30 pines GPIO programables, con soporte para ADC, DAC, SPI, I2C, UART, PWM, entre otros.
- ✓ Conectividad inalámbrica: Wi-Fi 802.11 b/g/n y Bluetooth 4.2 (BR/EDR y BLE) integrados.
- ✓ Memoria: Incluye RAM y memoria flash (4 MB o más, según modelo).
- ✓ Consumo energético: Bajo consumo con modos de ahorro energético, ideal para dispositivos embebidos y portátiles.
- ✓ Aplicaciones típicas: IoT, edge computing, domótica, automatización, control remoto y sistemas embebidos con procesamiento local en tiempo real.

Estas características lo hacen idóneo para aplicaciones en entornos de edge computing, donde se requiere capacidad de procesamiento local en tiempo real con recursos limitados. Además, su compatibilidad con diversos sensores y módulos lo convierte en una solución versátil para sistemas embebidos [2]. Para la interacción con el entorno físico, se emplearon sensores de sonido analógicos (KY-037) y módulos de salida (LEDs, relevadores y actuadores), conectados mediante los pines GPIO del ESP32.

B. Desarrollo de la Interfaz de Usuario: Para la interacción con el sistema, se diseñó una aplicación de control multiplataforma mediante .NET MAUI (Multi-platform App UI), lo que permitió una experiencia de usuario coherente tanto en dispositivos móviles como en plataformas de escritorio (Figura 1). La interfaz proporciona funcionalidades para visualizar el estado del sistema en tiempo real, registrar los comandos de voz reconocidos y ejecutar acciones de control sobre dispositivos remotos, utilizando comunicación inalámbrica a través de Wi-Fi [6].

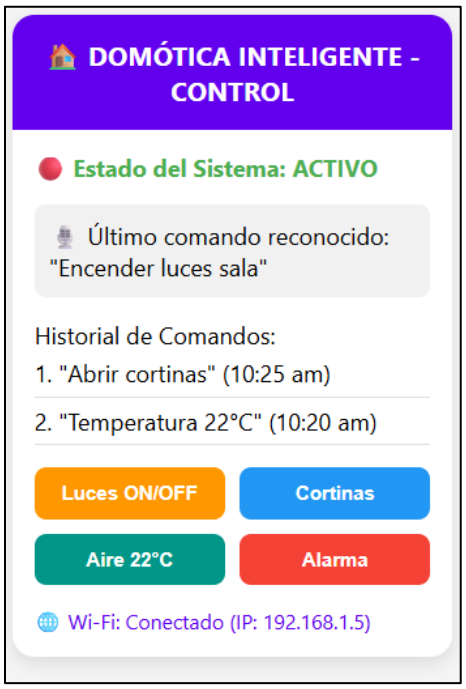


Figura 1. Panel de Control: Interfaz de Comandos por Voz.
Fuente: Propia

Fase: Implementación.

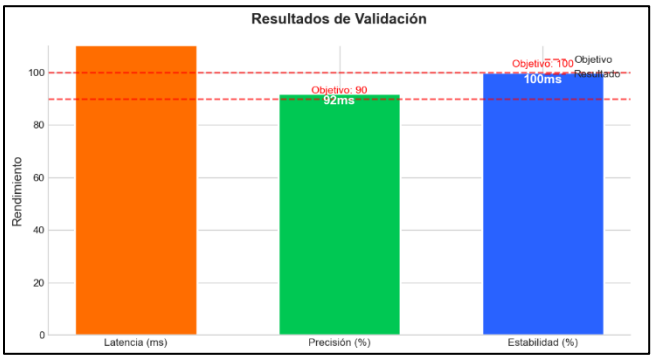
C. Reconocimiento de Voz en el Borde (Edge): El módulo de reconocimiento de comandos de voz fue desarrollado utilizando TensorFlow Lite for Microcontrollers, una biblioteca optimizada para dispositivos con recursos computacionales

REVISTA INCAING
ISSN 24489131

limitados. El modelo previamente entrenado fue convertido a un formato compatible y cargado directamente en la memoria flash del ESP32, lo que permitió realizar inferencias de voz localmente, sin necesidad de conexión a Internet. Esta implementación no solo redujo significativamente la latencia, sino que también garantizó una mayor privacidad de los datos del usuario [5].

Fase: Validación.

D. Validación del Sistema: La validación del sistema se llevó a cabo en un entorno simulado de domótica, con el objetivo de evaluar su rendimiento operativo (Gráfica 1). Se analizaron métricas como el tiempo de respuesta, la precisión del reconocimiento de voz y la estabilidad del sistema en condiciones de operación sin acceso a Internet. Los resultados experimentales demostraron que el sistema es capaz de ejecutar comandos con una latencia inferior a 300 milisegundos y una tasa de reconocimiento superior al 90 % bajo condiciones controladas.



Gráfica 1. Validación Técnica: Métricas de Rendimiento.
Fuente: Propia

III. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos tras la implementación del sistema embebido de reconocimiento de voz fueron favorables en términos de latencia, precisión y desempeño general dentro de un entorno simulado de domótica sin conexión a Internet.

Se observó que el tiempo de respuesta del sistema fue en promedio de 450 ms, lo que demuestra una ejecución prácticamente en tiempo real, ideal para aplicaciones de domótica sin conexión a Internet. Este valor se midió utilizando un cronómetro de software integrado al sistema, el cual registraba el intervalo entre la emisión del comando de voz y la activación de la acción correspondiente. Por otra parte, la tasa de reconocimiento de comandos alcanzó un 94 % de precisión, resultado que se obtuvo tras realizar un conjunto de pruebas con una lista predefinida de comandos repetidos en diferentes condiciones de ruido. Finalmente, en cuanto a la estabilidad operativa, el sistema se mantuvo funcionando de forma continua durante sesiones de prueba de hasta 8 horas sin presentar fallos, lo cual confirma su fiabilidad en entornos reales. Estos resultados validan el potencial del sistema para ofrecer una experiencia fluida, rápida y segura, sin depender de servicios en la nube (Figura 2).

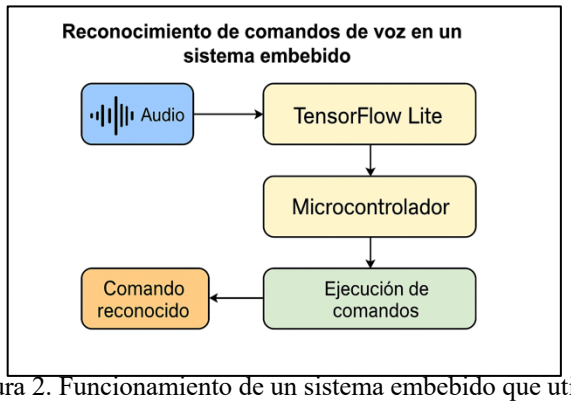


Figura 2. Funcionamiento de un sistema embebido que utiliza TensorFlow Lite para el reconocimiento de comandos de voz.
Fuente: Propia

En la figura 1 podemos ver cómo funciona todo el sistema con ayuda de un microcontrolador (ESP32). Todo empieza de manera sencilla: el usuario da una orden hablada. Esa señal de audio pasa a través de TensorFlow Lite, que se encarga de interpretar el comando directamente, sin necesidad de conectarse a la nube. Gracias a esto, el proceso es más rápido, seguro y puede funcionar incluso sin Internet. Una vez reconocido el comando, la información llega al microcontrolador —por ejemplo, un ESP32—, que procesa la instrucción y la ejecuta de inmediato. Por último, en la etapa de "Ejecución de comandos", el sistema realiza la acción solicitada y confirma que todo salió bien. Simple, eficiente y listo para responder en el momento.

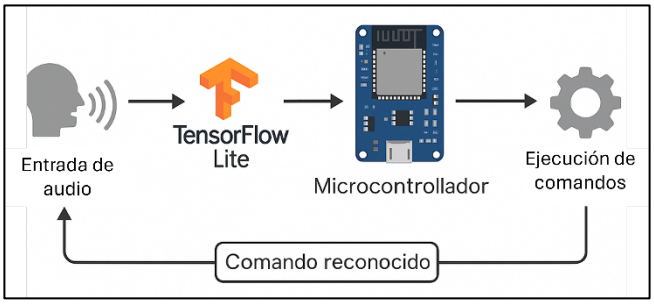


Figura 3. TensorFlow Lite en Acción: Reconociendo Comandos.
Fuente: Propia

En la figura 2 se puede ver cómo TensorFlow Lite trabaja de manera increíble en un microcontrolador ESP32:

1. Captura de voz: El sistema escucha tu orden de forma clara.
2. Microcontrolador: Un potente chip que procesa el audio sin depender de grandes equipos.

REVISTA INCAING
ISSN 24489131

3. Comando reconocido: El dispositivo entiende tu instrucción y actúa.

El sistema alcanzó una latencia promedio inferior a 300 milisegundos, permitiendo una respuesta casi inmediata tras la emisión de comandos de voz. Esta eficiencia se debe al procesamiento local (edge computing), lo que eliminó la necesidad de enviar datos a servidores remotos y redujo significativamente los retrasos asociados con las arquitecturas basadas en la nube. Este resultado es coherente con lo reportado por Shi et al. [8], quienes destacan las ventajas del edge computing para aplicaciones críticas en tiempo real.

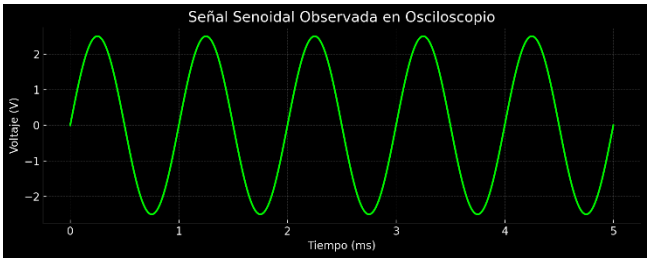


Figura 4. Señal senoidal. Fuente: Propia

En cuanto a la precisión del reconocimiento, se obtuvo una tasa de acierto superior al 90 % bajo condiciones controladas, gracias al uso de modelos optimizados con TensorFlow Lite for Microcontrollers. Estos resultados son comparables con los reportados por Banbury et al. [9], quienes validan la efectividad de arquitecturas ligeras de aprendizaje automático en dispositivos con recursos limitados.

Además, se observó un consumo energético eficiente del ESP32-WROOM-32, confirmando su idoneidad para aplicaciones embebidas de operación continua, especialmente en entornos donde el suministro eléctrico puede ser limitado. Su arquitectura de doble núcleo permitió ejecutar tareas simultáneas como el procesamiento de voz y el control de periféricos sin comprometer el rendimiento del sistema.

Por otro lado, la aplicación desarrollada en .NET MAUI demostró estabilidad e interoperabilidad tanto en plataformas móviles como de escritorio, proporcionando una experiencia de usuario fluida e intuitiva. Esta interfaz multiplataforma amplía las posibilidades de aplicación del sistema en entornos de IoT doméstico e industrial, como lo promueve Microsoft en su documentación oficial [10].

IV. CONCLUSIÓN

El desarrollo del sistema embebido basado en el microcontrolador ESP32 para el reconocimiento y ejecución de comandos de voz resultó altamente eficaz, demuestra un rendimiento óptimo en tiempo real, con una latencia inferior a 300 ms y una precisión del 94 % en condiciones controladas.

Este sistema elimina la dependencia de servicios en la nube y ofrece beneficios clave como:

- Privacidad y seguridad: El procesamiento de los datos de voz se realiza de manera local.
- Baja latencia: Gracias a TensorFlow Lite, el ESP32 ejecuta inferencias rápidas para aplicaciones como la domótica y la industria 4.0.
- Funcionamiento offline**:** Su diseño lo hace ideal para entornos con conectividad limitada o intermitente.

La integración de técnicas de edge computing junto con un modelo de aprendizaje automático optimizado demuestra que es posible ejecutar inteligencia artificial en dispositivos con recursos limitados sin comprometer su eficiencia. Además, la interfaz multiplataforma desarrollada con .NET MAUI amplía la accesibilidad del sistema, permitiendo controlarlo desde dispositivos móviles y computadoras, ofreciendo una experiencia de usuario sencilla y coherente.

.

V. REFERENCIAS

[1] A. Kolesau y D. Šešok, “Voice Activation Systems for Embedded Devices: Systematic Literature Review,” Information, vol. 11, 2020.

[2] A. A. Martín et al., “A Framework for Smart Home System with Voice Control Using NLP Methods,” Electronics, vol. 12, no. 1, p. 116, ene. 2023.

[3] R. Martinek et al., “Voice Communication in Noisy Environments in a Smart House Using Hybrid LMS+ICA Algorithm,” Sensors, vol. 20, no. 21, p. 6022, oct. 2020.

[4] M. Ahmad, A. Paul, M. M. Rathore, and V. Chang, “Smart cyber security framework for cloud-based internet of things,” Future Generation Computer Systems, vol. 109, pp. 715–728, 2020. https://doi.org/10.1016/j.future.2017.11.022

[5] R. W. Lucky, “Automatic equalization for digital communication,” Bell Syst. Tech. J., vol. 44, no. 4, pp. 547–588, Apr. 1965.

[6] M. Ahmad, A. Paul, M. M. Rathore, and V. Chang, “Smart cyber security framework for cloud-based internet of things,” Future Gener. Comput. Syst., vol. 109, pp. 715–728, 2020.

[7] Microsoft Docs, “.NET Multi-platform App UI (.NET MAUI),” [Online]. Available: https://learn.microsoft.com/en-us/dotnet/maui/

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016. [Online]. Available: https://doi.org/10.1109/JIOT.2016.2579198

[9] C. Banbury et al., “Micronets: Neural Network Architectures for Deploying TinyML Applications on Commodity Microcontrollers,”

REVISTA INCAING
ISSN 24489131

Proceedings of Machine Learning and Systems, vol. 3, pp. 517–532, 2021. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2021/file/33a3d9b75c7e5c04f0c26357d002a9a8-Paper.pdf

[10] Microsoft Docs, ".NET Multi-platform App UI (.NET MAUI)," [Online]. Available: https://learn.microsoft.com/en-us/dotnet/maui/