Interacción Persona-Máquina

U6: Interfaces basadas en reconocimiento de voz y sonido (I)

Javier Rodríguez Juan
j.rodriguezjuan@ua.es
José García Rodríguez
jgr@ua.es

Universidad de Alicante

2025-2026

Grado en Ingeniería en Inteligencia Artificial

Universitat d'Alacant
Universidad de Alicante

3DPERCEPTIONLAB

dtic

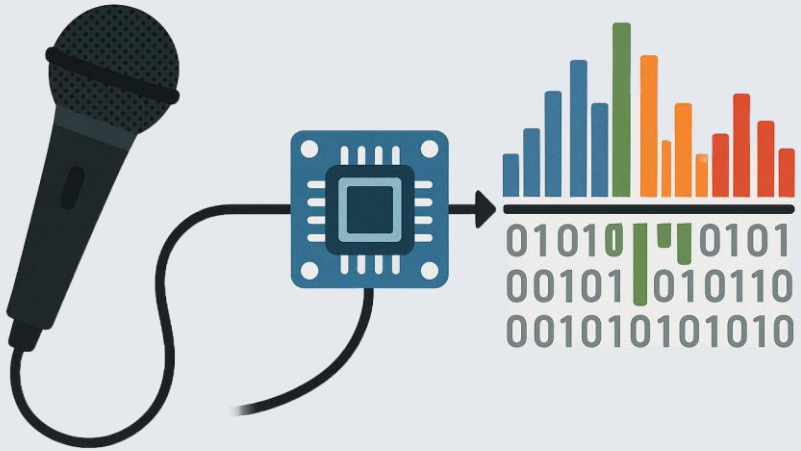
Contenidos

- Micrófonos
- Preprocesamiento
- Asistentes de voz
- Reconocimiento de sonidos ambientales

Micrófonos

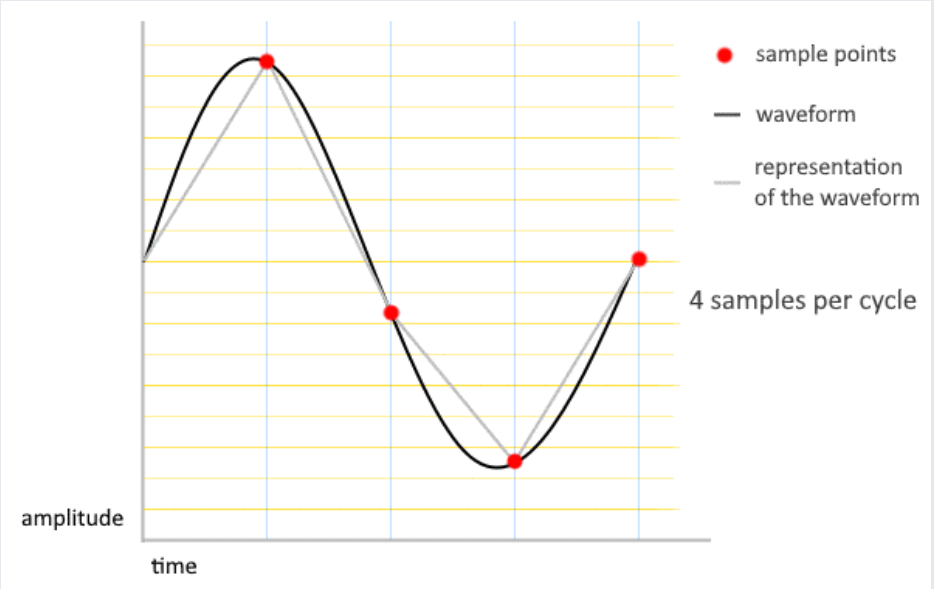
Micrófonos: Introducción

- Un micrófono es un transductor que convierte las variaciones de presión del sonido (ondas sonoras) en señales eléctricas. Estas señales luego se digitalizan mediante una tarjeta de sonido. Así se obtiene una representación numérica de la onda.



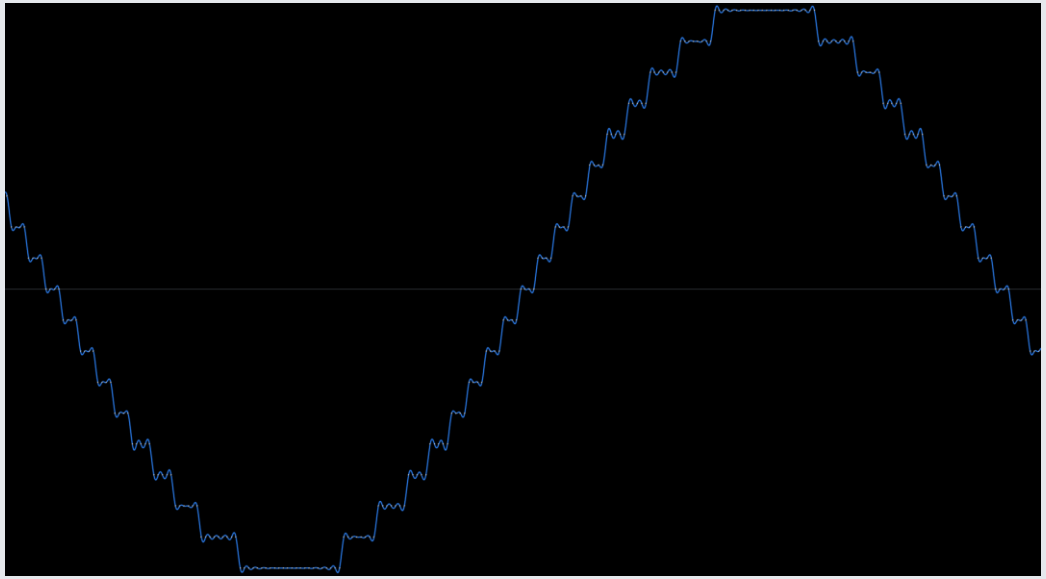
Micrófonos: Características I

- **Frecuencia de muestreo.** Son las muestras (“fotografías”) por segundo tomadas. Cuanto mayor la frecuencia de muestreo, más detalles del sonido se capturan. *16 kHz es suficiente para la voz y con 44.1 kHz se incluye todo el espectro de sonidos.*



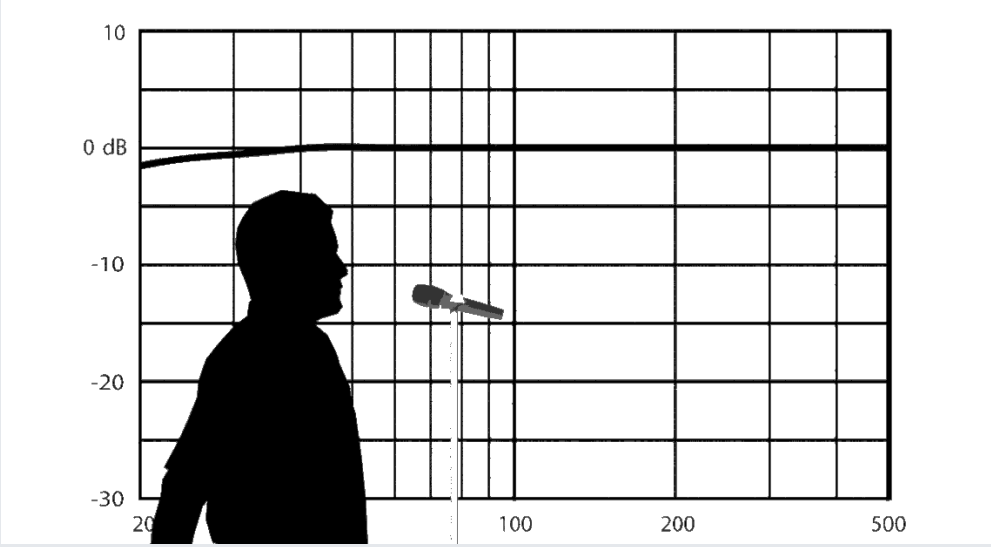
Micrófonos: Características II

- **Resolución en bits.** Es la precisión con la que se representa (codifica) cada muestra de audio, medida en bits. A más bits, mejor calidad y, menos ruido en rangos de voz elevados. Una gran cantidad de bits también crea archivos de audios más pesados. *Los CD utilizan una resolución de 16 bits.*



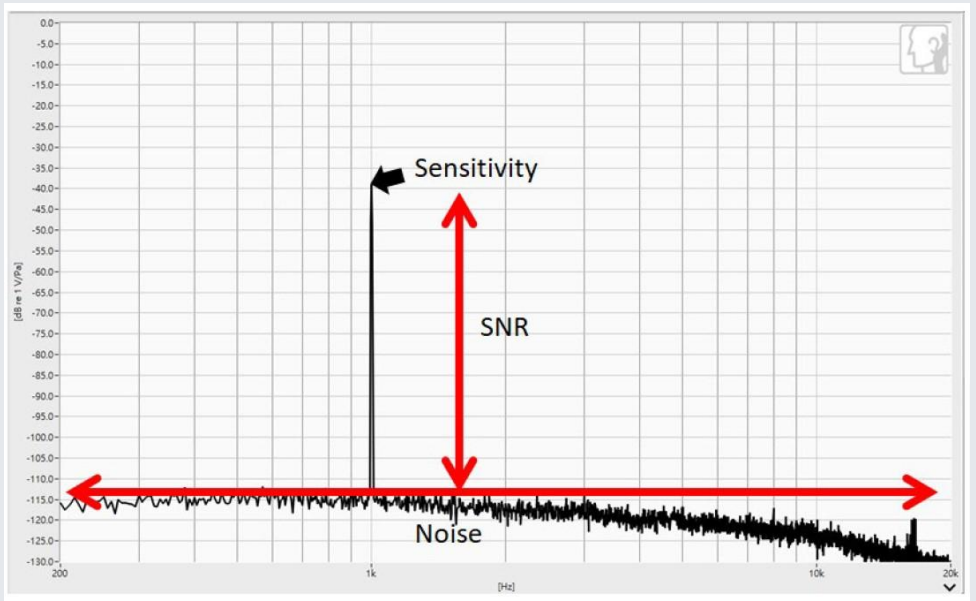
Micrófonos: Características III

- **Sensibilidad.** Es la cantidad de señal que genera el micrófono ante un sonido. Ayuda a determinar si un micro es adecuado para voces cercanas, entornos silenciosos o grabaciones a distancia. En asistentes de voz, la sensibilidad debe compensar la distancia del usuario. *Un micro muy sensible detectará voces suaves con claridad, pero también captará más ruido de fondo.*



Micrófonos: Características IV

- **Relación señal-ruido (SNR).** Es la diferencia entre el volumen de la señal útil (voz) y el ruido que añade el propio micrófono, medida en decibelios (dB). Cuanto más alto sea el SNR, más claro será el audio. *Un SNR de 70 dB significa que la voz está 70 dB por encima del ruido de fondo generado por el micro.*



Micrófonos: Tipos

De array (Campo lejano)

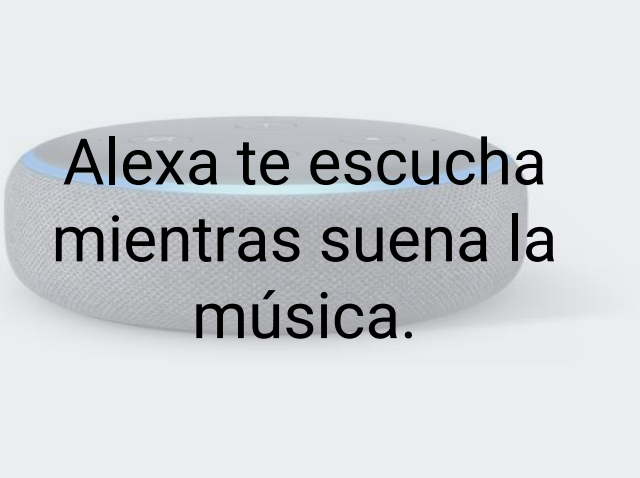
Definición

Varios micrófonos distribuidos por el dispositivo de forma circular/lineal.

Utilidad

Permiten detectar desde que dirección viene la voz y enfocarse en ella (beamforming).

Ejemplo



Micrófonos: Tipos

De diadema

Definición

Van integrados en unos auriculares, muy cerca de la boca del usuario

Utilidad

Muy robustos al ruido. Se usan en interacción dirigida por voz de alta precisión.

Ejemplo

Micrófonos: Tipos

Direccionales

Definición

Son alargados y focalizan el sonido en una dirección estrecha, rechazando el ruido lateral.

Utilidad

Se usan para grabar voces desde cierta distancia sin captar el ruido del entorno.

Ejemplo

Micrófonos: Tipos

MEMS (Micro-Electro-Mechanical Systems)

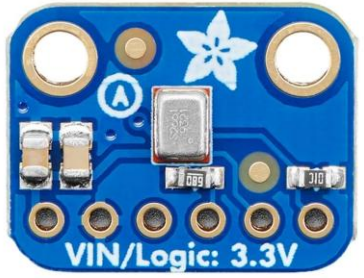
Definición

Micros en miniatura fabricados con semiconductores. Integran todo en un solo chip de silicio.

Utilidad

Permiten incorporar captura de voz o sonido ambiental en dispositivos compactos (móviles, wearables, etc).

Ejemplo



Preprocesamiento

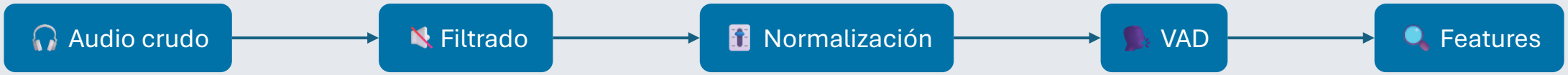
Preprocesamiento: Introducción

- Para que el audio capturado por un micrófono pueda ser usado por una arquitectura de IA, normalmente necesitaremos un paso previo dónde este audio se adapta al tipo de entrada esperada por la arquitectura. Este paso previo para preparar el audio es lo que llamamos **preprocesamiento** del audio.

$$audio = [x1, x2, x3, \dots]$$

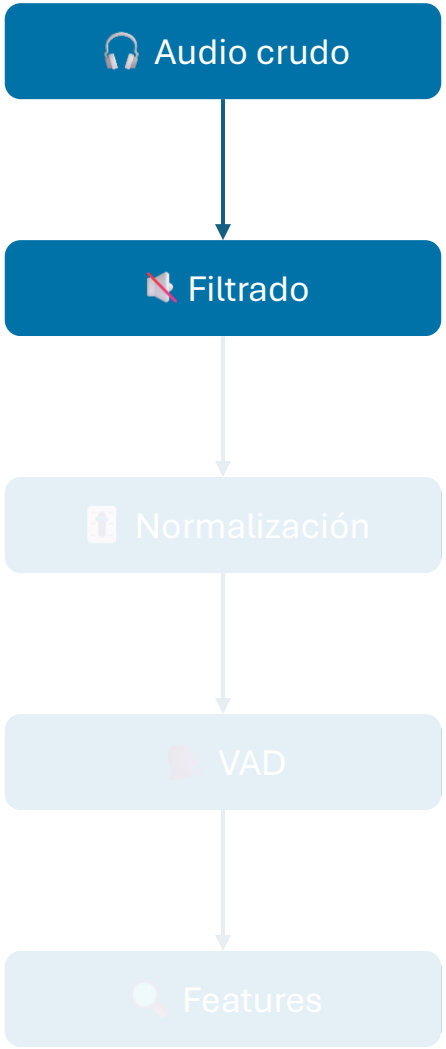
Preprocesamiento: Etapas

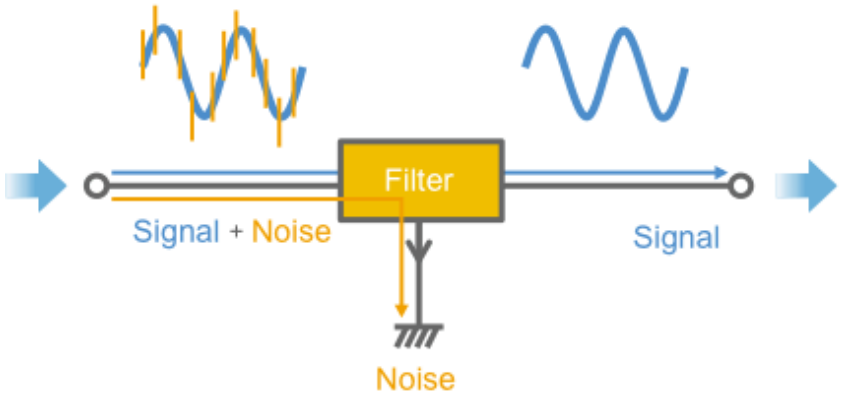
- El número de etapas a la hora de preprocesar un audio dependerá mucho de la arquitectura de IA específica a usar, así como de las necesidades de nuestro caso de uso. A continuación, se presenta un esquema típico de pipeline de preprocesamiento. En la práctica algunas etapas podrían añadirse o eliminarse según necesidades.



Preprocesamiento: Etapas

1. **Captura:** Obtención del audio a través de un micrófono.
2. **Filtrado:** Eliminar información no deseada. Técnicas comunes:
   - Filtro pasa banda: Mantiene frecuencias del rango del habla
   - *Espectral substraction*: Resta el espectro de un ruido estimado
   - Aprendizaje profundo: Existen redes neuronales específicas para el *Noise reduction* (usualmente usado en asistentes de voz modernos).

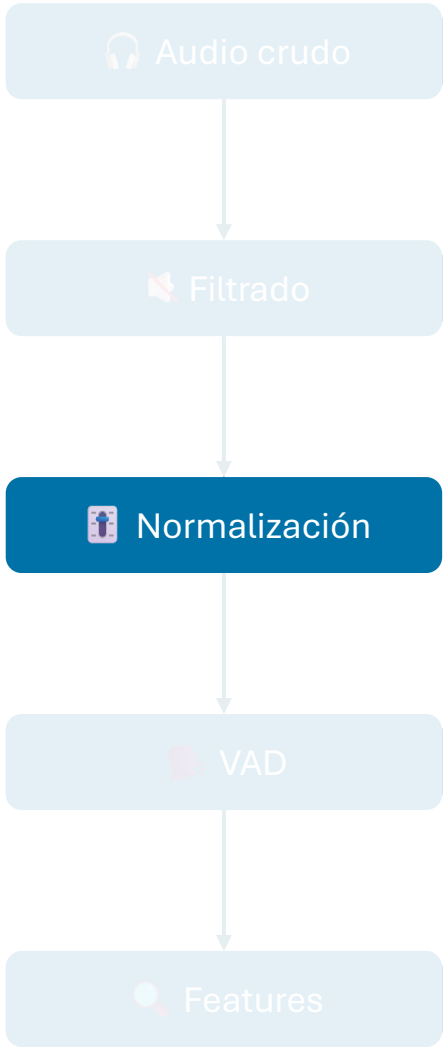


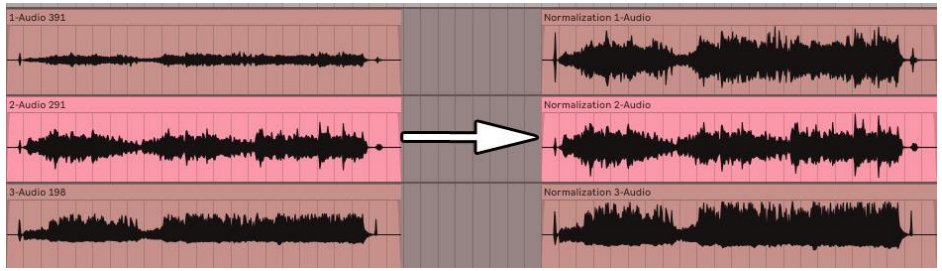


Filtro de reducción de ruido. Extraído de enlace.

Preprocesamiento: Etapas

3. **Normalización:** Ajusta el volumen/energía para que el modelo no dependa del micrófono o del usuario. Una práctica común es escalar la amplitud de la señal al rango [-1,1]

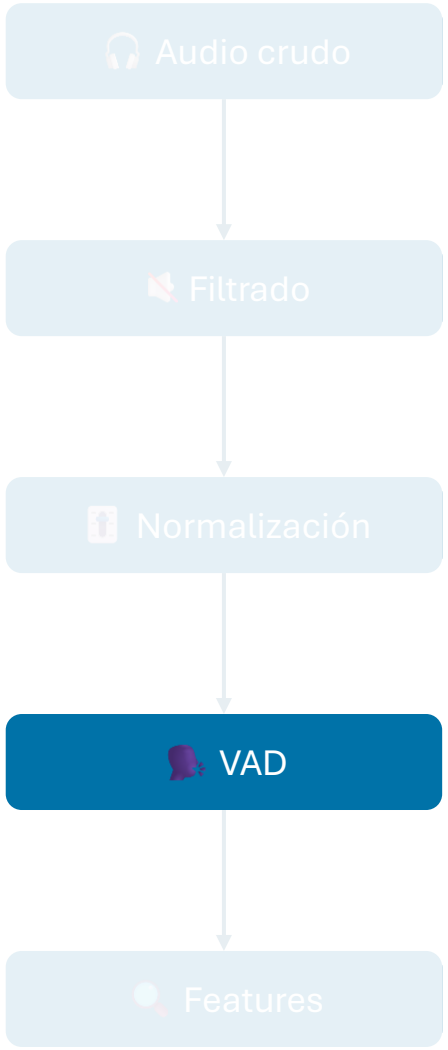


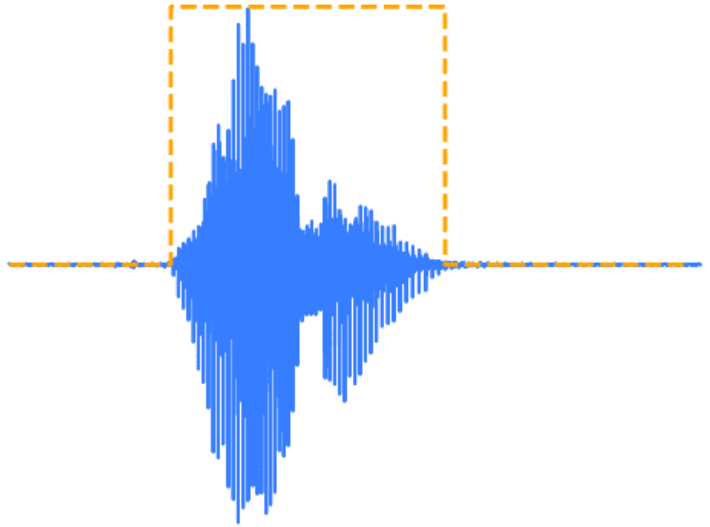


Normalización de audio. Extraído de enlace.

Preprocesamiento: Etapas

4. **VAD (Voice Activity Detection):** Separar segmentos donde hay habla de los silencios/ruido. Este componente detecta cuándo empieza y termina de hablar la persona para evitar el procesamiento innecesario de intervalos vacíos. Se puede hacer de varias formas:
   - Basado en energía (mide amplitudes del sonido)
   - Basado en modelos (RNNs, CNNs de detección de voz)

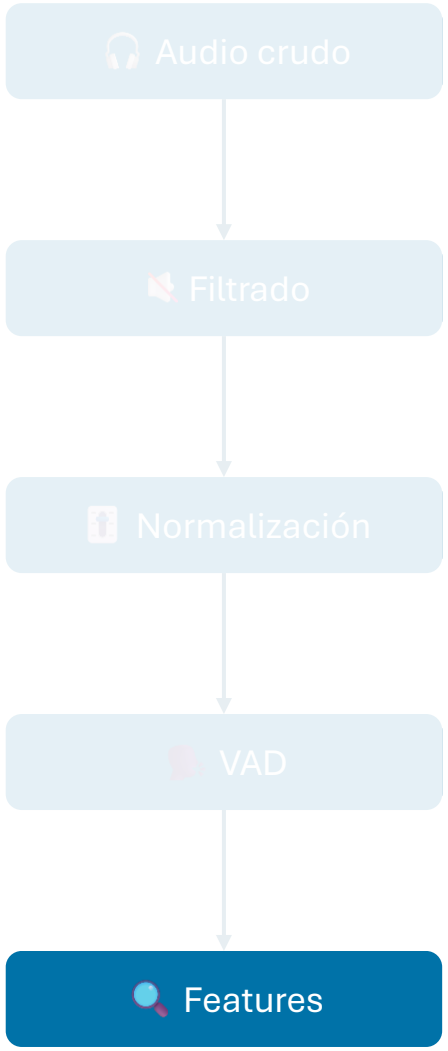


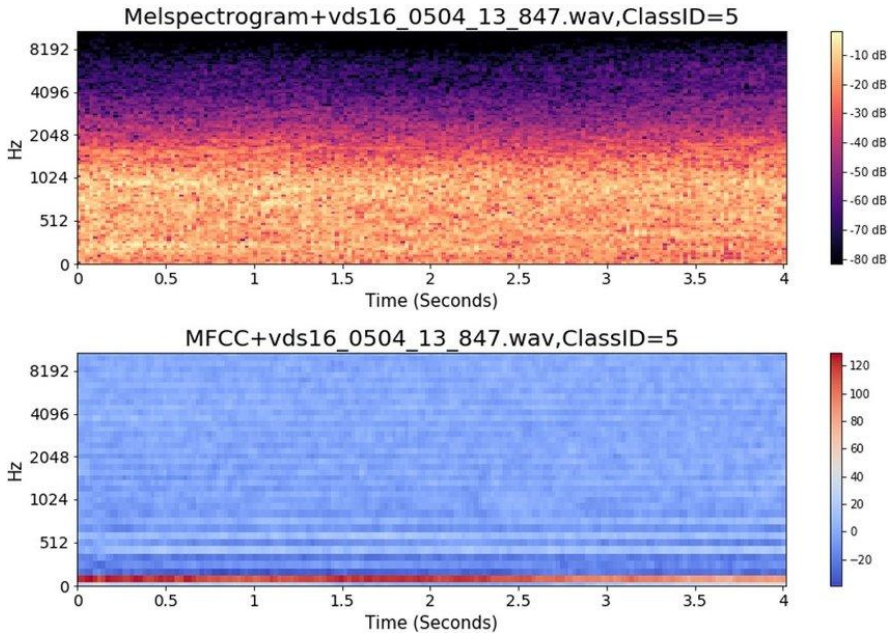


Normalización de audio. Extraído de la documentación de Picovoice.

Preprocesamiento: Etapas

5. **Extracción de features:** Convertir el audio en una representación numérica informativa para redes neuronales. Este paso se compone de dos pasos principales: Primero se divide el audio en ventanas de 20-40ms (*windowing*) y luego se obtiene la representación. A continuación, las principales:
   - Espectograma de corto plazo (STFT)
   - Mel Spectrogram
   - Mel-Frequency Cepstral Coefficients (MFCC)

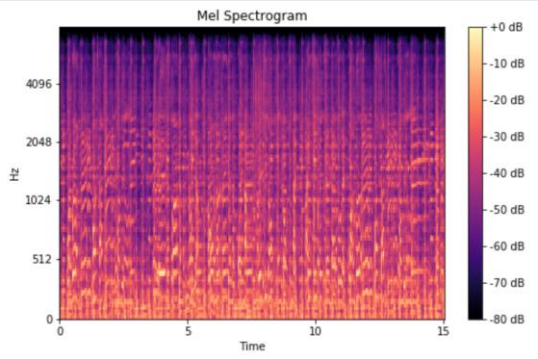




Visualización de representaciones. Extraído de *Nam et al. (2020). Traffic Density Classification Using Sound Datasets: An Empirical Study on Traffic Flow at Asymmetric Roads*.

Preprocesamiento: Embeddings

- Representaciones como el espectrograma de Mel o los coeficientes MFCC ya pueden ser usados como input de modelos de *Deep learning*, sin embargo, estas representaciones pueden convertirse en *embeddings*, que enriquecen la representación incluyendo información semántica/fonética.
- Estos *embeddings* los generan arquitecturas tipo CNN/Transformer como Wav2Vec [7], YAMNet [12] o el encoder de Whisper [15].



$$features = [x1, x2, x3, \dots]$$

Asistentes de voz

Asistentes de voz: Definición y hardware

Sistemas de IA diseñados para interactuar mediante diálogo hablado. Sus micrófonos tienen las siguientes características:

- Diseñados para captura de campo lejano *(far-field)*: Para que el asistente pueda escucharte desde la lejanía, el usuario no tiene porque estar cerca del dispositivo.
- Usan micrófonos de array. Normalmente utilizan arrays de múltiples micrófonos (ej. 4 u 8 micrófonos dispuestos circularmente)

Asistentes de voz: Software

Tecnologías para mejorar la calidad de la escucha:

- ***Beamforming*:** Orientar la escucha hacia la voz del usuario y filtrar ruidos de fondo
- **Reducción de ruido**
- **Cancelación de eco:** Eco del propio sonido del asistente si este está reproduciendo audio (p. ej. música)
- ***Wake Word Spotting:*** Detección de palabras clave (p. ej. Detectar cuándo se dice “Alexa”)

Algunos ejemplos de asistentes pueden ser: *Siri* (Apple), *Google Assistant* (Google), *Alexa* (Amazon) o *Cortana* (Microsoft)

Asistentes de voz: Software

Tecnologías pa

- *Beamformin* filtrar ruidos de fondo
- Reducción d
- Cancelación d este está reproduciend
- *Wake Word S* tar cuándo se dice “Ale

Algunos ejem le Assistant* (Google), *Alexa*

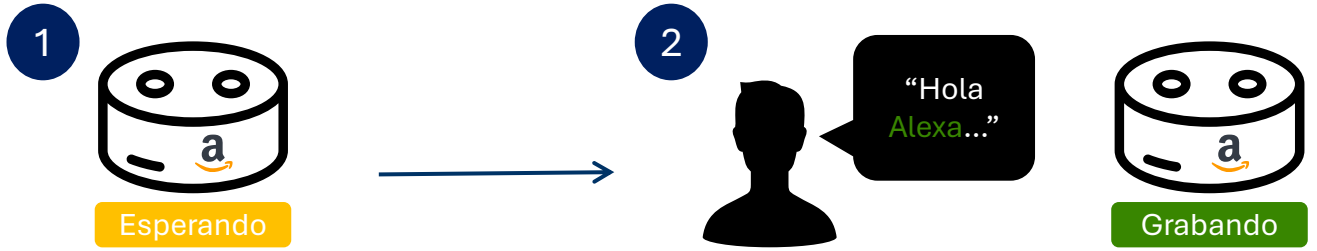
Asistentes de voz: Pipeline

Cada asistente tiene sus peculiaridades, pero en general todos siguen un pipeline similar:

1. **Reposo:** Dispositivo en reposo hasta la escucha de la palabra de activación.

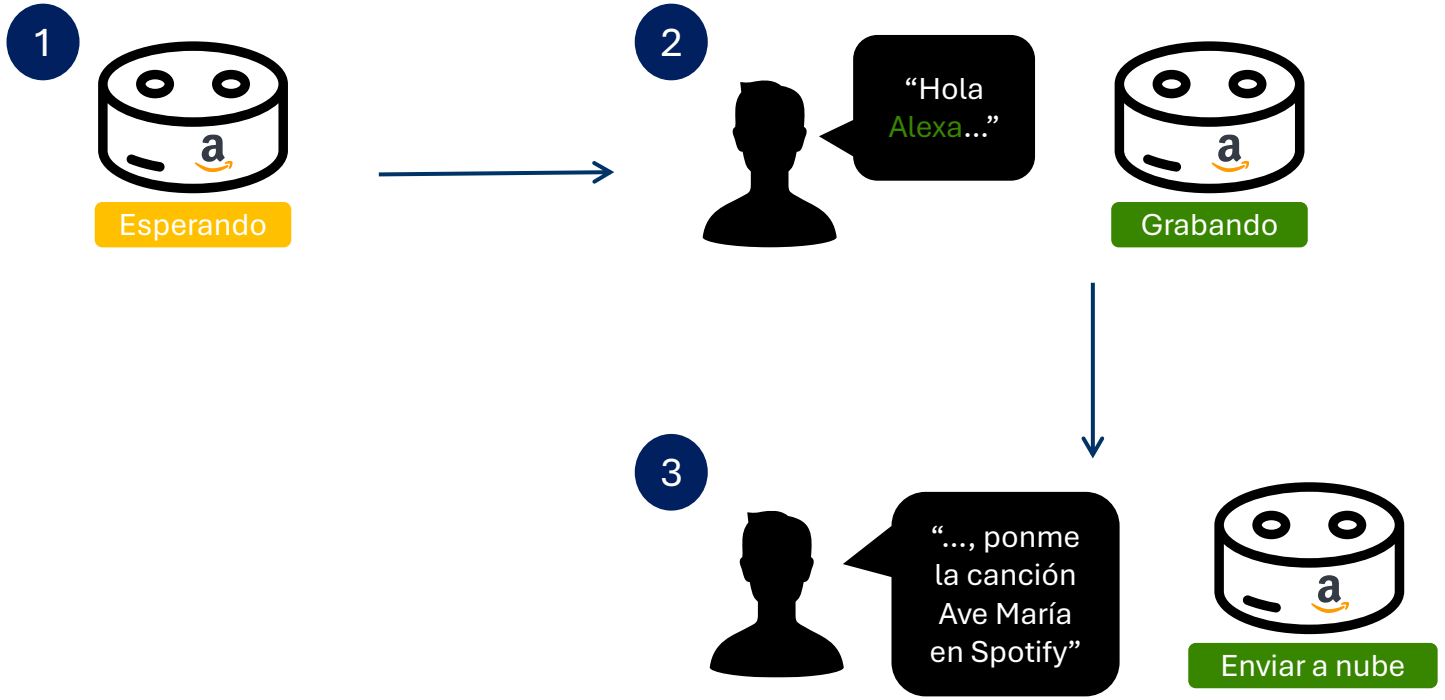
Asistentes de voz: Pipeline

2. **Activación:** Dispositivo detecta la palabra de activación (“*Hey Siri*”, “*Ok Google*”, “*Alexa*”, etc). Esto suele hacerse con un modelo ligero de reconocimiento de palabras clave *(Wake Word Spotting)* que corre localmente en el dispositivo.



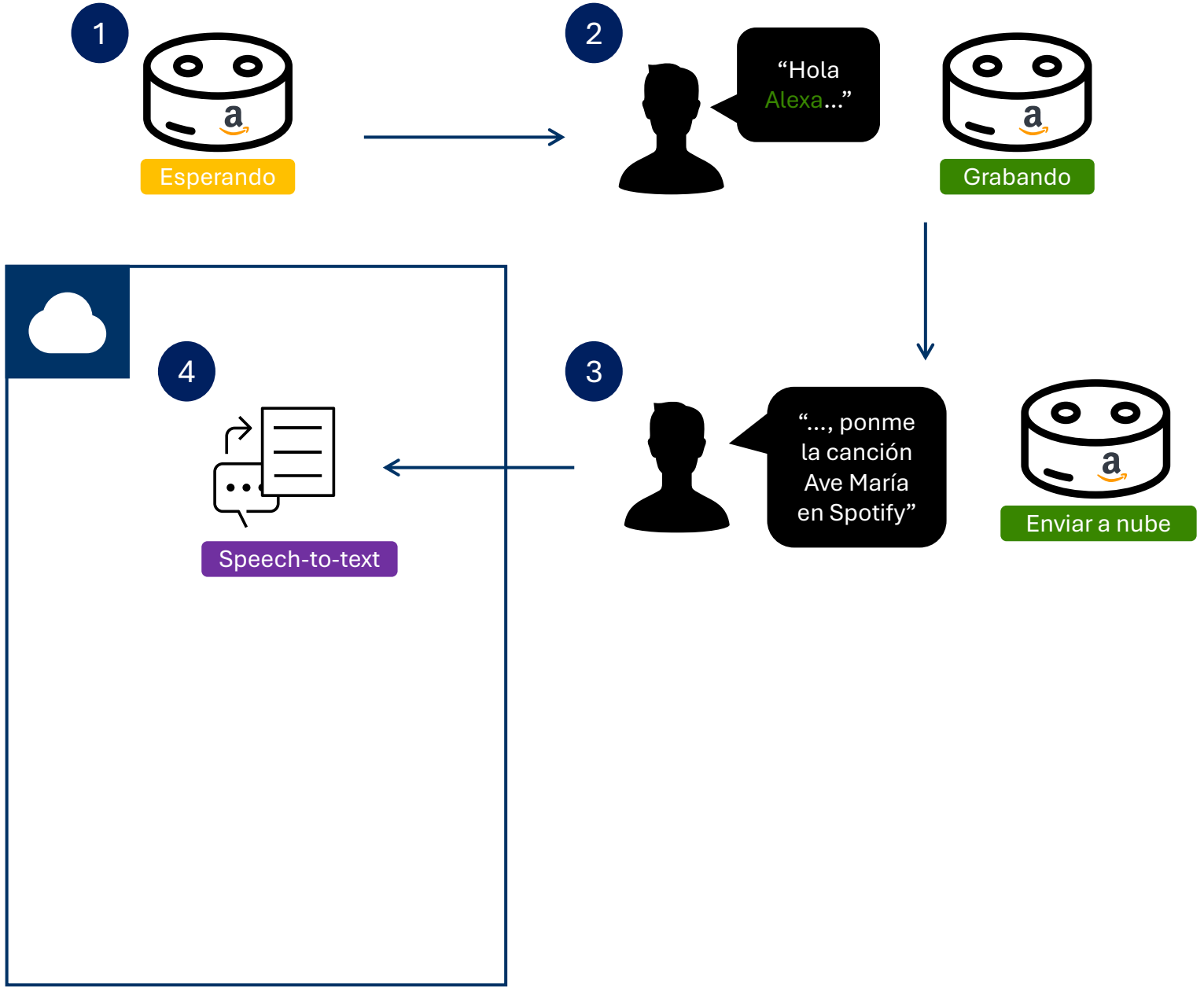
Asistentes de voz: Pipeline

3. **Captura y envío de audio:** Una vez activado, el asistente graba la voz hasta detectar final de frase (modelo *end-of-turn*) y envía audio a la nube para su procesamiento.



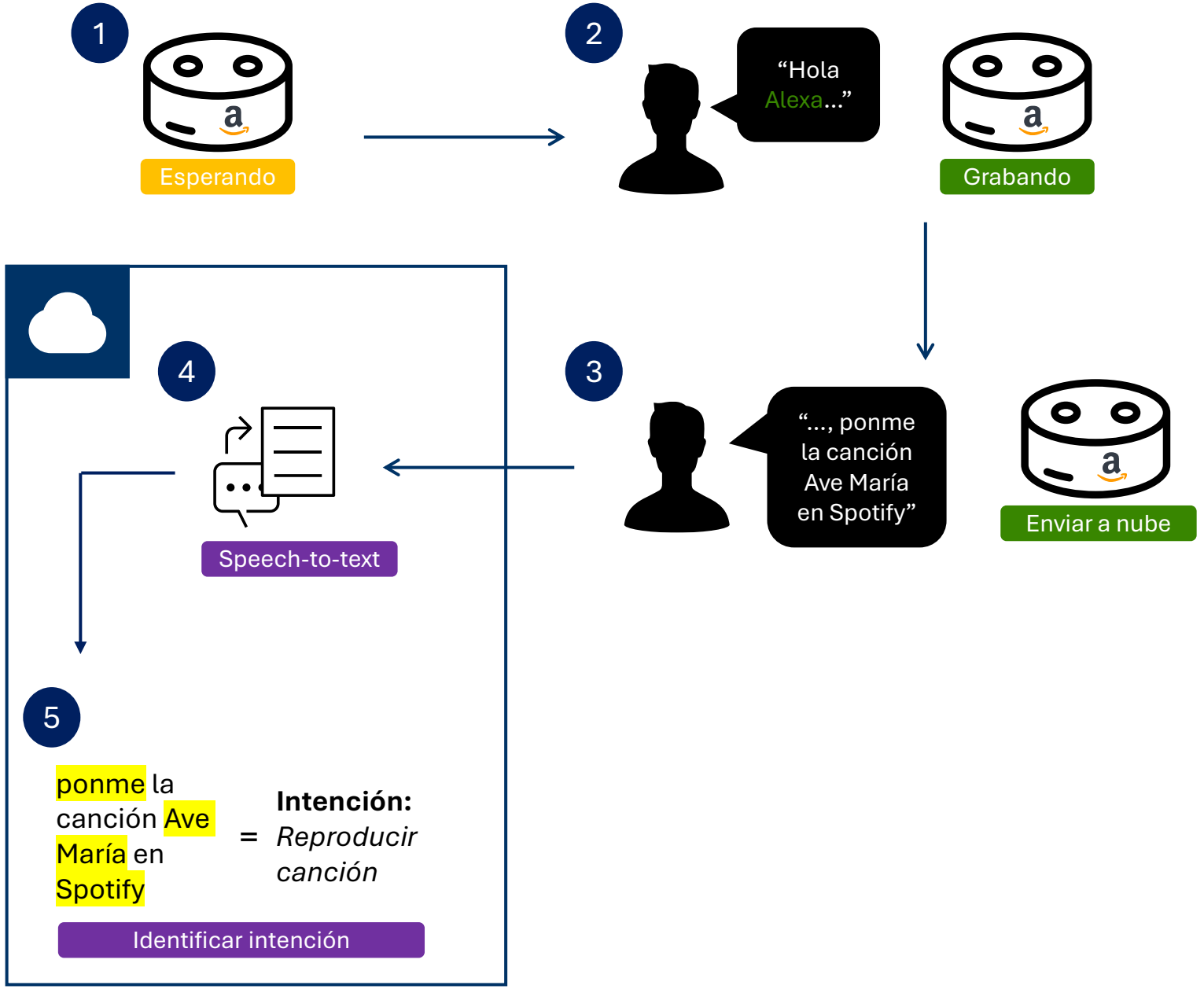
Asistentes de voz: Pipeline

4. **Reconocimiento de voz (ASR):** En la nube, un modelo de reconocimiento automático del habla convierte el audio en texto. Antiguamente se usaban modelos clásicos de *machine learning* como los Modelos Ocultos de Markov (HMM). Hoy en día lo más común es el uso del *Deep Learning,* dónde las redes neuronales son capaces de realizar el reconocimiento con mucha precisión.



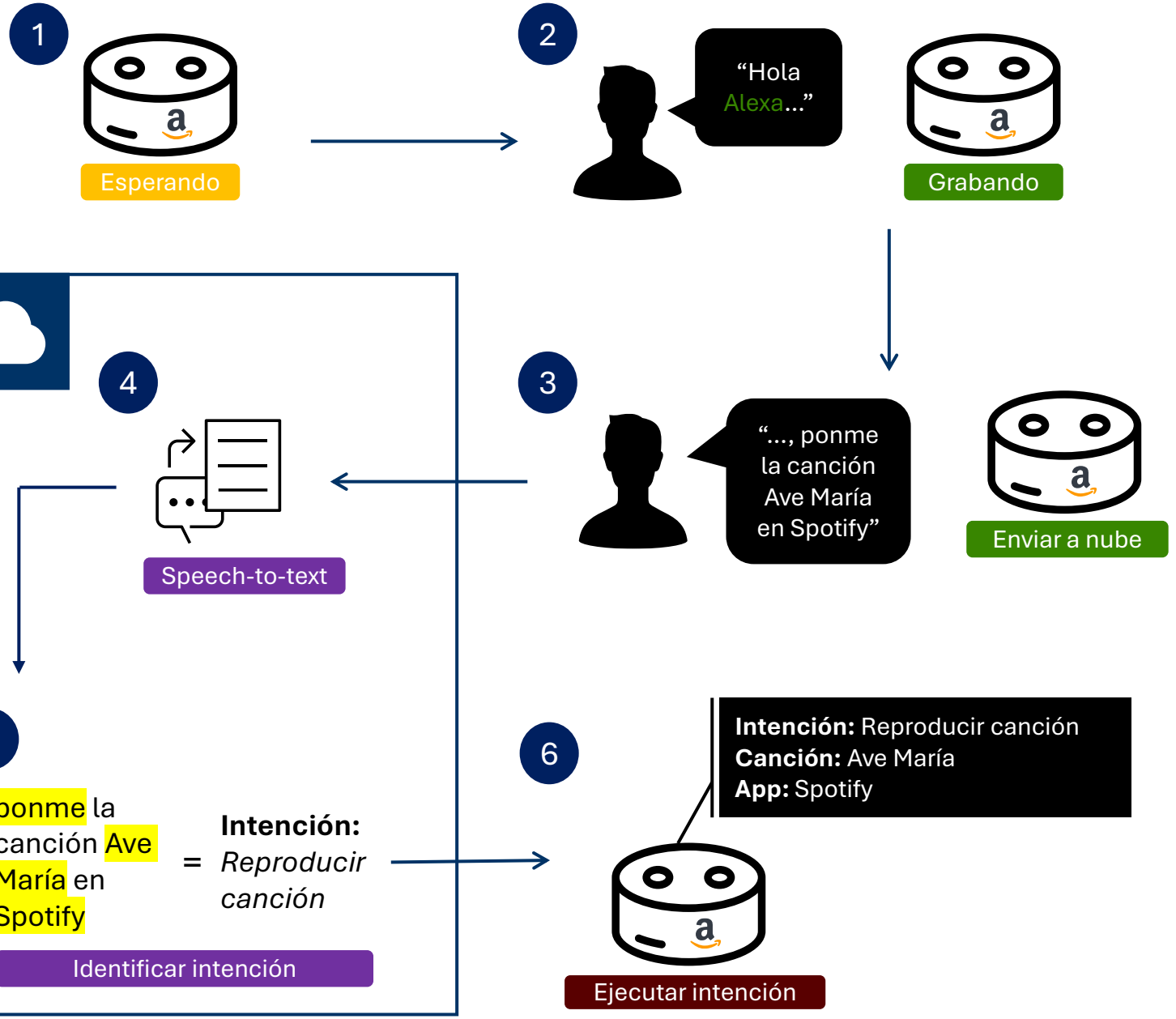
Asistentes de voz: Pipeline

5. **Compresión del lenguaje natural (NLU):** Una vez el texto se ha transcrito, el asistente debe entender la intención del usuario. Es decir, identificar que acción y respuesta se requiere. Este suele ser un módulo más complejo donde se realizan distintas tareas, como la tokenización, la clasificación de la intención y la extracción de entidades. Los asistentes modernos usan LLMs (p. ej. GPT-4o) para mejorar la comprensión.



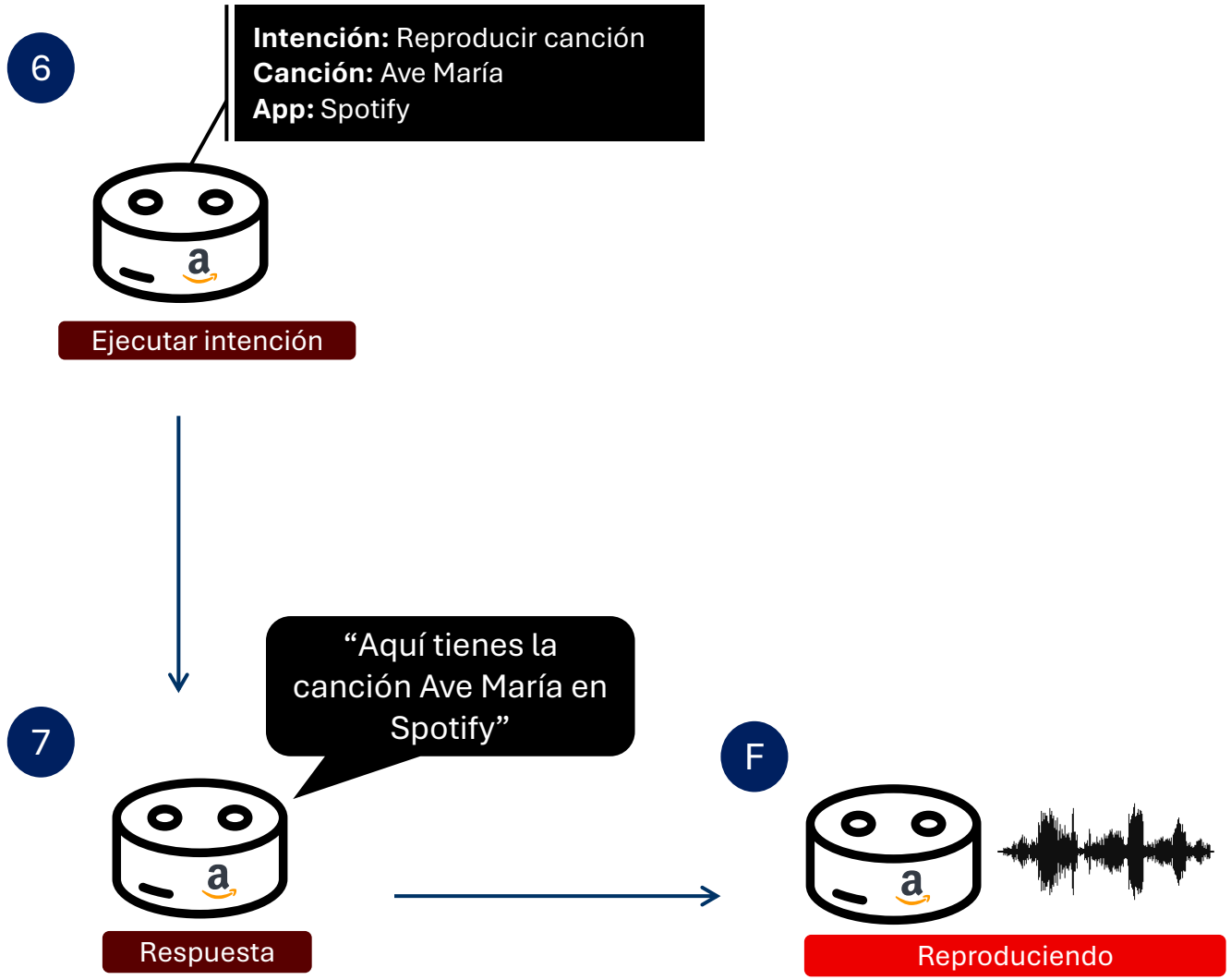
Asistentes de voz: Pipeline

6. **Gestión del diálogo y acción:** La información viaja desde la nube al asistente, el cuál determina cómo responder. Puede implicar consultar alguna base de conocimiento, ejecutar una acción o simplemente elaborar una respuesta.



Asistentes de voz: Pipeline

7. **Síntesis del habla (TTS):** Finalmente, los asistentes generan una respuesta conversacional usando modelos *Text-To-Speech*. Al igual que en el ASR, en el TTS también domina el uso de redes neuronales para generar voces hiper realistas y respuestas naturales.



Asistentes de voz: Retos I

Y esto no es todo! Los asistentes son sistemas de IA complejos que usan una gran variedad de **submódulos** para mejorar la precisión, usabilidad y naturalidad en las interacciones. Estos submódulos buscan solucionar distintos desafíos a los que se enfrenta un sistema inteligente cuándo se despliega en entornos reales. Pongamos el ejemplo de Alexa…

Voice-adapted

Alexa usa propiedades acústicas de la voz de la persona que realiza la petición de activación "*Alexa*", para filtrar las voces de fondo que producen ruido en las siguientes interacciones.

Context-aware

Alexa también usa el contexto del dispositivo, que puede incluir el día que es, las interacciones anteriores del usuario, sus preferencias o el lugar en el que está colocado el dispositivo.

Error handling

Al ejecutar ciertas acciones, se pide confirmación repitiendo alguno de los datos descritos por el usuario. Esto también ocurre cuándo la confianza del asistente es baja.

Asistentes de voz: Retos I

Y esto no es todo! Los asistentes son sistemas de IA complejos que usan una gran variedad de **submódulos** para mejorar la precisión, usabilidad y naturalidad en las interacciones. Estos submódulos buscan solucionar distintos desafíos a los que se enfrenta un sistema inteligente cuándo se despliega en entornos reales. Pongamos el

Esto junto a otros componentes (como los modelos *wake word spotting* o *end-of-turn*) es lo que componen lo que conocemos hoy en día como **asistentes de voz**.

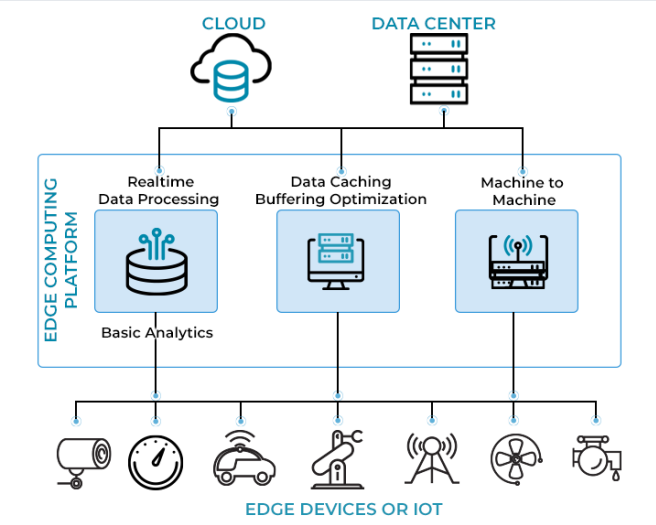
Alexa usa propiedades acústicas de la voz de la persona que realiza la petición de activación "*Alexa*", para filtrar las voces de fondo que producen ruido en las siguientes interacciones.

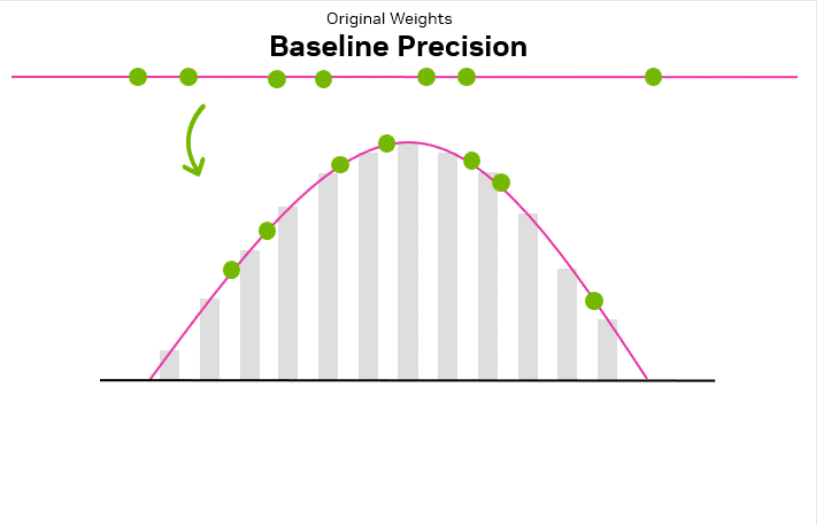
Alexa también usa el contexto del dispositivo, que puede incluir el día que es, las interacciones anteriores del usuario, sus preferencias o el lugar en el que está colocado el dispositivo.

Al ejecutar ciertas acciones, se pide confirmación repitiendo alguno de los datos descritos por el usuario. Esto también ocurre cuándo la confianza del asistente es baja.

Asistentes de voz: Retos II

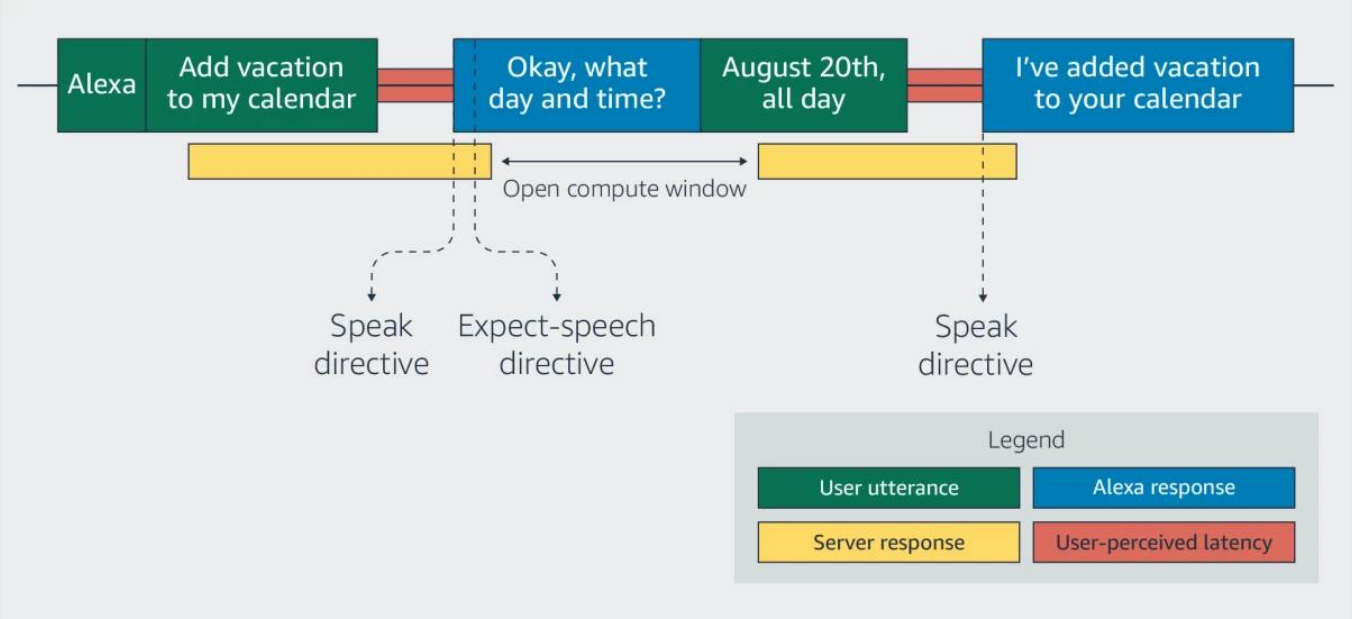
Y ojo! Que todo esto debe ocurrir en tiempo real para que la interacción sea fluida. Las grandes empresas optimizan la latencia llevando ciertos cómputos al dispositivo (p. ej. *on-the-edge computing*), usando compresión de modelos (p. ej. *model quantization*) y servidores muy escalables (p. ej. AWS EC2 Autoscaling).





Asistentes de voz: Retos III

Pero aún hay más! Los asistentes de voz no solo presentan retos a nivel de IA, sino también a nivel de arquitectura software. Estos son obras de ingeniería sofisticadas que han pasado por un largo proceso iterativo de desarrollo para adaptarse perfectamente al uso en la vida diaria.



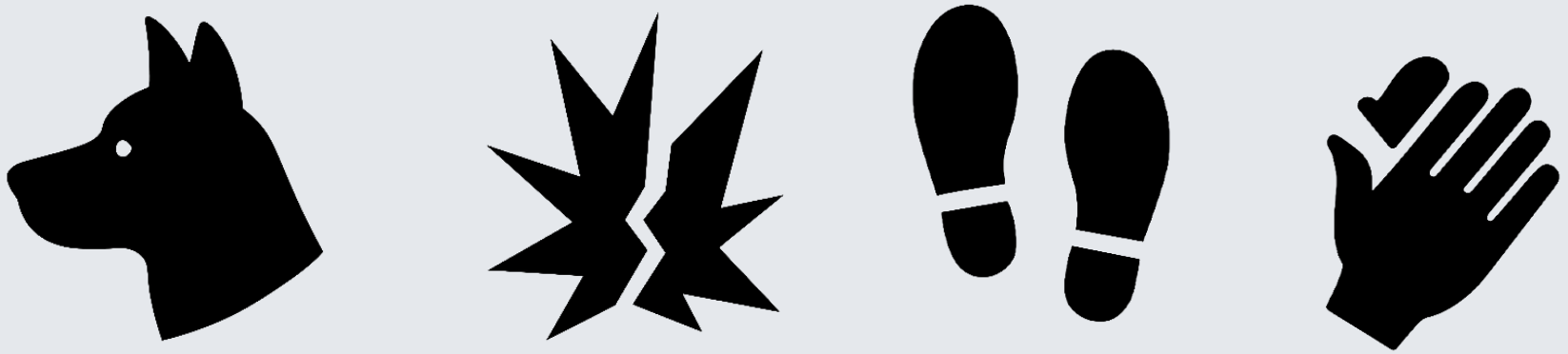
“This work is the culmination of very close collaboration between scientists and engineers to design contextual machine learning to operate at Alexa scale”

Debprakash Patnaik, Ph.D.
Senior ML Engineering Manager @ Amazon AGI

Reconocimiento de sonidos ambientales (ESR)

ESR: Introducción

Nuestros entornos están llenos de **sonidos** relevantes: timbres, pasos, aplausos, sirenas, ladridos, rotura de cristales, entre muchos más. El **reconocimiento de sonidos ambientales** (también llamado *Environmental Sound Recognition*, ESR) es la capacidad de un sistema para identificar y clasificar estos sonidos no verbales.



ASR vs ESR

Similitudes	Diferencias
En ambos campos predomina el uso de redes neuronales como CNNs o Transformers	Mientras que el análisis del habla tiene una estructura sintáctica clara, los sonidos son más diversos.
	Escasez de datos para sonidos raros o eventos de corta duración.
	Mientras que dentro del ASR hay una gran cantidad de subtareas (transcripción, subtitulado en streaming, clasificación), el ESR es más limitado y suele trabajarse como una tarea de clasificación de audio

ESR: Limitación

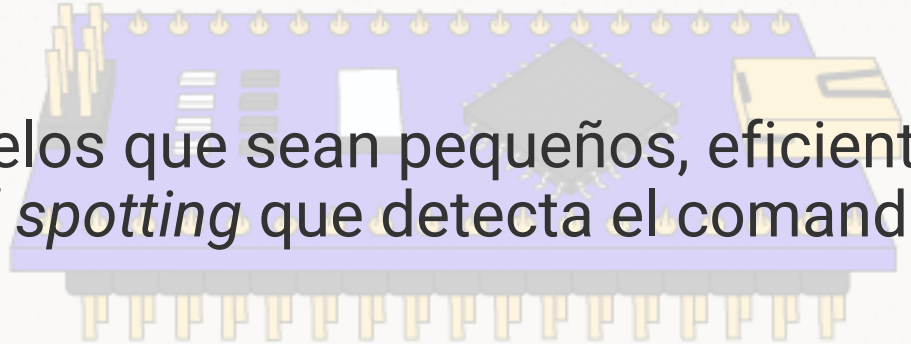
En aplicaciones reales (asistentes de voz, wearables, sensores IoT...), necesitamos detectar eventos acústicos como pasos, alarmas, conversaciones o caídas directamente en el dispositivo:

- Sin enviar audio a la nube → Mejor privacidad
- Con baja latencia → Respuestas en tiempo real
- Con bajo consumo energético → Adecuado para dispositivos ligeros

Esto propició el desarrollo del **TinyML**.

ESR: TinyML

TinyML (Tiny Machine Learning) es una rama de la inteligencia artificial que se centra en ejecutar modelos de aprendizaje automático directamente en dispositivos muy pequeños, con recursos muy limitados (microcontroladores, sensores, wearables, IoT).



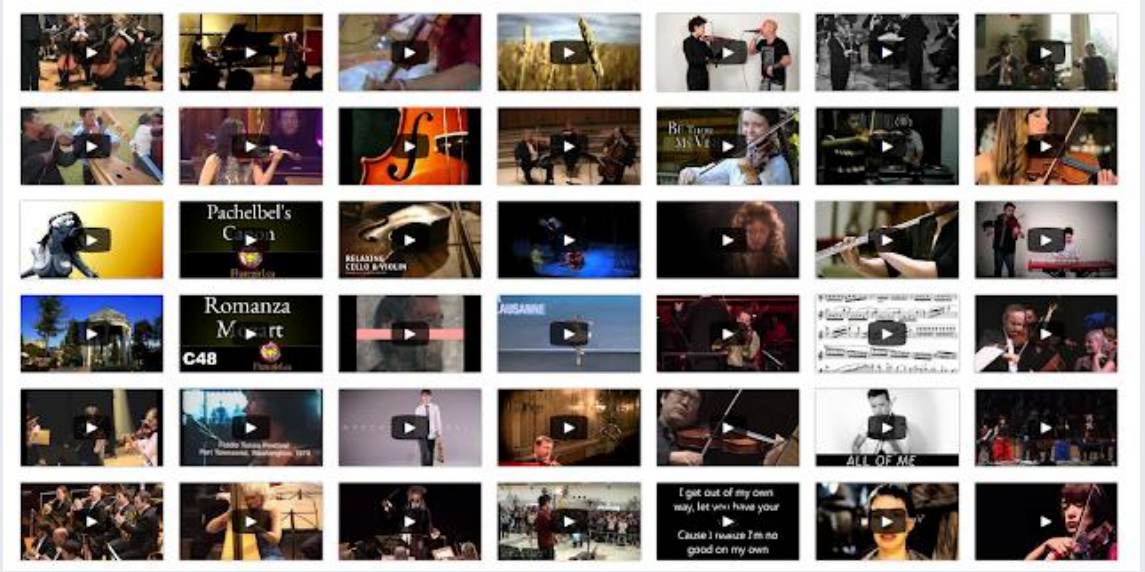
El TinyML requiere modelos que sean pequeños, eficientes y rápidos (p. ej. El modelo de *wake word spotting* que detecta el comando “Hey Google” en el *Google Assistant*).

ESR: Modelos

Modelo	Arquitectura Interna	Ideal para...	¿TinyML?
YAMNet [12]	Red ligera basada en una CNN (MobileNet). Utiliza capas convolucionales separables para extraer características del espectrograma log-Mel.	Clasificación de sonidos en tiempo real. Detección simple o embedded (dispositivos móviles).	✅ Si (muy ligero)
PANNs [13]	Conjunto de modelos **CNN** con atención global. Procesan espectrogramas Mel y permiten interpretar qué parte del audio activa cada clase.	Clasificación robusta de sonido, tagging, localización de eventos.	⚠️ Parcial, dependiendo de la versión del modelo
AST [14]	Modelo basado en **Transformer tipo ViT**. Toma espectrogramas log-Mel y los divide en patches como si fueran imágenes	Clasificación avanzada, transfer learning, alto rendimiento en tareas complejas.	❌ No (modelo pesado, necesita GPU o CPU potente)

ESR: Datasets

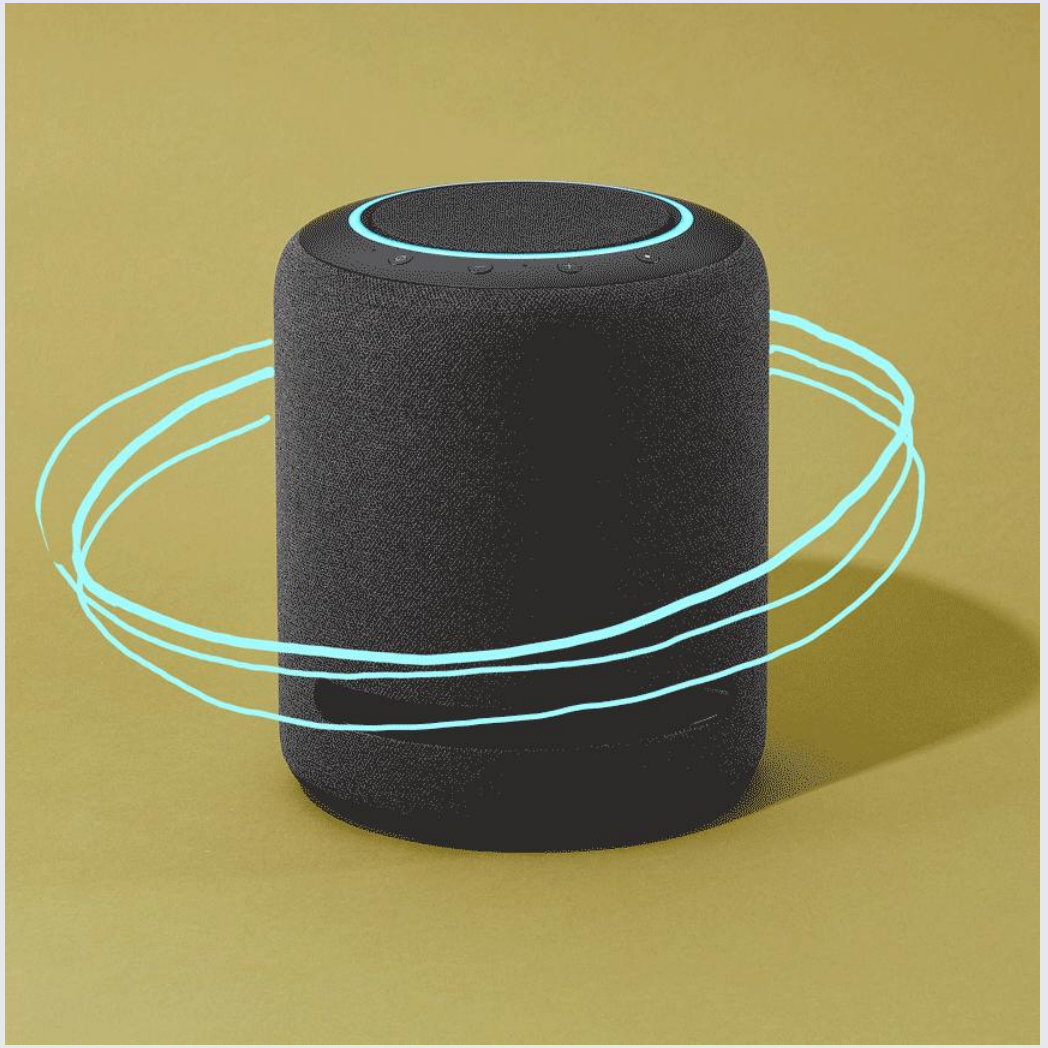
- **UrbanSound8k [9]:** 8,732 sonidos urbanos variados en 10 clases
- **ESC-50 [10]:** 2,000 clips cortos en 50 clases
- **AudioSet [11]:** Dataset de gran escala creado por Google. Alberga más de 2M de clips y tiene 527 clases de sonido distintos



Segmentos representando la clase *Violin*, en el dataset *AudioSet*. Extraído de https://research.google/blog/announcing-audioset-a-dataset-for-audio-event-research/

ESR: Alexa Guard

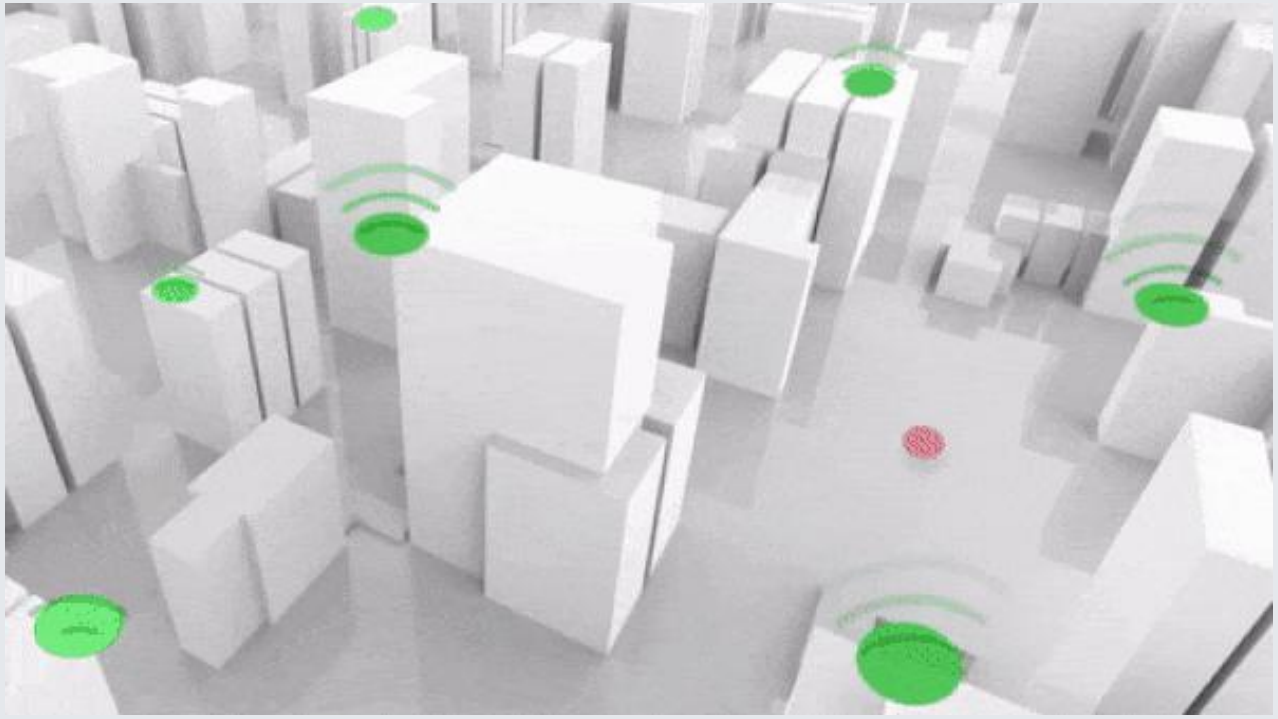
En una **casa inteligente**, un asistente podría *escuchar* eventos importantes. Amazon Alexa cuenta con la función **Alexa Guard**, donde los altavoces Echo detectan automáticamente sonidos de alarma de humo o cristal rompiéndose cuando el usuario no está en casa, y envían una alerta al móvil.



ESR: Seguridad

En entornos de seguridad, el análisis de audio puede detectar disparos, rompimiento de puertas o cristales en intrusiones. Un ejemplo de esto es *ShotSpotter*, un sistema de la empresa SoundThinking que detecta disparos y triangula su posición.

Este tipo de sistemas es crítico en entornos con altas tasas de criminalidad, como es el caso de Estados Unidos, que tiene una amplia red de micrófonos alrededor de más de 170 ciudades para tratar de reducir la violencia armada. Recientemente, este sistema se ha lanzado también en América Latina (Uruguay).



ESR: Vehículos autónomos

En el campo de la conducción autónoma [9], se investiga usar micrófonos para detectar sirenas de ambulancias, bocinas u otros sonidos de tráfico que un vehículo debería tener en cuenta.

Esto complementaría la visión (cámaras, LIDAR) para una percepción del entorno más completa.

Referencias

- [1] Hidden Markov Models (2025). ScienceDirect. https://www.sciencedirect.com/topics/neuroscience/hidden-markov-model
- [2] Mael Fabien (2020, Mayo 26) Introduction to Automatic Speech Recognition (ASR). https://maelfabien.github.io/machinelearning/speech_reco/#
- [3] Jurafsky, D., & Martin, J. H. (2025). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/
- [4] Matt Payne (2021, Sept 29). What is Beam Search? Explaining The Beam Search Algorithm. Width.ai. https://www.width.ai/post/what-is-beam-search
- [5] Andrew Maas (2025). Spoken Language Processing (CS224S). Stanford. https://web.stanford.edu/class/cs224s/semesters/2025-spring/syllabus
- [6] Mamyrbayev Orken et al. (2022). A study of transformer-based end-to-end speech recognition system for Kazakh language. Scientific Reports. https://doi.org/10.1038/s41598-022-12260-y
- [7] Alexei Baevski et al. (2020). Wav2vec 2.0: A framework for Self-Supevised Learning of Speech Representations. https://arxiv.org/pdf/2006.11477
- [8] Loren Lugosch (2020). Sequence-to-sequence learning with Transducers. https://lorenlugosch.github.io/posts/2020/11/transducer/

Referencias

- [9] André Luiz Florentino (2025). A dataset for environmental sound recognition in embedded systems for autonomous vehicles. Scientific Data. https://doi.org/10.1038/s41597-025-05446-2
- [10] Karol J. Piczak (2015). ESC: Dataset for Environmental Sound classification. Proceedings of the 23rd Annual ACM Conference on Multimedia. http://dx.doi.org/10.1145/2733373.2806390
- [11] Jort F. Gemmeke (2017). Audio Set: An ontology and human-labeled dataset for audio events. IEEE ICASSP. https://ieeexplore.ieee.org/abstract/document/7952261
- [12] Konstantinos Drossos (2020). Sound Event Detection with Depthwise Separable and Dilated Convolutions. https://arxiv.org/abs/2002.00476
- [13] Qiuqiang Kong (2019). PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. https://arxiv.org/abs/1912.10211
- [14] Yuan Gong (2021). AST: Audio Spectrogram Transformer. https://arxiv.org/abs/2104.01778
- [15] Radford et al. (2022). Robust speech recognition via large-scale weak supervision. https://doi.org/10.48550/arXiv.2212.04356