

vosk-cli-dictation — Manual del usuario

Instalar, configurar y usar el dictado por voz sin conexión

Version v0.5.4-beta

2026-09-18

Índice general

Arranque rápido	3
Capítulo 1 — Inicio	4
Requisitos previos	4
Varias formas de instalar	4
Recuperar un modelo de lenguaje	5
Primer arranque	5
El primer dictado	6
Dictar la puntuación	6
Desinstalar y actualizar	6
Capítulo 2 — El motor de reconocimiento de voz	8
¿Para quién es esta herramienta?	8
Lo que hace Vosk	8
Modelo acústico y modelo de lenguaje	9
Nada sale de la máquina	9
Capítulo 3 — Los tres estados: ÉCOUTE, DICTÉE, VEILLE	10
Resumen	10
DICTÉE — dictado en curso	10
ÉCOUTE — en espera	10
¿Cómo iniciar y detener un dictado?	10
VEILLE — consumo casi nulo, voluntario	11
El umbral de confianza determina el momento de activación	11
Cuándo usar el modo de espera	11
El desarrollo de una sesión completa	12
Capítulo 4 — Configuración y personalización	13
¿Dónde se encuentra el archivo de configuración?	13

Elegir el idioma de la plantilla de configuración	13
Personalizar los atajos de teclado	13
Elegir cómo comienza y termina la dictación	14
Puntuación: dos modos de entrada, por idioma	14
Ajustar el umbral de confianza para la activación	14
Agregar vocabulario y corregir confusiones	15
¿Dónde se buscan los modelos de lenguaje?	15
Capítulo 5 — Integración del sistema: Wayland y X11	16
¿Cómo llega el texto a la aplicación?	16
El micrófono	16
Las herramientas del sistema utilizadas, y su selección automática	16
Activar ydotool (solución de respaldo)	17
Atajos de teclado: X11 y Wayland no son iguales	17
¿Qué sucede si no hay ninguna herramienta de inyección disponible?	18
Capítulo 6 — Solución de problemas	19
Síntoma → causa probable → acción	19
¿Dónde buscar un mensaje de error completo?	20
Si el problema persiste	21

Arranque rápido

Para aquellos que quieran dictar en dos minutos, sin leer el resto de inmediato.

`vosk-cli-dictation` está diseñado para entornos con recursos limitados y para la transcripción offline, no para lograr la máxima precisión en una estación de trabajo potente (Capítulo 2). Los atajos de teclado que se muestran a continuación asumen una sesión X11; en Wayland, se utilizan a través del entorno de escritorio (Capítulo 5).

1. Instalación. Paquete nativo de la distribución:

```
# Debian 12+, Ubuntu 24.04+, Mint 22+ ...
sudo apt install ./vosk-cli-dictation-<version>_all.deb

# Fedora
sudo dnf install ./vosk-cli-dictation-<version>.fc42.noarch.rpm

# openSUSE (Leap)
sudo zypper install ./vosk-cli-dictation-<version>.leap156.noarch.rpm

# Arch Linux
sudo pacman -U ./vosk-cli-dictation-<version>-any.pkg.tar.zst
```

2. Recuperar un modelo de lenguaje, si no hay uno presente (el paquete no los incluye, ya que son demasiado grandes):

```
vosk-cli-dictation --download-model fr
```

El pequeño modelo del lenguaje (aquí `fr`; también `en`, `de`, `es`) se descarga, se verifica mediante su suma SHA256 y se instala en `~/.local/share/vosk-models/`. La primera vez que se inicia la herramienta sin ningún modelo, esta ofrece la misma descarga de forma interactiva. También se puede descargar un modelo manualmente desde <https://alphacephei.com/vosk/models> y descomprimirlo en este directorio.

3. Dictado.

```
vosk-cli-dictation
```

Para iniciar la dictación, diga la frase de inicio (“iniciar dictado”), hable, y luego diga la frase de finalización (“terminar dictado”) para terminar, y copie el texto en el portapapeles.

Eso es todo para empezar. Lo que sigue explica por qué y cómo avanzar: personalizar los accesos directos, comprender el modo de espera, solucionar problemas de bloqueo.

Capítulo 1 — Inicio

Requisitos previos

- **Un sistema Linux** con una sesión gráfica X11 o Wayland. Sin un servidor de visualización, no hay una ventana activa donde escribir el texto reconocido.
- **Python 3** y un entorno virtual (python3-venv).
- **Una herramienta de inyección de texto:** wtype o ydotool bajo Wayland, xdotool bajo X11. Al menos una debe estar presente, de lo contrario, la herramienta se niega a iniciar (Capítulo 5).
- **Un micrófono reconocido por el sistema** como dispositivo de entrada por defecto (Capítulo 5).
- **Espacio en disco para el modelo:** de 50 MB (modelos “small”) a más de un gigabyte. En funcionamiento, una sesión utiliza aproximadamente 300 MB de RAM (Capítulo 2).

El paquete Debian declara e instala por sí mismo todas sus dependencias, sistema y Python. En el modo “clone” de Git, `pip install -r requirements.txt` instala las bibliotecas de Python; la herramienta de inyección (wtype, xdotool o ydotool) debe instalarse por separado.

Varias formas de instalar

vosk-cli-dictation ofrece varios canales, sin que uno reemplace a otro.

El paquete nativo de la distribución (recomendado para uso común)

```
# Debian 12+, Ubuntu 24.04+, Mint 22+ ...
sudo apt install ./vosk-cli-dictation-<version>_all.deb

# Fedora
sudo dnf install ./vosk-cli-dictation-<version>.fc42.noarch.rpm

# openSUSE (Leap)
sudo zypper install ./vosk-cli-dictation-<version>.leap156.noarch.rpm

# Arch Linux
sudo pacman -U ./vosk-cli-dictation-<version>-any.pkg.tar.zst
```

Independientemente del formato, el paquete instala un único comando, `vosk-cli-dictation`, en `/usr/bin/`. También crea un entorno Python aislado e instala el motor de reconocimiento de voz (Vosk) en él; este paso requiere conexión a internet durante la instalación, solo una vez. No se requiere ser superusuario después de la instalación.

La instalación en sí misma nunca modifica los archivos personales: ninguna configuración existente se sobrescribe, ni se crea ni se modifica ningún directorio de la cuenta de usuario en esta etapa. Solo en el primer lanzamiento, y al recuperar un modelo de lenguaje (a continuación), aparecen archivos en el directorio personal; nunca antes, nunca sin una acción explícita del usuario.

Clona el repositorio Git (para seguir el desarrollo o contribuir)

```
git clone https://github.com/RonanDavalan/vosk-cli-dictation.git
```

```
cd vosk-cli-dictation
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

En este modo, la herramienta se inicia con `python3 src/main.py` en lugar de a través del comando `vosk-cli-dictation` del paquete, y su configuración se almacena en `config/config.yaml`, junto con el código; nunca en el directorio personal. Los dos modos no se mezclan: un clon Git ignora la configuración del paquete, y viceversa.

Recuperar un modelo de lenguaje

Ninguno de los dos canales incluye un modelo de reconocimiento de voz, ya que estos pesan entre 50 MB y más de un gigabyte, lo cual es demasiado para incluirlo con la herramienta.

La forma más sencilla, al instalar mediante paquetes, es dejar que la herramienta se encargue de ello:

```
vosk-cli-dictation --download-model fr
```

Esta orden recupera el modelo pequeño del idioma solicitado (`fr`, o `en`, `de`, `es`), verifica su suma SHA256, lo instala en `~/.local/share/vosk-models/`, y luego se detiene. En el primer lanzamiento sin ningún modelo, se ofrece la misma opción de forma interactiva: responder “sí” descargará el modelo pequeño del idioma de la interfaz.

Sinón, o para un modelo más completo que el pequeño modelo predeterminado, descargue manualmente en <https://alphacephei.com/vosk/models>, y luego descomprima:

- Instalación por paquete: en `~/.local/share/vosk-models/`.
- Clon de Git: en `vosk-model/`, en la raíz del repositorio clonado.

Sin un modelo presente en la ubicación correcta, la herramienta muestra al iniciarse las ubicaciones que ha buscado y la dirección de descarga; nunca un error silencioso.

Primer arranque

```
vosk-cli-dictation
```

(en modo clonación de Git, el mismo comando es `python3 src/main.py`, desde la raíz del repositorio y con el entorno virtual activado).

Se muestran tres cosas: la verificación de las dependencias externas (ver Capítulo 5), el modelo cargado y, luego, un recordatorio de los comandos disponibles: frases de activación y desactivación, atajos de teclado y palabras de puntuación. Este recordatorio se regenera en cada inicio a partir de la configuración activa: siempre refleja lo que es realmente utilizable, y nunca una lista fija en la documentación que podría volverse incorrecta después de una personalización.

El primer dictado

Dos formas equivalentes de comenzar una dictada, a elección:

- **Por voz:** Pronuncie la frase de activación de la lengua activa («iniciar dictado»). Se necesita la frase completa: una sola palabra («iniciar») no activa nada; la activación solo se mantiene para la frase completa, con suficiente confianza.
- **Por el teclado:** Presione **Alt+H**; este atajo alterna entre ÉCOUTE (el estado inicial, el que se tiene inmediatamente después del lanzamiento) y DICTÉE (ver Capítulo 3 para los estados); sirve tanto para iniciar como para pausar. Nunca copia nada en el portapapeles: vea más abajo para finalizar un dictado. En una sesión **Wayland**, estos atajos solo se capturan para las ventanas que pasan por XWayland; para una ventana Wayland nativa, deben pasarse por el entorno de escritorio (Capítulo 5).

Luego, simplemente habla normalmente. El texto reconocido se escribe directamente en la aplicación que tiene el foco: un terminal, un editor, un navegador, cualquier cosa.

Para finalizar y recuperar el texto en el portapapeles: diga la frase de finalización («terminar dictado») o presione **Alt+S**. Una frase de finalización escuchada por debajo del umbral de confianza se absorbe; no se escribe en el documento ni se activa, y se muestra una indicación: repítala o use **Alt+S**.

Un control más fiable, tanto en X11 como en Wayland, consiste en vincular las subcomandos (`--toggle`, `--finalize`, `--sleep-toggle`) a accesos directos del escritorio (Capítulo 5) en lugar de utilizar los accesos directos integrados.

Dictar la puntuación

Las palabras de puntuación se dictan como el resto del texto — «coma», «exclamación», «nueva línea», etc. La lista completa depende del idioma activo y se muestra al inicio (ver arriba); también es personalizable, y la forma de activar un comando de puntuación puede variar según el modo de entrada del idioma (Capítulo 4).

Desinstalar y actualizar

Paquete nativo. El gestor de la distribución elimina la orden y el entorno Python aislado (`/usr/lib/vosk-cli-dictation/venv`): `sudo apt remove vosk-cli-dictation` (Debian 12+ y derivados; `sudo apt purge vosk-cli-dictation` también elimina el modelo de configuración predeterminado), `sudo dnf remove vosk-cli-dictation` (Fedora), `sudo zypper remove vosk-cli-dictation` (openSUSE), `sudo pacman -R vosk-cli-dictation` (Arch Linux). El archivo de configuración personal (`~/.config/vosk-cli-dictation/config.yaml`) y los modelos de lenguaje (`~/.local/share/vosk-models/`) nunca se ven afectados por la desinstalación; se pueden eliminar manualmente si se desea.

Clonar Git. Eliminar el directorio clonado: contiene todo (código, entorno virtual `venv/`, posibles plantillas en `vosk-model/`). Nada se escribe en otra parte en este modo.

Actualización. Con un paquete nativo, instala el archivo más reciente con el mismo comando que para la primera instalación (`apt install`, `dnf install`, `zypper install`, `pacman -U`): reemplaza la versión exis-

tente sin modificar la configuración personal. Con el clon de Git, haz `git pull`, y luego vuelve a ejecutar `pip install -r requirements.txt` en el entorno virtual activado para reflejar cualquier nueva dependencia.

Capítulo 2 — El motor de reconocimiento de voz

¿Para quién es esta herramienta?

Antes de entrar en los principios técnicos, una aclaración honesta: vosk-cli-dictation no está diseñado para competir con los motores de dictado más precisos del mercado en un ordenador de escritorio moderno. No es una debilidad oculta; es una elección de arquitectura deliberada, que hace que la herramienta sea excelente en un campo específico:

- **Hardware limitado** — Raspberry Pi, portátil antiguo, máquina sin GPU. El motor nunca necesita aceleración de hardware. En funcionamiento, una sesión utiliza aproximadamente **300 MB de RAM** (de 209 a 383 MB según el idioma y la máquina — modelos `small` medidos: `vosk-model-small-fr-0.22`, `-en-us-0.15`, `-de-0.15`, `-es-0.42`) — modesto en comparación con un navegador, pero no insignificante en una máquina con pocos recursos.
- **Reactividad en tiempo real** — el texto aparece a medida que se habla, sin esperar el final de un segmento. Esto es una propiedad de la arquitectura del motor (ver más abajo), no un ajuste que se pueda perder al cambiar un parámetro. La decodificación no se retrasa en relación con el habla: el procesamiento de un bloque de audio requiere menos tiempo que la duración del bloque mismo (factor de tiempo real RTF < 1.0), incluso en el caso más exigente: el dictado en francés en un Raspberry Pi 4 (RTF 0.73) o en un AMD A4-6210 (RTF 0.79). El francés es el idioma más costoso de decodificar; en el mismo hardware modesto, el inglés, el alemán y el español funcionan con un RTF de 0.20–0.33.
- **Nada sale de la máquina** — ningún dato de audio ni texto se transmite a un servicio remoto, durante el reconocimiento o en cualquier otro momento.
- **Una cadena de dependencias mínima** — no se requiere instalar ninguna biblioteca de aprendizaje profundo pesada, ni ningún controlador de tarjeta gráfica.

En contrapartida, la comprensión del significado sigue siendo limitada: dos palabras que se pronuncian igual pero se escriben de forma diferente («haya»/«halla», «tubo»/«tuvo») no siempre se distinguen correctamente a partir del sonido únicamente. El Capítulo 4 explica cómo corregir estos casos uno por uno, una vez identificados durante el uso; esto no es automático.

Si el objetivo es redactar textos largos con la máxima precisión en un ordenador de escritorio potente, otra herramienta será más adecuada. Si el objetivo es dictar en un dispositivo modesto, en tiempo real y sin depender de la red, vosk-cli-dictation está diseñado precisamente para eso.

Lo que hace Vosk

Vosk es el motor de reconocimiento de voz utilizado por la herramienta; es una biblioteca independiente, no un componente propio. Se basa en una arquitectura más antigua que los motores basados en transformadores (como Whisper): un modelo acústico asociado a un modelo de lenguaje estadístico, siguiendo la línea del software Kaldi.

Esta arquitectura tiene una consecuencia directa y útil: la transcripción se realiza en tiempo real, al ritmo en que el sonido llega del micrófono, con resultados que se muestran gradualmente, no en bloques de varios

segundos que habría que esperar. Esto es lo que da la sensación de una transcripción “en vivo”.

Modelo acústico y modelo de lenguaje

Cada modelo descargado (Capítulo 1) contiene dos partes, que son invisibles durante su uso pero útiles para comprender:

- El **modelo acústico** reconoce los sonidos del habla y los asocia con unidades de lenguaje (fonemas).
- El **modelo de lenguaje** evalúa qué secuencias de palabras son estadísticamente plausibles en el idioma en cuestión, para decidir entre varias interpretaciones posibles del mismo sonido.

Los modelos ofrecidos por el proyecto Vosk existen en varios tamaños. `vosk-cli-dictation` utiliza por defecto los modelos compactos (“small”), que ocupan aproximadamente 50 MB en disco, lo que representa un buen compromiso para equipos con recursos limitados. No hay nada que impida declarar un modelo más grande en la configuración (Capítulo 4) si la máquina objetivo lo permite, a cambio de una mayor huella de memoria.

Nada sale de la máquina

Ninguna etapa del reconocimiento de voz requiere conexión a la red. El único momento en que se necesita una conexión es durante la instalación inicial del paquete (descarga del motor de Python) y la descarga manual de un modelo de lenguaje; nunca durante el uso normal.

Capítulo 3 — Los tres estados: ÉCOUTE, DICTÉE, VEILLE

Vosk-cli-dictation tiene tres estados distintos, cada uno con un costo de CPU diferente. Confundirlos es la causa más común de problemas (“¿por qué la palabra ‘activación’ ya no funciona?”), y este capítulo los distingue claramente.

ÉCOUTE, DICTÉE y VEILLE son las etiquetas que el programa muestra en la pantalla. Permanecen en francés independientemente del idioma de dictado: identifican un estado interno, no un texto que se deba traducir.

Resumen

Estado	¿Cómo entrar?	Micro activo	Descifrado	Despertador
DICTÉE	Alt+H (o frase de activación), desde ÉCOUTE	Sí	Sí	— (dictado en curso)
ÉCOUTE	Alt+H , desde DICTÉE	Sí	Sí	Sí
VEILLE	Alt+W	No	No	No

DICTÉE — dictado en curso

Lo que dice es reconocido y se transcribe en la aplicación activa. Para finalizar y copiar el resultado en el portapapeles: diga la frase de finalización (“terminar dictado”), o presione **Alt+S**.

ÉCOUTE — en espera

Este es el estado predeterminado entre dos dictados. El micrófono permanece activo y el motor continúa escuchando, por una única razón: poder escuchar la frase de activación en cualquier momento. El costo de CPU de este modo no es nulo; es el precio de la capacidad de respuesta vocal.

Para el español, este costo no se reduce mediante una gramática restringida: con el modelo español, la gramática restringida decodifica aproximadamente dos veces más lento que el vocabulario completo. Por lo tanto, el motor mantiene la escucha completa, y la activación se realiza al final de las frases, bajo el control del umbral de confianza.

La activación no es simplemente una palabra: se necesita la **frase completa**, y el motor solo la recuerda si la confianza de reconocimiento es suficiente. Una palabra aislada («iniciar») nunca se interpreta como una activación; esto es una protección contra las activaciones inesperadas durante una conversación.

¿Cómo iniciar y detener un dictado?

Tres formas de controlar la transición ÉCOUTE → DICTÉE → ÉCOUTE, ajustables a través de `interaction.mode` en la configuración (Capítulo 4):

- **toggle** (por defecto, sin cambios): la descrita anteriormente: un gesto inicia (**Alt+H** o frase de activación), un gesto diferente detiene (**Alt+S** o frase de detención).

- **maintenue** (push-to-talk): el dictado dura exactamente el tiempo en que la tecla configurada bajo `hotkeys.push_to_talk` (una tecla simple, no una combinación) permanece presionada; al soltarla, la sesión se detiene y se copia automáticamente.
- **silence**: una pulsación (o la frase de activación) inicia; el dictado se detiene y se copia automáticamente después de un silencio de `interaction.silence_timeout_ms` (700 ms por defecto, ajustable por idioma), tan pronto como se ha captado al menos una palabra. Es el único de los tres modos que permite tener las manos completamente libres durante toda la frase.

En los tres modos, la opción de detención por voz y **Alt+S** / `--finalize` siguen siendo mecanismos de seguridad válidos; ningún modo elimina un método de detención existente.

VEILLE — consumo casi nulo, voluntario

Un tercer estado, activado explícitamente con **Alt+W**, nunca por voz (por diseño: el micrófono está desactivado en este estado, por lo que no se puede escuchar nada). La captura de audio y la decodificación de voz se suspenden, el costo de la CPU vuelve a ser de **0,0 %**, y el micrófono se libera.

El costo de esto: la función de activación por voz ya no funciona mientras el modo de suspensión está activo. Es una decisión deliberada, no un fallo oculto; la herramienta lo muestra explícitamente en el momento del cambio. Solo **Alt+W** permanece activo durante la suspensión (no depende del micrófono) para salir de ella.

Al cambiar al modo de espera durante un dictado, primero se detiene correctamente, como si se hubiera presionado **Alt+H**; de esta manera, no se pierde texto en medio de una frase.

El umbral de confianza determina el momento de activación

El mecanismo de “activación” solo se activa si la confianza promedio de las palabras de la frase alcanza el umbral configurado (0,55 por defecto, ajustable por idioma; ver Capítulo 4). Consecuencias concretas:

- Una frase de parada que se escucha por debajo del umbral es **absorbida**: no se escribe en el documento, ni se activa; se muestra un indicador, y basta con repetirla (o usar **Alt+S**). Este mecanismo protege el documento: una frase ordinaria que se asemeje a “terminar dictado” nunca aparecerá en él.
- Una frase de activación o de parada que se dice **dos veces seguidas** se escribe literalmente en el texto, en lugar de activarse; esto es intencionado: la repetición inmediata es rara en el control por voz, y este comportamiento evita que una repetición se pierda.

Cuándo usar el modo de espera

Útil cada vez que el micrófono debe permanecer desactivado durante un período significativo sin salir del programa: una reunión, una llamada, un momento en el que el consumo de CPU es importante (en un portátil con batería, en un hardware con limitaciones). En una máquina donde el consumo de CPU no es importante, permanecer en ÉCOUTE de forma permanente es perfectamente razonable: es una elección, no una obligación de optimizar.

El desarrollo de una sesión completa

1. **Lanzar** la herramienta en una terminal visible. Se inicia en ÉCOUTE.
2. **Abrir la aplicación objetivo** (editor, navegador, terminal) y darle el foco del teclado.
3. **Pasar a DICTÉE**: decir la frase de activación, o pulsar **Alt+H**.
4. **Hablar** normalmente. El texto se escribe en tiempo real en la aplicación, y los signos de puntuación («coma», «exclamación», «nueva línea»...) se interpretan sobre la marcha (Capítulo 4).
5. **Pausar** si es necesario: **Alt+H** vuelve a ÉCOUTE sin copiar nada, el texto ya escrito permanece en la aplicación. Pulsar **Alt+H** de nuevo para recomenzar.
6. **Terminar**: decir la frase de finalización, o **Alt+S**, o enlazar `--finalize` a un atajo del escritorio (Capítulo 5). El texto completo de la sesión se coloca entonces en el portapapeles.
7. **Pegar** (Ctrl+V) donde se quiera — el portapapeles original se restaura justo después.
8. **Poner en suspensión** entre usos: **Alt+W** (micrófono desactivado, 0 % CPU). Pulsar **Alt+W** de nuevo para reactivar.
9. **Salir**: Ctrl+C en la terminal de lanzamiento.

Capítulo 4 — Configuración y personalización

Todo el comportamiento de `vosk-cli-dictation` se controla a través de un único archivo YAML, `config.yaml`. Este capítulo explica dónde se encuentra y qué se puede modificar en él.

¿Dónde se encuentra el archivo de configuración?

La ubicación depende del canal de instalación (Capítulo 1) y sigue un orden de prioridad preciso: la primera ubicación disponible es la que se utiliza, y las siguientes nunca se consultan.

1. La variable de entorno `VOSK_CONFIG`, si está definida, se utiliza tal cual, y nunca se crea automáticamente.
2. `config/config.yaml` del repositorio, en modo clon Git.
3. `~/.config/vosk-cli-dictation/config.yaml`, en la instalación por paquete, se crea automáticamente en el primer lanzamiento, a partir de una plantilla, con un mensaje que indica dónde se ha creado el archivo.
4. La plantilla en sí misma, es de solo lectura.

En la práctica: con el paquete Debian, modificar `~/.config/vosk-cli-dictation/config.yaml` — es un archivo personal, que nunca es sobrescrito por una actualización del paquete.

Elegir el idioma de la plantilla de configuración

El paquete Debian incluye cuatro plantillas localizadas: `config.en.yaml`, `config.fr.yaml`, `config.de.yaml`, `config.es.yaml`. Se encuentran en `/usr/share/vosk-cli-dictation/`.

En el primer lanzamiento, el que corresponde a la configuración regional del sistema (`LANGUAGE` y luego `LANG`, se recurre al inglés si ninguno de los cuatro coincide) se copia a `~/.config/vosk-cli-dictation/config.yaml`.

Para cambiar a otra plantilla después de este primer lanzamiento, identifíquelas con `dpkg -L vosk-cli-dictation | grep config` y luego reemplace su archivo personal con el deseado:

```
cp /usr/share/vosk-cli-dictation/config.de.yaml ~/.config/vosk-cli-dictation/config.yaml
```

En modo clonación de Git, las cuatro plantillas se encuentran directamente en `config/` del repositorio.

Personalizar los atajos de teclado

Sección `hotkeys`: del archivo. Cada atajo se define por una lista de teclas `pynput_keys` y un nombre mostrado `display_name`:

```
hotkeys:
  toggle_recording:
    pynput_keys: ["alt_l", "h"]
    display_name: "[Alt+H]"
```

Tres atajos son configurables aquí — `toggle_recording` (Capítulo 1), `finalize_session` y `sleep_toggle`

(Capítulo 3). **Ctrl+C**, que cierra el programa, no forma parte de ellos: es la señal de interrupción estándar del sistema, siempre activa, y nunca se puede reconfigurar desde este archivo.

Bajo Wayland, estos atajos solo funcionan para las ventanas que pasan por XWayland (ver Capítulo 5). Para las ventanas nativas de Wayland, son el entorno de escritorio el que gestiona los atajos, a través de subcomandos de control.

Una cuarta entrada, `push_to_talk`, solo cuenta con una tecla simple (no una combinación): sirve para el modo “push-to-talk” (Capítulo 3), y no tiene efecto en los otros dos modos.

Elegir cómo comienza y termina la dictación

Sección `interaction`: del archivo (Capítulo 3 para la presentación de los tres modos):

```
interaction:
  mode: toggle          # toggle | maintain | silence
  silence_timeout_ms: 700
```

`silence_timeout_ms` se configura por idioma exactamente como el umbral de confianza a continuación, bajo `language_settings.<lang>.silence_timeout_ms` — útil si un modelo decodifica más lentamente en una máquina determinada y activa un tiempo de espera demasiado corto.

Puntuación: dos modos de entrada, por idioma

Cada idioma define, según `language_settings.<lang>.punctuation`, su lista de signos de puntuación y su método de entrada:

- **mode: inline** (por defecto) — las palabras de puntuación se reconocen y se reemplazan directamente durante la dictado: decir «coma» inserta una coma.
- **mode: explicit** — un comando de puntuación solo se aplica si está inmediatamente precedido por la palabra disparadora del idioma: es necesario decir «puntuación coma» para obtener una coma, mientras que «coma» solo se escribe completamente. Útil cuando el idioma o el vocabulario provoca falsos disparos con frecuencia.

La lista real de palabras (signos de puntuación simples, dobles, saltos de línea, caracteres especiales —guion, paréntesis, comillas...), se muestra por el programa al inicio; está definida en las secciones `simple`, `double`, `layout` y `special`. La tipografía aplicada al texto (las comillas rectas “...” y el hecho de no insertar un espacio antes ? ! : ;...) se configura mediante el bloque `typography` del idioma.

Ajustar el umbral de confianza para la activación

El encendido y el apagado solo se mantienen si la confianza promedio de las palabras de la frase alcanza el umbral `recognition.confidence_threshold` (0,55 por defecto, Capítulo 3). Se puede ajustar por idioma:

```
language_settings:
  es:
    confidence_threshold: 0.55
```

Un umbral más alto hace que el sistema sea más estricto (menos falsos positivos, pero la frase de parada se interrumpe con mayor frecuencia); un umbral más bajo hace lo contrario.

Agregar vocabulario y corregir confusiones

Dos mecanismos distintos, según `language_settings.<lang>`:

- **custom_vocabulary** — correcciones sensibles a mayúsculas y minúsculas, aplicadas después de todo lo demás. Útil para abreviaturas o nombres propios que el motor escribe sistemáticamente en minúsculas:

```
custom_vocabulary:  
  "ia": "IA"
```

- **recognition_aliases** — corrige una confusión fonética recurrente del motor (dos palabras que se pronuncian casi igual, Capítulo 2). Por ejemplo, si el motor reconoce consistentemente «halla» en lugar de «haya»:

```
recognition_aliases:  
  "halla": "haya"
```

Estos dos ajustes se descubren con el uso, no de antemano: dictar, identificar una confusión recurrente y añadirla.

¿Dónde se buscan los modelos de lenguaje?

Como en la configuración, la búsqueda sigue un orden preciso, siendo el primer directorio encontrado en el disco el que tiene prioridad.

1. Una ruta absoluta — a pesar de su nombre, la clave `path_relative` también puede contener una ruta absoluta completa; en este caso, tiene prioridad sobre todo lo demás mencionado a continuación.
2. `~/.local/share/vosk-models/` (o `$XDG_DATA_HOME/vosk-models/` si esta variable está definida).
3. `/usr/share/vosk-models/` — plantillas instaladas a nivel del sistema.
4. `vosk-model/` en la raíz del proyecto, solo en modo clon Git.

Si no se encuentra ningún modelo, el mensaje que se muestra al inicio enumera precisamente estas ubicaciones y la dirección de descarga, y recuerda el comando `vosk-cli-dictation --download-model` — nunca un fallo sin explicación. Este comando instala el modelo pequeño en la ubicación 2 mencionada anteriormente (`~/.local/share/vosk-models/`), el único lugar donde un modelo recuperado sin privilegios de administrador puede encontrarse posteriormente.

Capítulo 5 — Integración del sistema: Wayland y X11

`vosk-cli-dictation` no tiene una interfaz gráfica propia; en lugar de eso, escribe el texto reconocido directamente en la aplicación que tenga el foco del teclado, sea cual sea. Este capítulo explica cómo funciona y qué varía según el sistema de visualización que se utilice.

¿Cómo llega el texto a la aplicación?

A diferencia de lo que uno podría imaginar, el texto **no se escribe carácter por carácter**. El mecanismo real es el siguiente: el texto reconocido se coloca en el portapapeles, y luego se simula un atajo de teclado para “pegar” (Ctrl+V). Esto es más fiable y más rápido que simular una escritura completa, especialmente en pasajes largos.

El portapapeles original se restaura inmediatamente después de este pegado (Ctrl+V); por lo tanto, la función de dictado no sobrescribe lo que se había copiado previamente por más de un instante. En Wayland con una aplicación GTK, esta restauración puede ocurrir antes de que la ventana haya terminado de leer el portapapeles, lo que pega el contenido anterior; la configuración `clipboard.defer_restore: true` retrasa la restauración hasta el final de la sesión de dictado y elimina este riesgo.

El micrófono

`vosk-cli-dictation` captura el sonido del **dispositivo de entrada predeterminado del sistema** (PulseAudio o PipeWire). No hay ninguna configuración de dispositivo en `config.yaml`: para cambiar de micrófono, se define el dispositivo correcto como entrada predeterminada, en la configuración de sonido del escritorio, con `pavucontrol`, o en la línea de comandos (`pactl set-default-source <nom>`, la lista se proporciona con `pactl list sources short` o `arecord -l`).

`vosk-cli-dictation --diagnose` abre un flujo de prueba en este dispositivo y indica si la apertura falla: micrófono ausente, desconectado o ya utilizado por otra aplicación.

Las herramientas del sistema utilizadas, y su selección automática

La inyección requiere **dos** herramientas: un simulador de escritura y una herramienta de portapapeles. Se seleccionan automáticamente según el tipo de sesión, sin necesidad de configuración:

- **Bajo Wayland:** `wtype` (entrada de texto) y `wl-clipboard` (portapapeles), que utilizan el protocolo Wayland de forma nativa. Si `wtype` no está presente, la entrada de texto pasa por `ydotool` — pero `wl-clipboard` sigue siendo necesario.
- **Bajo X11:** `xdotool` (entrada de texto) y `xclip` (portapapeles).

`ydotool` es el último recurso para la entrada de texto (funciona en ambos mundos a través de `/dev/uinput`, pero depende de la disposición del teclado para sus códigos de tecla). Al menos un simulador de entrada de texto debe estar presente; de lo contrario, la herramienta se niega a iniciar y lo indica claramente. El paquete Debian depende de `wl-clipboard` y recomienda `xclip`; en el clon de Git, instálalos manualmente. Sin una herramienta de portapapeles, el texto dictado nunca llega al portapapeles: nada se escribe en la

aplicación, y un `Ctrl+V` pega el último contenido copiado manualmente. El comando `vosk-cli-dictation --diagnose` verifica el entorno (tipo de sesión, herramientas, modelos, micrófono, portapapeles, socket de control) y muestra lo que falta.

Activar ydotool (solución de respaldo)

Esta sección solo se aplica a los sistemas que recurren a `ydotool`, por falta de `wtype` (Wayland) o de `xdotool` (X11). Cuando `wtype` o `xdotool` están disponibles, nada de lo que sigue es necesario; en particular, es inútil unirse al grupo `input`. El paquete Debian aplica la misma regla: su mensaje de instalación solo ofrece añadir al grupo `input` si se recurre a la alternativa `ydotool`, y especifica que `xdotool` y `wtype` no lo necesitan.

Tres condiciones, que deben verificarse en orden si el texto no se escribe:

1. **El paquete `ydotool` está instalado.** En Debian, se encuentra en `trixie-backports`, no en la rama estable; active este repositorio si es necesario antes de la instalación.
2. **El servicio `ydotool` está en funcionamiento.** `ydotool` necesita este demonio en segundo plano para funcionar.
3. **El usuario pertenece al grupo `input`.** Es este grupo el que da acceso a `/dev/uinput`, de lo contrario, `ydotool` fallará silenciosamente para este usuario:

```
sudo usermod -aG input $USER
```

Es necesario cerrar sesión y volver a iniciarla para que este cambio de grupo tenga efecto; un simple reinicio de la terminal no es suficiente.

Atajos de teclado: X11 y Wayland no son iguales

Los atajos integrados `Alt+H`, `Alt+S`, `Alt+W` son detectados por la propia herramienta a través de la biblioteca `pynput`, que observa el servidor X. Bajo Wayland, solo detecta las ventanas que pasan por `XWayland` (muchas aplicaciones Electron y antiguas), nunca una ventana **Wayland nativa**: para estas últimas, son los atajos del entorno de escritorio los que deben controlar la herramienta.

Cada entorno de escritorio puede asociar una combinación de teclas a un comando. Por el lado de `vosk-cli-dictation`, los subcomandos de control que se pueden utilizar son:

Acción deseada	Subcomando
Iniciar/pausar el dictado	<code>vosk-cli-dictation --toggle</code>
Terminar el dictado y copiar el texto	<code>vosk-cli-dictation --finalize</code>
Entrar/salir de la noche	<code>vosk-cli-dictation --sleep-toggle</code>
Conocer el estado actual	<code>vosk-cli-dictation --status</code>
Cancelar el dictado en curso sin escribir nada	<code>vosk-cli-dictation --cancel</code>
Finalizar la sesión correctamente	<code>vosk-cli-dictation --stop</code>

Ejemplo en KDE Plasma: en *Configuración del sistema* → *Atajos* → *Atajos personalizados*, cree una nueva entrada llamada “Comando/URL” cuyo comando sea `vosk-cli-dictation --finalize`, y asígnele una combinación (por ejemplo, **Meta+S**). La subcomando se conecta a la sesión actual, ejecuta la acción y muestra el estado resultante (OK DICTÉE, OK ÉCOUTE, OK VEILLE).

Bajo X11, estas subcomandos son la forma más confiable: un atajo `pynput` no es consumido; la combinación alcanza también la ventana objetivo, y al ser presionada en un terminal, inserta una secuencia parásita (por ejemplo, `^[s` para **Alt+S**). Vincular `--finalize` (o `--toggle`, `--sleep-toggle`) a un atajo del escritorio proporciona una salida limpia desde cualquier ventana.

¿Qué sucede si no hay ninguna herramienta de inyección disponible?

La herramienta no se inicia y lo indica explícitamente, con una sugerencia de instalación. No hay un modo “degradado” “sin inyección”: sin `wtype/xdotool/xdotool`, no existe ninguna forma fiable de escribir el texto reconocido en otra aplicación.

Esta verificación inicial solo se aplica al simulador de escritura. Una herramienta de portapapeles faltante (`wl-clipboard` bajo Wayland, `xclip` bajo X11) no impide el inicio, pero interrumpe la función de dictado: el texto dictado solo aparece en la terminal y cada `Ctrl+V` pega el último contenido copiado manualmente. `vosk-cli-dictation --diagnose` lo indica.

Capítulo 6 — Solución de problemas

Síntoma → causa probable → acción

Síntoma	Causa probable	Acción
El programa se niega a iniciarse, aparece un mensaje en las herramientas de inyección	Ninguna herramienta de inyección (wtype/xdotool/ydotool) está instalada o funcional	Instalar uno de ellos (ver Capítulo 5)
“No se pudo cargar ningún modelo de reconocimiento de voz” al inicio	No se encontró ningún modelo en las ubicaciones esperadas	El mensaje enumera las ubicaciones buscadas y la URL de descarga; consulte el Capítulo 1
Nada se reconoce, parece que el micrófono está silenciado	Dispositivo de entrada predeterminado incorrecto, o micrófono silenciado a nivel del sistema	Verificar / cambiar la configuración predeterminada (ajustes de sonido, pavucontrol); vosk-cli-dictation --diagnose — ver Capítulo 5
--diagnose indica el fallo al abrir el flujo de audio	El dispositivo está siendo utilizado por otra aplicación, o no se tienen los permisos de acceso necesarios para el micrófono	Cerrar la otra aplicación; verificar los permisos del dispositivo (ver Capítulo 5)
Nada se escribe en la aplicación, pero no se muestra ningún error	La herramienta se ha plegado sobre ydotool, pero el usuario no está en el grupo input	Verificar la pertenencia al grupo, volver a conectarse: véase el Capítulo 5
La aplicación siempre recupera el último contenido copiado manualmente; el texto dictado solo aparece en la terminal	No se ha instalado ninguna herramienta de portapapeles (wl-clipboard en Wayland, xclip en X11); el texto dictado nunca llega al portapapeles	Instálalo. El paquete Debian ≥ 0.5.1 depende de wl-clipboard y recomienda xclip; en una copia clonada de Git, instálalos manualmente; consulta el Capítulo 5
Bajo Wayland, con una aplicación GTK, la ventana pega instantáneamente el contenido anterior del portapapeles	La restauración del portapapeles ocurre antes de que la ventana termine de leer la selección	clipboard.defer_restore: true en la configuración; en este caso, la restauración se pospone hasta el final de la sesión; véase el Capítulo 5
Ningún sonido de notificación	“paplay ausente, o ningún archivo de sonido configurado.”	Advertencia menor, sin impacto en el reconocimiento — instalar pulseaudio-utils si se desea el sonido

Síntoma	Causa probable	Acción
¿Puede una palabra aislada («iniciar») desencadenar accidentalmente el dictado?	No: la activación requiere la frase completa	El despertar solo se produce con la totalidad de “iniciar dictado”; no hay nada que corregir
La frase de activación no inicia la función de dictado	Modo de espera activo (Alt+W), o frase incompleta, o nivel de confianza por debajo del umbral	Salir del modo de espera con Alt+W; pronunciar la frase completa; repetir si es necesario — ver Capítulo 3
«terminar dictado» no deja de dictar, y nada se escribe	Frase de parada por debajo del umbral de confianza — absorbida por seguridad	Repítala, o use Alt+S / --finalize — véase el Capítulo 3
La frase de activación o de desactivación se escribe en el texto en lugar de activarse	Repetir una frase dos veces seguidas — comportamiento deseado	Decirla una sola vez, tras un breve silencio (ver Capítulo 3)
Alt+H / Alt+S / Alt+W no responden	Ventana Wayland nativa en primer plano — pynput solo muestra ventanas X11 y XWayland	Vincular --toggle, --finalize, --sleep-toggle a los atajos del escritorio; consulte el Capítulo 5
Alt+S inserta ^[s en el terminal	Comportamiento conocido: el acceso directo no se consume por la herramienta	Preferir un acceso directo en el escritorio vinculado a --finalize — ver Capítulo 5
Alt+H no copia nada al portapapeles	Comportamiento normal: Alt+H pausa la ejecución, pero no la finaliza	Utilizar Alt+S o la frase de parada para copiar; ver Capítulo 1
Una palabra se reconoce sistemáticamente de forma incorrecta	Confusión fonética del motor (palabras que suenan parecido)	Agregar una corrección en recognition_aliases — ver Capítulo 4
La herramienta finaliza la sesión automáticamente: “el motor de reconocimiento no se reinicia”	El decodificador ha devuelto varias veces tramas ilegibles consecutivamente, y la reconstrucción del motor no ha cambiado nada	Reiniciar la herramienta. Si esto se repite, adjuntar al informe el archivo que se encuentra en ~/.local/state/vosk-cli-dictation/diagnostics/ (ver más abajo)
La configuración modificada parece que nunca se aplica	El archivo editado no es el que se está utilizando realmente (existen varias ubicaciones posibles)	Verificar el orden de resolución; véase el Capítulo 4

¿Dónde buscar un mensaje de error completo?

vosk-cli-dictation muestra sus errores directamente en el terminal donde se ejecutó. Reiniciar desde un terminal visible (no en segundo plano) es lo primero que se debe hacer para diagnosticar un problema

no listado anteriormente. Si hay dudas sobre el entorno, `vosk-cli-dictation --diagnose` verifica todo (herramientas, modelos, micrófono, portapapeles, socket de control) y muestra el estado de cada elemento.

Un solo caso escribe un archivo: cuando el decodificador devuelve una salida ilegible, la cadena bruta se guarda una vez por sesión en `~/.local/state/vosk-cli-dictation/diagnostics/`. Este archivo es útil para adjuntarlo a un informe de problema de reconocimiento.

Si el problema persiste

El repositorio del proyecto tiene una página pública de seguimiento de problemas: <https://github.com/RonanDavalan/vosk-cli-dictation/issues>