

WhispSkrid — Manual del usuario

Instalar y usar el dictado por voz sin conexión con pulsar para hablar

Version v1.0.1

2026-09-18

Índice general

Capítulo 1 — Instalación	2
En dos minutos	2
Requisitos previos	2
Dos formas de instalar	3
Recuperar un modelo	4
Primer arranque	4
Una primera dictada	4
Desinstalar y actualizar	5
Capítulo 2 — Cómo usar	6
¿Para quién es esta herramienta?	6
El modelo de interacción: apoyo para hablar estricto	6
Dos formas de iniciar una dictada	7
Sesión persistente y cliente	7
Configuración	8
Gestionar los modelos	9
Resumen de la línea de comandos	10
Capítulo 3 — Solución de problemas	11
¿Cómo llega el texto a la aplicación?	11
El micrófono	11
Las herramientas de inyección y su selección automática	11
Atajos: X11 y Wayland no son iguales	12
--diagnose	12
Síntoma → causa probable → acción	12
¿Dónde buscar un mensaje de error?	14
Si el problema persiste	14

Capítulo 1 — Instalación

En dos minutos

Para aquellos que quieren dictar inmediatamente, sin leer el resto.

1. Instale el paquete nativo de su distribución.

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid_1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

2. Recuperar un modelo Whisper, si no está presente (el paquete no lo incluye, ya que es demasiado grande):

```
whispskrid --download-model base
```

El modelo base se descarga, se verifica y luego se guarda en `~/.local/share/whispskrid/whisper-models/`. En el primer lanzamiento, sin ningún modelo, la herramienta ofrece la misma descarga de forma interactiva.

Lanzar la herramienta.

```
whispskrid
```

Muestra un indicador de “listo” y espera. Mantén presionada la tecla **Mayús derecha**, habla, suelta: el texto transcrito se escribe en la ventana activa.

Esto es todo para empezar. El resto explica cómo instalar desde el código fuente, cambiar de modelo o idioma, y cómo usar la herramienta con el teclado del escritorio (Capítulo 2).

Requisitos previos

- **Un sistema Linux** con una sesión gráfica X11 o Wayland. Sin un servidor de pantalla, no hay una ventana activa donde escribir el texto transcrito.
- **Python 3** y un entorno virtual (`python3-venv`).
- **Una herramienta de inyección de texto**: `ydotool` o `xdotool` para la escritura, `wl-clipboard` bajo Wayland o `xclip` bajo X11 para el portapapeles. Si no hay un motor de escritura funcional, la herramienta no se niega a iniciarse: pasa a un modo degradado (el texto transcrito permanece visible en el terminal, sin inyección - Capítulo 3).
- **Un micrófono reconocido por el sistema** como dispositivo de entrada predeterminado (Capítulo 3).

- **Espacio en disco para el modelo:** aproximadamente 145 MB para el modelo base, más para modelos más grandes (Capítulo 2). En funcionamiento, una sesión mantiene el modelo en memoria - aproximadamente 150 a 300 MB de RAM para base.

El paquete nativo declara e instala todas sus dependencias, tanto del sistema como de Python, independientemente de la distribución. En modo “clone” de Git, `.venv/bin/pip install -e .` instala las bibliotecas de Python; la herramienta de inyección (`ydotool` o `xdotool`, más `wl-clipboard` o `xclip`) debe instalarse por separado.

Dos formas de instalar

WhispSkrid ofrece dos canales, y ninguno de los dos reemplaza al otro.

El paquete nativo (recomendado para uso común)

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid-1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

El paquete instala un comando único, `whispskrid`, en `/usr/bin/`. También crea un entorno Python aislado e instala el motor de transcripción (`faster-whisper`) en él; este paso requiere conexión a internet durante la instalación, solo una vez. No se requiere ningún permiso de superusuario después.

Cada paquete ha sido instalado, verificado por `whispskrid --diagnose` y luego desinstalado correctamente en un contenedor de cada una de las versiones anteriores (Debian 12 y 13, Ubuntu 22.04 y 24.04, Fedora 42, openSUSE Leap 15.6, Arch Linux). El paquete de Debian también se ha validado en uso real, con dictado incluido, en varias máquinas físicas, incluyendo una con Debian 13; los demás paquetes aún no se han validado en una sesión gráfica real. Los demás sistemas no están cubiertos; no por principio, simplemente el alcance que se ha probado.

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 al iniciarse por primera vez y al descargar un modelo, 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/whispskrid.git
cd whispskrid
python3 -m venv .venv
```

```
.venv/bin/pip install -e .  
.venv/bin/whispskrid --diagnose
```

En este modo, la herramienta se inicia mediante `.venv/bin/whispskrid` en lugar de mediante el comando `whispskrid` del paquete, y su configuración se encuentra en `config/config.yaml`, junto con el código; nunca en el directorio personal. Los dos modos no se mezclan: un clon de Git ignora la configuración del paquete, y viceversa.

Recuperar un modelo

Ninguno de los dos canales incluye un modelo de transcripción; estos pesan alrededor de 75 MB (`tiny`) hasta varios gigabytes (`large-v3`), lo cual es demasiado para incluirse con la herramienta.

La forma más sencilla, en la instalación por paquetes, es dejar que la herramienta se encargue de ello:

```
whispskrid --download-model base
```

Este comando recupera el modelo solicitado — `base` por defecto, o uno de `tiny`, `small`, `medium`, `large-v3` (y sus variantes `.en`, especializadas en inglés) — verifica que se cargue, lo instala en `~/.local/share/whispskrid/whisper-models/`, y luego se detiene. En el primer lanzamiento sin ningún modelo, se presenta la misma opción de forma interactiva.

El modelo y el idioma son dos opciones distintas: un modelo Whisper es multilingüe, es la opción `-l` (o `default_language` en la configuración, Capítulo 2) la que establece el idioma de reconocimiento. Sin ella, Whisper detecta el idioma por sí solo.

Si no se encuentra ningún modelo al inicio, la herramienta muestra las ubicaciones buscadas y los pasos a seguir; nunca un fallo silencioso.

Primer arranque

```
whispskrid
```

(en modo clone de Git, el mismo comando es `.venv/bin/whispskrid`, desde la raíz del repositorio).

Al iniciarse, la herramienta carga la configuración, carga el modelo una sola vez, abre el micrófono y luego muestra una barra de estado “listo” y espera. Este proceso permanece en primer plano: es la sesión persistente. Mantiene el modelo en memoria para que cada dictado comience sin tiempo de carga (Capítulo 2). Solo puede haber una sesión activa a la vez.

Para salir: **Ctrl+C** en el terminal de lanzamiento, o `whispskrid --stop` desde otro terminal.

Una primera dictada

WhispSkrid funciona con la función de “presionar para hablar”: se mantiene presionada una tecla mientras se habla y se suelta cuando se termina.

1. Dar el foco del teclado a la aplicación objetivo (editor, navegador, terminal).

2. **Mantener presionada la tecla Shift derecha.** Un breve sonido confirma el inicio de la grabación.
3. **Hablar** normalmente, mientras se mantiene presionada la tecla.
4. **Soltar.** El audio grabado se transcribe; el texto obtenido se inserta en la ventana activa.

No hay frase de activación, ni estado de espera, ni temporizador: la captura dura exactamente el tiempo que se mantiene presionada la tecla. Cada vez que se suelta la tecla, se genera una inyección independiente.

Bajo una sesión **Wayland**, esta tecla solo se captura para las ventanas que pasan por XWayland; para una ventana Wayland nativa, es necesario vincular un comando de control a un acceso directo del escritorio (Capítulo 2).

Desinstalar y actualizar

Paquete nativo. `sudo apt remove whispskrid` (Debian), `sudo dnf remove whispskrid` (Fedora), `sudo zypper remove whispskrid` (openSUSE) ou `sudo pacman -R whispskrid` (Arch) desinstala la herramienta y el entorno de Python aislado (`/usr/lib/whispskrid/venv`). En Debian, `sudo apt purge whispskrid` también elimina la configuración predeterminada. El archivo de configuración personal (`~/.config/whispskrid/config.yaml`) y las plantillas (`~/.local/share/whispskrid/whisper-models/`) nunca se ven afectados por la desinstalación, en ninguna distribución; se pueden eliminar manualmente si se desea.

Clonar Git. Eliminar el directorio clonado: contiene todo (código, entorno virtual `venv/`, posibles modelos en `whisper-models/`). Nada se guarda en otro lugar en este modo.

Actualización. Con el paquete, instala el archivo más reciente usando el mismo comando que para la instalación (`apt install`, `dnf install`, `zypper install` ou `pacman -U`): reemplaza la versión existente sin modificar la configuración personal. Con el clon de Git, haz `git pull`, y luego ejecuta de nuevo `.venv/bin/pip install -e .` para reflejar cualquier nueva dependencia.

Capítulo 2 — Cómo usar

¿Para quién es esta herramienta?

WhispSkrid es una herramienta de dictado por línea de comandos y sin conexión, construida alrededor de Whisper. Su objetivo es un uso simple y predecible: se mantiene presionada una tecla, se habla, se suelta la tecla, y el texto se escribe en la ventana activa. No hay nada más: ni frases de activación, ni comandos de voz, ni estados de espera que gestionar.

La transcripción es proporcionada por `faster-whisper`, una implementación de Whisper basada en `CTranslate2`, optimizada para funcionar en procesador. El modelo predeterminado es `base`, un buen equilibrio entre velocidad y precisión para la mayoría de las máquinas; se puede cambiar en cualquier momento (ver más abajo). Toda la transcripción se realiza localmente: ningún sonido ni texto abandona la máquina, en ningún momento.

El modelo de interacción: apoyo para hablar estricto

Una dictada es un episodio aislado, sin conexión o continuidad entre un episodio y otro:

1. **Sostener** — Presionar y mantener la tecla de “sostener para hablar”. La captura de audio comienza; un breve sonido lo confirma.
2. **Hablar** — El audio se acumula mientras se mantiene la tecla presionada. No hay corte automático, ni temporizador, ni detección de fin de frase.
3. **Liberar** — La captura se detiene. El búfer de audio completo se transcribe de una sola vez.
4. **Insertar** — El texto transcrito, después de un ligero post-procesamiento, se inserta en la ventana activa (Capítulo 3 para el mecanismo).

No hay texto “acumulado” entre dos dictados, por lo tanto, no hay un comando para “copiar la sesión”: cada ejecución genera su propia inyección.

El límite de tiempo

Única parada automática: `capture.max_seconds` (300 segundos por defecto). Si la tecla se mantiene presionada por más tiempo, la grabación se detiene automáticamente, lo que se ha grabado se transcribe e inyecta normalmente, y se registra una advertencia. No es un temporizador de conveniencia; es una protección contra una tecla atascada o un buffer de audio que se infla indefinidamente.

El silencio automático no está activado

La función de detección automática de silencio (VAD) aún no está disponible. Las claves `vad.enabled` (por defecto `false`) y `vad.silence_ms` ya están presentes en la configuración; `vad.enabled: true` muestra una advertencia “no implementado aún” y la herramienta continúa funcionando en modo de “hablar al presionar”.

Dos formas de iniciar una dictada

La tecla local

Un detector de atajos (pynput) observa el teclado. La tecla por defecto es **Shift derecho** (`shift_r`), sin un atajo global por defecto en los escritorios comunes y fácil de mantener. Se puede cambiar en `hotkeys.push_to_talk` en la configuración. En esta configuración, el mantenimiento es nativo: la captura dura exactamente desde el momento en que se presiona hasta que se suelta.

pynput observe el servidor X. Bajo Wayland, solo captura las ventanas que pasan por XWayland, nunca una ventana Wayland nativa; es una funcionalidad de “mejor esfuerzo”. Para evitar que se inicie este monitor, establecer `hotkeys.pynput_enabled: false` y controlar mediante la vía indicada a continuación.

Los subcomandos de control

Vía universal, independiente del tipo de sesión: se asocian estos subcomandos a los atajos globales del escritorio. Como un atajo de escritorio no transmite la noción de “tecla presionada / tecla liberada”, esta vía funciona mediante un interruptor:

Acción	Subcomando
Iniciar la captura	<code>whispskrid --dictate</code>
Detener la captura e inyectar	<code>whispskrid --dictate-stop</code>
Iniciar o detener según el estado	<code>whispskrid --toggle</code>
Cancelar la captura actual sin inyectar	<code>whispskrid --cancel</code>
Conocer el estado actual	<code>whispskrid --status</code>
Finalizar la sesión actual	<code>whispskrid --stop</code>

Ejemplo en KDE Plasma: en *Configuración del sistema* → *Atajos* → *Atajos personalizados*, crear una entrada con “Comando/URL” cuyo comando sea `whispskrid --toggle`, y asignarle una combinación de teclas. Una primera pulsación inicia la captura, una segunda la detiene e inyecta.

Las dos vías controlan el mismo estado interno. Una captura iniciada con un botón puede ser cancelada por `--cancel`, y viceversa.

Sesión persistente y cliente

`whispskrid` lanzar sin argumentos de control inicia una **sesión persistente** en primer plano: carga la configuración y el modelo, abre el micrófono, inicia el detector de teclas y el socket de control, y luego espera.

`whispskrid` se lanzó con un argumento de control (`--dictate`, `--status`, `--stop...`) y actuó como **cliente**: se conecta a la sesión en curso a través de un socket Unix (en `$XDG_RUNTIME_DIR`), envía el comando, muestra la respuesta (OK o ERR) y termina. Si no hay una sesión en curso, lo indica y termina con un error.

Una única sesión persistente puede estar activa a la vez: si el socket ya está respondiendo, una nueva instancia se negará a iniciarse.

Configuración

Todo el comportamiento se define a través de un único archivo YAML, `config.yaml`.

¿Dónde está el archivo?

La ubicación sigue un orden de prioridad: la primera que existe tiene prioridad, y las siguientes nunca se consultan.

1. La variable de entorno `WHISPSKRID_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/whispskrid/config.yaml`, en la instalación por paquete: se crea automáticamente al primer lanzamiento a partir de una plantilla, con un mensaje que indica dónde se ha creado el archivo.
4. La plantilla en sí, es de solo lectura.

En la práctica, sea cual sea el paquete nativo instalado, se modifica `~/.config/whispskrid/config.yaml` — archivo personal, que nunca es sobrescrito por una actualización del paquete.

Cambiar de modelo

```
models:
  default: "base"
  device: "auto"
  compute_type: "auto"
```

`models.default` es un nombre corto resuelto en el directorio de modelos, o una ruta absoluta. `device` tiene un valor de `cpu`, `cuda` o `auto`.

`compute_type` acepta cuatro valores: `int8`, `int8_float16`, `float16` y `auto`. Con `auto`, la elección se realiza automáticamente: `int8` en el procesador, `float16` en la tarjeta gráfica.

`--model NOM` en la línea de comandos sobrescribe `models.default` para una sesión. Un modelo más grande transcribe con mayor precisión, a costa de una huella de memoria y una latencia más elevadas.

Forzar el idioma

```
default_language: ""
```

Vide, la interfaz toma el idioma del sistema y el reconocimiento detecta el idioma por sí solo. `-l fr` (o `--lang fr`) en la línea de comandos fuerza el idioma de la interfaz y del reconocimiento para una sesión, sin modificar el archivo.

El botón de “hablar”

```
hotkeys:
  pynput_enabled: true
  push_to_talk: ["shift_r"]
  min_hold_ms: 250
```

```
mode: "hold"
```

Una combinación de funciones también es válida ([`"f4"`], protegida de los atajos del escritorio). Listar varias funciones las convierte en una **combinación**: todas deben mantenerse presionadas para iniciar la dictado, y en modo `hold`, soltar una de ellas detiene y inyecta. `min_hold_ms` cancela sin transcribir un toque más breve; evita un roce que abriría una grabación en silencio, que Whisper podría interpretar erróneamente; sin efecto fuera del modo `hold`. `mode: "toggle"` cambia la semántica: un toque inicia la grabación, el siguiente la detiene e inyecta, soltarla ya no hace nada. `mode: "armed"` (escucha de frases vocales) requiere modelos no proporcionados: activar falla correctamente mientras están ausentes.

El límite de tiempo y el procesamiento posterior

```
capture:
```

```
  max_seconds: 300
```

```
post_processing:
```

```
  trim: true
```

```
  capitalize_sentence_start: true
```

`trim` Recorta los bordes en blanco del texto transcrito. `capitalize_sentence_start` Fuerza una mayúscula en la primera letra insertada. Whisper realiza la puntuación y la capitalización de la primera letra de cada frase por sí mismo: el post-procesamiento se limita a estos dos ajustes; no hay ninguna tabla de sustitución, regla por idioma, ni inserción de espacios finos.

El collage en Wayland

```
clipboard:
```

```
  defer_restore: false
```

Pasé a `true`, el portapapeles solo se restaura al final de la sesión en lugar de restaurarse después de cada dictado. Útil con algunas aplicaciones GTK bajo Wayland (Capítulo 3).

Gestionar los modelos

Los modelos se almacenan en una carpeta gestionada:

- instalado mediante paquete: `/usr/share/whispskrid/whisper-models/` (solo lectura), complementado con `~/.local/share/whispskrid/whisper-models/` para las descargas del usuario;
- desde el código fuente: `whisper-models/` en la raíz del repositorio.

`whispskrid --download-model [NOM]` Descarga un modelo en la carpeta de usuario y verifica que se cargue. Orden de magnitud de la huella en disco en cuantificación `int8`:

<code>tiny</code>	75 Mo
<code>base</code>	145 Mo
<code>small</code>	480 Mo
<code>medium</code>	1,5 Go
<code>large-v3</code>	3 Go

En memoria RAM, en el pico de uso: en CPU tiny ~0.4 GB, base ~0.5–0.6 GB, small ~1.3 GB, medium ~3.5–3.7 GB, large-v3 ~6.8 GB; en GPU (x86) tiny ~0.2 GB, base ~0.3 GB, small ~0.7 GB, medium ~1.7–2 GB, large-v3 ~3.2–3.9 GB de VRAM. Rangos medidos en cinco máquinas, cuatro idiomas: valores por modelo. En la práctica: tiny y base siguen el tiempo real en todos los casos, incluyendo en ARM; small se queda atrás en ARM (2 a 3 veces más lento); medium requiere un x86; large-v3 requiere un x86 con ~8 GB de RAM y solo funciona en GPU (en CPU, incluso un x86 supera el tiempo real).

La herramienta funciona sin conexión una vez que el modelo está presente: el caché de Hugging Face es solo un recurso de respaldo, no la forma habitual de usarla.

Resumen de la línea de comandos

```
whispskrid [-l LANG] [--model NOM]
whispskrid --dictate | --dictate-stop | --toggle | --cancel
whispskrid --status | --stop
whispskrid --diagnose
whispskrid --download-model [NOM]
whispskrid --help | --version
```

Capítulo 3 — Solución de problemas

¿Cómo llega el texto a la aplicación?

WhispSkrid no tiene una interfaz gráfica; escribe el texto transcrito directamente en la aplicación que tiene el foco del teclado, independientemente de cuál sea.

El texto no se **escribe carácter por carácter**. El mecanismo real es: el texto transcrito se coloca en el portapapeles, y luego se simula un atajo de teclado “pegar” (Ctrl+V). Esto es más fiable y más rápido que simular una pulsación de tecla completa, especialmente en textos largos.

El portapapeles original se restaura inmediatamente después del pegado; dictar no sobrescribe lo que se había copiado, por lo tanto, no es permanente. 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 y elimina este riesgo.

El micrófono

WhispSkrid captura el sonido del **dispositivo de entrada predeterminado del sistema** (PulseAudio o PipeWire). No hay una configuración de dispositivo en `config.yaml`: para cambiar de micrófono, se define el dispositivo correcto como entrada predeterminada; esto se hace en la configuración de sonido del escritorio, con `pavucontrol`, o en la línea de comandos (`pacctl set-default-source <nom>`), donde la lista de dispositivos está dada por `pacctl list sources short` o `arecord -l`).

`whispSkrid --diagnose` abre un flujo de prueba en este dispositivo e indica si la apertura falla: el micrófono está ausente, desconectado o ya está siendo utilizado por otra aplicación.

Las herramientas de inyección y su selección automática

La inyección requiere **dos** herramientas: un motor de escritura y una herramienta de portapapeles. El motor de escritura se selecciona una sola vez al inicio, mediante una consulta directa, y nunca se vuelve a probar antes de cada copia.

- **Primera opción:** `ydotool` (entrada a través del controlador, utilizando el demonio `ydotoold`, códigos de teclas dependientes de la distribución del teclado).
- **Si no:** `xdotool` (modo estándar, con detección de la ventana).
- **Si no:** modo degradado: el texto transcrito permanece visible en el terminal y la entrada está desactivada para la sesión, en lugar de un fallo silencioso en cada dictado.

El portapapeles sigue el tipo de sesión: `wl-clipboard` bajo Wayland (siempre requerido, incluso con `ydotool`), `xclip` bajo X11. Sin una herramienta de portapapeles, el texto transcrito nunca llega al portapapeles: nada se escribe en la aplicación y un Ctrl+V vuelve a pegar el último contenido copiado manualmente.

Activar ydotool (solución de respaldo)

Esta sección se aplica únicamente a los sistemas donde el motor seleccionado es `ydotool`. Cuando `xdotool` realiza la entrada, nada de lo que sigue es necesario; en particular, es inútil unirse al grupo `input`.

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.
2. **El servicio `ydotool` está en funcionamiento.** `ydotool` necesita este demonio en segundo plano.
3. **El usuario pertenece al grupo `input`.** Es este grupo el que da acceso a `/dev/uinput`, de lo contrario, `ydotool` fallará silenciosamente:

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

Es necesario cerrar sesión y volver a iniciarla para que este cambio de grupo surta efecto.

Atajos: X11 y Wayland no son iguales

La tecla de “presionar para hablar” integrada es escuchada por la propia herramienta a través de `pynput`, que observa el servidor X. Bajo Wayland, `pynput` solo escucha las ventanas que pasan por **XWayland** (muchas aplicaciones Electron y antiguas), nunca una ventana **Wayland nativa**. Para estas últimas, es necesario enlazar una subcomando de control (`--toggle`, `--dictate`, `--dictate-stop`) a un atajo del entorno de escritorio (Capítulo 2).

`--diagnose`

`whispskrid --diagnose` verifica el entorno y devuelve un código de salida: 0 si todo está correcto, y un valor distinto de cero en caso contrario. Cada punto se evalúa en la línea `ok / !!`:

- **Entorno** — tipo de sesión (X11 / Wayland), backend de inyección retenido.
- **Herramientas** — presencia de `ydotool`, `ydotool`, `xdotool`, `wl-copy` / `wl-paste` o `xclip`, `paplay`; acceso de lectura-escritura a `/dev/uinput`.
- **Backend** — importación de `faster_whisper` y de `ctranslate2`; `device` y `compute_type` efectivos; CUDA disponible o no (informativo).
- **Modelos** — carpeta de modelos resuelta; presencia del modelo por defecto; carga de prueba (montaje en memoria y liberación).
- **Entrada de audio** — apertura de un flujo de captura en el formato esperado.
- **Portapapeles** — lectura/escritura de ida y vuelta.
- **Socket de control** — ruta, directorio de ejecución accesible en escritura, sesión activa o no.
- **Configuración** — ruta del `config.yaml` efectivamente cargado.

Síntoma → causa probable → acción

Síntoma	Causa probable	Acción
El texto transcrito permanece en la terminal; nada se inyecta	Ningún motor de escritura funcional (ni ydotool ni xdotool): modo degradado	Instale ydotool o xdotool, y luego reinicie la sesión
Mensaje “ningún modelo” al inicio	No se encontró ningún modelo en las ubicaciones esperadas	whispskrid --download-model base — véase el Capítulo 1
Nada se está grabando, parece que el micrófono está silenciado	Mal dispositivo de entrada predeterminado, o micrófono silenciado a nivel del sistema	Verificar/cambiar la entrada predeterminada (pavucontrol); whispskrid --diagnose
--diagnose indica el fallo al abrir el flujo de audio	Dispositivo ocupado por otra aplicación, o permisos de acceso insuficientes	Cerrar la otra aplicación; verificar los permisos del dispositivo
Nada se escribe en la aplicación, no se muestra ningún error	La herramienta se ha desplegado en ydotool, pero el usuario no está en el grupo input	Unirse al grupo, reconectarse: ver arriba
La aplicación siempre recupera el último contenido copiado manualmente	Ninguna herramienta de portapapeles instalada (wl-clipboard en Wayland, xclip en X11)	Instálalo
Bajo Wayland, con una aplicación GTK, la ventana muestra brevemente el contenido anterior	La restauración del portapapeles ocurre antes de que la ventana haya leído la selección	clipboard.defer_restore: true en la configuración
La tecla Mayús derecha no hace nada	Ventana Wayland nativa en primer plano — pynput solo muestra ventanas X11 y XWayland	Lier --toggle / --dictate a un acceso directo en el escritorio; ver Capítulo 2
No hay sonido al inicio de la grabación	paplay ausente, o ningún archivo de sonido configurado	Advertencia menor: instale pulseaudio-utils si desea sonido
La captura se detiene automáticamente después de cinco minutos	Golpe de esquina capture.max_seconds ejecutado	Comportamiento deseado; ajustar el valor en la configuración si es necesario (ver Capítulo 2)
vad.enabled: true sin efecto	La detección de silencio no está implementada en esta versión	Configuración reservada; la herramienta continúa funcionando con la función de “presionar para hablar” estricta; consulte el Capítulo 2

Síntoma	Causa probable	Acción
La transcripción es lenta	El modelo es demasiado grande para la máquina, o <code>compute_type</code> es incompatible	Pasar a base o <code>small</code> ; dejar <code>compute_type: auto</code> — ver Capítulo 2
La configuración modificada parece que nunca se tiene en cuenta	El archivo editado no es el que se está utilizando realmente	Verificar el orden de resolución; véase el Capítulo 2
“Ya hay una sesión en curso” al inicio	Una sesión persistente ya está en funcionamiento (solo puede haber una a la vez)	<code>whispskrid --stop</code> , o usar la sesión existente

¿Dónde buscar un mensaje de error?

WhispSkrid muestra sus errores directamente en el terminal donde se inició la sesión persistente. Reiniciar desde un terminal visible es lo primero que se debe hacer para diagnosticar un problema que no esté listado aquí. En caso de duda, `whispskrid --diagnose` verifica todo y muestra el estado de cada punto.

Si el problema persiste

El repositorio del proyecto tiene una página de seguimiento pública: <https://github.com/RonanDavalan/whispskrid/issues>