

# Manual de Usuario: Aplicación de Voz a Texto con Búsqueda de Productos

---

Este manual describe cómo utilizar la aplicación de voz a texto para buscar productos, así como su funcionamiento interno.

## 1. Introducción

---



La aplicación permite a los usuarios realizar búsquedas de productos utilizando comandos de voz. El sistema procesa el lenguaje natural, identifica la intención del usuario y los filtros de búsqueda, consulta una base de datos de productos (Google Sheets) y devuelve resultados relevantes.

## 2. Interfaz de Usuario (Frontend)


---

La interfaz de usuario es una aplicación web sencilla que consta de los siguientes elementos:




### 2.1. Elementos Principales

- **Botón de Micrófono** (): Inicia y detiene la grabación de voz. Cuando está grabando, cambia a un icono de  y tiene una animación de pulsación.
- **Estado (Status)**: Muestra mensajes informativos sobre el estado de la aplicación (ej. "Escuchando...", "Error", "Texto copiado").
- **Área de Texto de Entrada** (`textInput`): Muestra la transcripción en tiempo real de lo que el usuario está diciendo. También se puede usar para escribir texto manualmente.
- **Área de Respuesta del Servidor** (`responseOutput`): Muestra la respuesta recibida del sistema de búsqueda de productos (n8n).

### 2.2. Controles

- **Botón  Guardar**: Guarda el texto actual del área de entrada en una lista de mensajes guardados y envía la consulta al sistema de backend (n8n) para buscar productos.
- **Botón "Limpiar"**: Borra el contenido de las áreas de texto de entrada y respuesta.
- **Botón "Copiar"**: Copia el contenido del área de texto de entrada al portapapeles.




### 2.3. Mensajes Guardados

- **Sección "Mensajes Guardados"**: Muestra un historial de las consultas que el usuario ha guardado.
- **Botón  Limpiar Todo**: Elimina todos los mensajes guardados del historial.
- **Elementos de Mensaje Individual**: Cada mensaje guardado incluye la fecha, un botón  Copiar para copiar solo ese mensaje, y un botón  Eliminar para borrarlo individualmente.


### 3. Uso de la Aplicación

---

#### 1. Iniciar una Búsqueda por Voz:

- Haz clic en el **botón del micrófono** ()
- El estado cambiará a "  Escuchando... Habla ahora".
- Comienza a hablar claramente. Verás cómo tu voz se transcribe en el **área de texto de entrada** en tiempo real.
- Cuando termines de hablar, haz clic nuevamente en el **botón del micrófono** () para detener la grabación.

#### 2. Enviar la Consulta:

- Una vez que el texto deseado esté en el **área de texto de entrada** (ya sea por voz o escrito manualmente), haz clic en el **botón** " Guardar".
- El sistema enviará tu consulta al backend y la respuesta aparecerá en el **área de respuesta del servidor**.

#### 3. Gestionar Mensajes:

- Puedes **copiar** o **limpiar** el texto actual usando los botones correspondientes.
- Los mensajes enviados se guardan automáticamente en la sección "Mensajes Guardados". Puedes copiarlos o eliminarlos individualmente, o borrar todo el historial con "Limpiar Todo".

### 4. Funcionamiento Interno (Backend - n8n Workflow)

---

El corazón de la lógica de búsqueda reside en un flujo de trabajo de n8n, que orquesta la comunicación entre la interfaz de usuario, la inteligencia artificial (Google Gemini) y la base de datos de productos (Google Sheets).

#### 4.1. Componentes del Flujo de Trabajo

##### 1. Webhook:

- **Función:** Es el punto de entrada del flujo de trabajo. Recibe las consultas de texto desde la aplicación web (frontend) a través de una solicitud POST.
- **URL:** `http://localhost:5678/webhook-test/1a6d9c4a-31cf-41e2-bcca-79dcaae94b2b` (Nota: Para un entorno de producción, esta URL sería pública y segura).

##### 2. HTTP Request (Google Gemini API):

- **Función:** Envía la consulta del usuario a la API de Google Gemini (modelo `gemini-2.5-flash`) para clasificar la intención y extraer filtros.
- **Instrucciones (Prompt):** Gemini recibe instrucciones detalladas para analizar el texto del usuario y devolver un objeto JSON con la `intent` (intención) y `filters` (filtros).
  - **Intenciones:**

- `buscar_productos` : Si se detectan filtros específicos (color, tamaño, ubicación, nombre, SKU, material, acabado).
- `listar_todo` : Si la consulta es general (ej. "Muéstrame todo").
- **Filtros Soportados:**
  - `Color` (ej. "blanco", "negro")
  - `Metros por caja` (ej. "grande", "chico", "2.5m2")
  - `Descripción` (ej. "exterior", "interior", "baño", "cerámica", "mate")
  - `Nombre` (ej. "Atlantis")
  - `SKU` (ej. "ABC123")

### 3. Code (Parse Gemini Response):

- **Función:** Procesa la respuesta de texto plano de Google Gemini, limpia cualquier formato adicional (como comillas o etiquetas `json`), y la convierte en un objeto JSON estructurado. Si hay un error en el parseo, por defecto establece la intención a `listar_todo`.

### 4. Selector (Switch Node):

- **Función:** Dirige el flujo de trabajo basándose en la `intent` (intención) clasificada por Gemini.
  - Si la intención es `listar_todo`, el flujo va directamente a obtener todos los productos.
  - Si la intención es `buscar_productos`, el flujo pasa por un nodo `code2` para preparar los filtros.

### 5. Code2 (Prepare Filters):

- **Función:** Toma los filtros extraídos por Gemini y los formatea en un array de objetos (`queryField`, `queryValue`) para que puedan ser utilizados en los siguientes pasos de filtrado.

### 6. Get row(s) in sheet1 (Google Sheets - All Products):

- **Función:** Recupera *todos* los productos de la hoja de cálculo de Google Sheets (`products sheet`, GID `585399714`). Este nodo se activa cuando la intención es `listar_todo` o como parte del proceso de filtrado.
- **Hoja de Cálculo:** `1EtoFbBxXJjN9XRuVzZXNbSktBrdkthbqp0PMLUn0w1k`

### 7. Merge (Combine Data):

- **Función:** Combina los datos de los filtros (del nodo `code2`) con los datos de todos los productos (del nodo `Get row(s) in sheet1`). Esto es crucial para que el nodo de filtrado (`code3`) tenga acceso tanto a los productos como a los criterios de búsqueda.

### 8. Code3 (Filter Products):

- **Función:** Este es el nodo de lógica principal para el filtrado. Recibe todos los productos y los filtros, y aplica la lógica para encontrar los productos que coinciden con *todos* los criterios de búsqueda (AND lógico).
- **Lógica de Filtrado:**

- Normaliza el texto (minúsculas, sin acentos) para comparaciones insensibles a mayúsculas/acentos.
- Maneja filtros por `Color` , `Metros por caja` (clasificando como "grande", "chico", "mediano" o por valor numérico), `Descripción` (incluyendo sinónimos para "exterior", "interior", "baño", etc.), `Nombre` y `SKU` .
- Si no se encuentran productos, devuelve un array vacío.

#### 9. **Code1 (Generate Response):**

- **Función:** Formatea la respuesta final que se enviará al usuario. Si se encontraron productos, los lista con detalles relevantes (SKU, Color, Tamaño, Formato, Uso recomendado, Descripción). Si no se encontraron productos, proporciona sugerencias para mejorar la búsqueda.
- **Mensajes de Error:** Incluye manejo de errores para proporcionar feedback útil al usuario si algo falla durante el procesamiento.

#### 10. **Respond to Webhook:**

- **Función:** Envía la respuesta final generada por `Code1` de vuelta a la aplicación web (frontend).

### 4.2. Flujo de Ejecución Típico

1. El usuario habla o escribe una consulta en la aplicación web.
2. La aplicación web envía el texto al **Webhook** de n8n.
3. El **Webhook** pasa el texto a **HTTP Request**, que lo envía a Google Gemini.
4. Google Gemini clasifica la intención y extrae los filtros, devolviendo un JSON.
5. El nodo **Code** parsea y valida la respuesta de Gemini.
6. El nodo **Selector** dirige el flujo:
  - Si es `listar_todo` , va directamente a `Get row(s) in sheet1` para obtener todos los productos.
  - Si es `buscar_productos` , va a `Code2` para preparar los filtros.
7. `Code2` (si aplica) formatea los filtros.
8. `Get row(s) in sheet1` obtiene todos los productos de Google Sheets.
9. El nodo **Merge** combina los productos con los filtros (si existen).
10. El nodo **Code3** filtra los productos según los criterios.
11. El nodo **Code1** genera la respuesta final basada en los productos filtrados (o un mensaje de "no encontrado").
12. El nodo **Respond to Webhook** envía esta respuesta de vuelta a la aplicación web.
13. La aplicación web muestra la respuesta al usuario.

## 5. Personalización y Mantenimiento

---

### 5.1. Base de Datos de Productos

La aplicación utiliza una hoja de cálculo de Google Sheets como base de datos de productos. Para añadir, modificar o eliminar productos, debes editar directamente la hoja de cálculo vinculada:

- **URL de la Hoja de Cálculo:**

`https://docs.google.com/spreadsheets/d/1EtoFbBxXJjN9XRuVzZXNbSktBrdkthbqp0PMLUnOw1k/edit?usp=drivesdk`

- **GID de la Hoja:** 585399714 (corresponde a la pestaña `products`)

Asegúrate de mantener un formato consistente en las columnas para que el filtrado funcione correctamente (ej. `Nombre`, `SKU`, `Color`, `Metros por caja`, `Descripción`).

### 5.2. Lógica de Clasificación (Google Gemini Prompt)

Si necesitas ajustar cómo la aplicación interpreta las consultas de los usuarios o añadir nuevos tipos de filtros, deberás modificar el `prompt` enviado a Google Gemini dentro del nodo **HTTP Request** en el flujo de trabajo de n8n. Es crucial mantener el formato JSON de salida que Gemini debe generar.

### 5.3. Lógica de Filtrado (Code3 Node)

Para refinar cómo se buscan y comparan los productos con los filtros (ej. añadir nuevos sinónimos para ubicaciones, cambiar la lógica de tamaño), edita el código JavaScript dentro del nodo **Code3** en n8n.

### 5.4. Apariencia (CSS)

Para cambiar la apariencia visual de la aplicación (colores, fuentes, diseño), edita el archivo `style.css`.

### 5.5. Comportamiento del Frontend (JavaScript)

Para modificar el comportamiento de la interfaz de usuario (ej. cómo se manejan los botones, la interacción con el micrófono), edita el archivo `script.js`.

## 6. Solución de Problemas Comunes

---

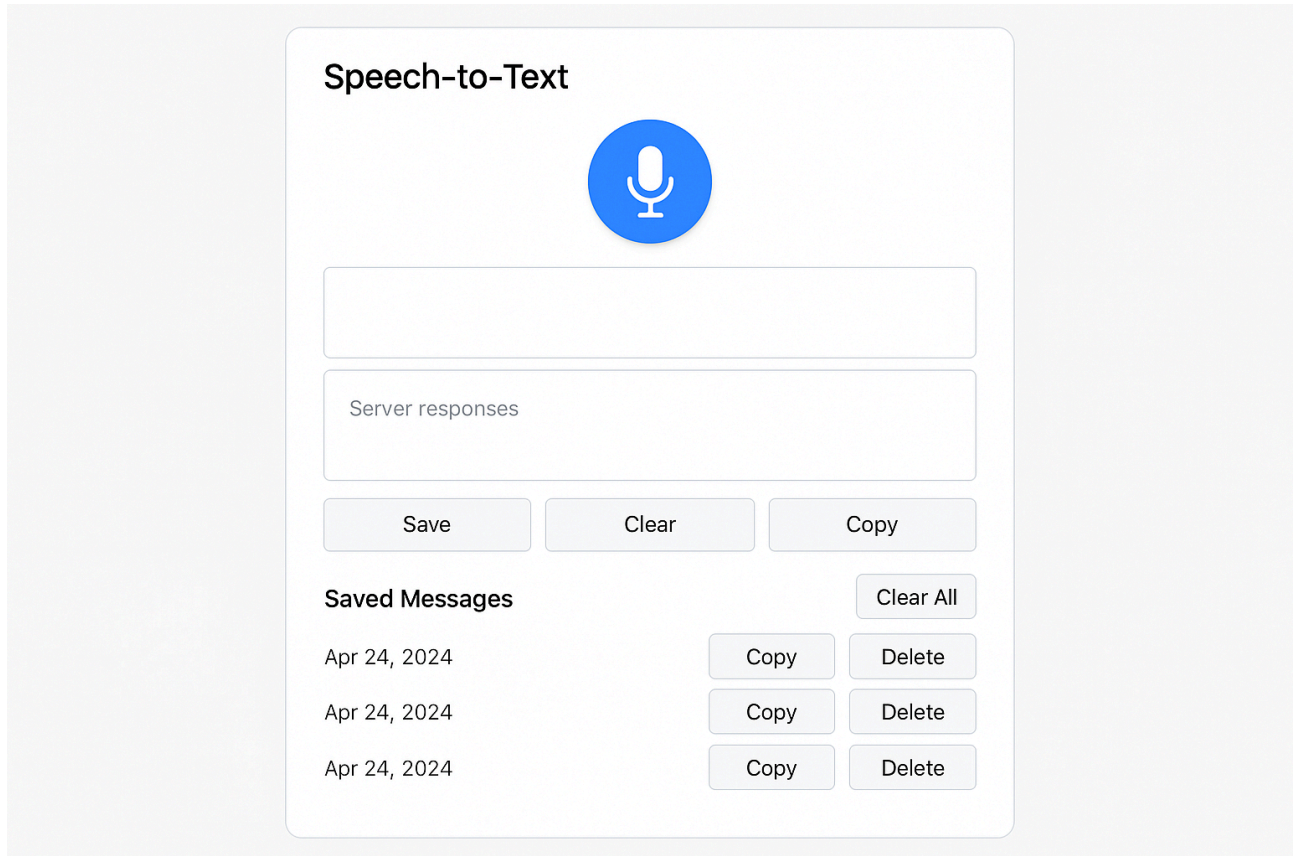
- **"Tu navegador no soporta reconocimiento de voz":** Usa Google Chrome o Safari, que tienen mejor soporte para la Web Speech API.
- **"Permiso denegado":** Asegúrate de haber concedido permiso al navegador para acceder a tu micrófono.
- **"No se detectó voz":** Habla más claro, más alto o más cerca del micrófono. Verifica que tu micrófono esté conectado y funcionando.
- **"Error al consultar: HTTP Error XXX":** Esto indica un problema en la comunicación con el servidor n8n. Verifica que el flujo de trabajo de n8n esté activo y que la `webhookUrl` en `script.js` sea correcta.

- **"No encontré productos que coincidan"**: Intenta reformular tu búsqueda con términos más generales o verifica que los productos existan en tu hoja de cálculo de Google Sheets con los atributos correctos.

---

## Fin del Manual de Usuario

### 2.1.1. Captura de Pantalla de la Interfaz de Usuario



#### 4.1.1. Diagrama del Flujo de Trabajo de n8n

