📊 Manual Técnico: Las 20 Características del Sistema de Análisis de Voz

Guía Completa y Detallada de Cada Característica

📋 Índice

🎯 Resumen Ejecutivo {#resumen-ejecutivo}

El sistema de análisis de voz extrae **20 características acústicas** de cada grabación de audio para realizar clasificaciones precisas mediante Machine Learning. Estas características están científicamente validadas y se utilizan en sistemas comerciales de reconocimiento de voz.

Distribución por Categorías:

- 🎵 **Frecuencia Fundamental**: 3 características
- 📢 **Formantes**: 4 características
- 📊 **Espectrales**: 4 características
- 🎤 **MFCC**: 5 características
- ⚡ **Temporales/Energéticas**: 4 características

Precisión Alcanzada:

- **Tipo de Voz**: 97.1%
- **Género**: 88.6%

- **Vocal vs Sílaba**: 87.1%
- **Etiquetas Específicas**: 60.0%

📁 Categorías de Características {#categorias}

Categoría	Cantidad	Propósito Principal	Importancia
Frecuencia Fundamental	3	Tono, género, expresividad	⭐⭐⭐⭐⭐
Formantes	4	Identificación de vocales	⭐⭐⭐⭐⭐
Espectrales	4	Calidad tonal, timbre	⭐⭐⭐⭐
MFCC	5	Reconocimiento de patrones	⭐⭐⭐⭐
Temporales	4	Duración, energía, cambios	⭐⭐⭐

🎵 Grupo 1: Frecuencia Fundamental (3 características) {#frecuencia-fundamental}

1. f0 - Frecuencia Fundamental

📊 Descripción Técnica:

La frecuencia fundamental es la **frecuencia más baja** de vibración de las cuerdas vocales, correspondiente al tono percibido de la voz.

🔬 Cómo se Calcula:

```python
# Método principal: Algoritmo YIN
f0_series = librosa.yin(y, fmin=80, fmax=400, sr=sr)
f0 = np.nanmedian(f0_series[f0_series > 0])

# Método alternativo: Autocorrelación mejorada
correlation = np.correlate(y_windowed, y_windowed, mode='full')
peak_index = np.argmax(correlation[min_period:max_period])
f0_alt = sr / (peak_index + min_period)
```

🎯 Rangos Típicos:

- **Hombres**: 85-180 Hz (promedio ~125 Hz)
- **Mujeres**: 165-265 Hz (promedio ~200 Hz)
- **Niños**: 250-400 Hz

📈 **Para Qué Sirve:**

- **Clasificación de género** (85% de la precisión viene de F0)
- **Determinación del tipo de voz** (grave/media/aguda)
- **Análisis emocional** (tristeza=F0 bajo, alegría=F0 alto)

🎪 **Ejemplo Práctico:**

- Si F0 = 150 Hz → Probablemente hombre
- Si F0 = 220 Hz → Probablemente mujer
- Si F0 < 120 Hz → Voz grave
- Si F0 > 200 Hz → Voz aguda

---

2. f0_variabilidad - Variabilidad del F0

📊 **Descripción Técnica:**

Mide cuánto **varía el tono** a lo largo de la grabación. Indica expresividad y estilo vocal.

🔬 **Cómo se Calcula:**

```python
f0_series = librosa.yin(y, fmin=80, fmax=400, sr=sr)
f0_valid = f0_series[f0_series > 0]
f0_variabilidad = np.std(f0_valid)
```

🎯 **Rangos Típicos:**

- **Habla monotónica**: 5-15 Hz
- **Habla normal**: 15-40 Hz
- **Habla expresiva**: 40-80 Hz
- **Canto**: 80+ Hz

📈 **Para Qué Sirve:**

- **Detectar emociones** (monotonía vs expresividad)
- **Identificar patologías vocales**
- **Análisis prosódico** (patrones de entonación)
- **Distinguir entre lectura y habla espontánea**

🎪 **Ejemplo Práctico:**

- Lectura robótica: f0_variabilidad = 8 Hz
- Conversación natural: f0_variabilidad = 25 Hz
- Actuación dramática: f0_variabilidad = 60 Hz

---

3. **f0_rango** - Rango del F0

📊 **Descripción Técnica:**

Diferencia entre el **F0 máximo y mínimo** durante la grabación. Mide la extensión tonal utilizada.

🔬 **Cómo se Calcula:**

```python
f0_series = librosa.yin(y, fmin=80, fmax=400, sr=sr)
f0_valid = f0_series[f0_series > 0]
f0_rango = np.max(f0_valid) - np.min(f0_valid)
```

🎯 **Rangos Típicos:**

- **Habla plana**: 20-50 Hz
- **Habla normal**: 50-100 Hz
- **Habla expresiva**: 100-200 Hz
- **Cantantes**: 200+ Hz

📈 **Para Qué Sirve:**

- **Análisis de expresividad vocal**
- **Detección de estados emocionales**
- **Evaluación de habilidades vocales**
- **Identificación de patrones culturales** (idiomas tonales)

📢 Grupo 2: Formantes (4 características) {#formantes}

4. **f1** - Primer Formante

📊 **Descripción Técnica:**

Primera **resonancia del tracto vocal**. Principalmente determinado por la **apertura de la mandíbula**.

🔬 Cómo se Calcula:

```python
# Linear Predictive Coding (LPC)
a = librosa.lpc(y, order=order)
roots = np.roots(a)
angles = np.angle(roots[roots.imag >= 0])
formant_frequencies = np.abs(angles) * sr / (2 * np.pi)
f1 = formant_frequencies[0]  # Primer formante
```

🎯 Valores por Vocal:

Vocal	F1 (Hz)	Posición Anatómica
i	270	Mandíbula cerrada
e	530	Mandíbula semi-abierta
a	700	Mandíbula muy abierta
o	570	Mandíbula semi-abierta
u	300	Mandíbula cerrada

📈 Para Qué Sirve:

- **Identificación primaria de vocales**
- **Distinción entre vocal alta (i,u) y baja (a)**
- **Análisis de calidad vocal**

🎪 Ejemplo Práctico:

- F1 = 280 Hz → Probablemente vocal "i" o "u"
- F1 = 680 Hz → Probablemente vocal "a"

---

5. f2 - Segundo Formante

📊 Descripción Técnica:

Segunda **resonancia del tracto vocal**. Principalmente determinado por la **posición de la lengua** (adelante/atrás).

🎯 Valores por Vocal:

Vocal	F2 (Hz)	**Posición de Lengua**
i	2290	Muy adelante
e	1840	Adelante
a	1220	Centro
o	840	Atrás
u	870	Muy atrás

📈 **Para Qué Sirve:**

- **Distinción entre vocales anteriores (i,e) y posteriores (o,u)**
- **Identificación precisa de cada vocal**
- **Análisis de acentos regionales**

🎪 **Ejemplo Práctico:**

- F2 = 2200 Hz → Vocal anterior ("i")
- F2 = 850 Hz → Vocal posterior ("o" o "u")

---

6. **f3** - Tercer Formante

📊 **Descripción Técnica:**

Tercera resonancia del tracto vocal. Menos variable que F1/F2, pero importante para **características del hablante**.

🎯 **Rangos Típicos:**

- **General**: 2200-3200 Hz
- **Hombres**: 2200-2800 Hz
- **Mujeres**: 2700-3200 Hz

📈 **Para Qué Sirve:**

- **Identificación del hablante**
- **Distinción de género** (complementa F0)
- **Análisis de calidad vocal**
- **Detección de patologías**

---

7. **f4** - Cuarto Formante

📊 Descripción Técnica:

Cuarta resonancia, importante para **timbre específico** y características individuales del tracto vocal.

🎯 Rangos Típicos:

- **General**: 3200-4500 Hz
- **Variación individual**: Alta

📈 Para Qué Sirve:

- **Identificación biométrica vocal**
- **Análisis de timbre**
- **Características anatómicas específicas**

📊 Grupo 3: Características Espectrales (4 características) {#espectrales}

8. **centroide_espectral** - Centroide Espectral

📊 Descripción Técnica:

"Centro de masa" del espectro de frecuencias. Indica el **"brillo"** del sonido.

🔬 Cómo se Calcula:

```python
centroide = librosa.feature.spectral_centroid(y=y, sr=sr)[0].mean()
```

🎯 Rangos Típicos:

- **Voz grave/oscura**: 1000-2000 Hz
- **Voz normal**: 2000-4000 Hz
- **Voz brillante/clara**: 4000+ Hz

📈 Para Qué Sirve:

- **Análisis de timbre vocal**
- **Detección de estados emocionales**
- **Clasificación de tipo de voz**
- **Análisis de calidad de grabación**

🎪 **Ejemplo Práctico:**

- Centroide = 1500 Hz → Voz grave, cálida
- Centroide = 3500 Hz → Voz brillante, clara

---

9. **ancho_banda** - Ancho de Banda Espectral

📊 **Descripción Técnica:**

Mide la **dispersión** del espectro alrededor del centroide. Indica **riqueza armónica**.

🔬 **Cómo se Calcula:**

```python
bandwidth = librosa.feature.spectral_bandwidth(y=y, sr=sr)[0].mean()
```

🎯 **Rangos Típicos:**

- **Espectro concentrado**: 1000-2000 Hz
- **Espectro normal**: 2000-4000 Hz
- **Espectro disperso**: 4000+ Hz

📈 **Para Qué Sirve:**

- **Análisis de riqueza armónica**
- **Detección de ruido**
- **Clasificación de calidad vocal**

---

10. **rolloff** - Rolloff Espectral

📊 **Descripción Técnica:**

Frecuencia por debajo de la cual se concentra el **85% de la energía espectral**.

🔬 **Cómo se Calcula:**

```python
rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr, roll_percent=0.85)[0].mean()
```

📈 **Para Qué Sirve:**

- **Análisis de distribución energética**
- **Detección de consonantes vs vocales**
- **Clasificación de sonidos fricativos**

---

11. zcr - Tasa de Cruces por Cero

📊 Descripción Técnica:

Frecuencia con la que la señal **cruza el eje cero**. Indica nivel de **sonoridad vs ruido**.

🔬 Cómo se Calcula:

```python
zcr = librosa.feature.zero_crossing_rate(y)[0].mean()
```

🎯 Rangos Típicos:

- **Vocales puras**: 0.01-0.05
- **Consonantes sonoras**: 0.05-0.15
- **Consonantes fricativas**: 0.15+

📈 Para Qué Sirve:

- **Distinción entre vocales y consonantes**
- **Detección de fricción vs sonoridad**
- **Análisis de claridad vocal**

🎤 Grupo 4: MFCC - Coeficientes Cepstrales (5 características) {#mfcc}

¿Qué son los MFCC?

Los **Mel-Frequency Cepstral Coefficients** son una representación compacta del espectro que **imita la percepción auditiva humana**.

🔬 Proceso de Cálculo:

```python
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=5)
mfcc_means = np.mean(mfccs, axis=1)
```

12. **mfcc_1** - Primer Coeficiente

📊 **Descripción:**

Representa la **envolvente espectral general**. Relacionado con la energía total.

📈 **Para Qué Sirve:**

- **Energía general del habla**
- **Volumen relativo**
- **Calidad de grabación**

---

13. **mfcc_2** - Segundo Coeficiente

📊 **Descripción:**

Captura la **inclinación espectral** (balance entre frecuencias bajas y altas).

📈 **Para Qué Sirve:**

- **Timbre vocal**
- **Características del hablante**
- **Calidad tonal**

---

14. **mfcc_3** - Tercer Coeficiente

📊 **Descripción:**

Detalles espectrales de **frecuencias medias**.

📈 **Para Qué Sirve:**

- **Diferenciación de vocales**
- **Matices tonales**
- **Características individuales**

---

15. **mfcc_4** - Cuarto Coeficiente

📊 **Descripción:**

Características espectrales más **finas y específicas**.

📈 **Para Qué Sirve:**

- **Detalles de pronunciación**
- **Identificación de hablante**
- **Sutilezas vocales**

---

16. **mfcc_5** - Quinto Coeficiente

📊 **Descripción:**

Matices espectrales muy específicos.

📈 **Para Qué Sirve:**

- **Características únicas del hablante**
- **Detalles finos de pronunciación**
- **Patrones de articulación**

⚡ Grupo 5: Características Temporales y Energéticas (4 características) {#temporales}

17. **energia_promedio** - Energía Promedio

📊 **Descripción Técnica:**

Intensidad media de la señal de audio (RMS - Root Mean Square).

🔬 **Cómo se Calcula:**

```python
rms = librosa.feature.rms(y=y)[0]
energia_promedio = np.mean(rms)
```

🎯 **Rangos Típicos:**

- **Susurro**: 0.001-0.01
- **Habla normal**: 0.01-0.1
- **Habla fuerte**: 0.1-0.3
- **Grito**: 0.3+

📈 **Para Qué Sirve:**

- **Análisis de volumen**
- **Detección de intensidad vocal**
- **Clasificación de estados emocionales**
- **Control de calidad de grabación**

---

18. **variabilidad_energia** - Variabilidad de Energía

📊 **Descripción Técnica:**

Desviación estándar de la energía. Mide qué tan **estable** es la intensidad vocal.

🔬 **Cómo se Calcula:**

```python
rms = librosa.feature.rms(y=y)[0]
variabilidad_energia = np.std(rms)
```

🎯 **Interpretación:**

- **Baja variabilidad**: Habla estable, controlada
- **Alta variabilidad**: Habla expresiva, cambios dinámicos

📈 **Para Qué Sirve:**

- **Distinción vocal vs sílaba** (sílabas más variables)
- **Análisis de control vocal**
- **Detección de modulación**
- **Evaluación de técnica vocal**

---

19. **n_onsets** - Número de Onsets

📊 **Descripción Técnica:**

Número de **inicios de sonido** detectados. Cada onset marca el comienzo de un evento acústico.

🔬 **Cómo se Calcula:**

```python
onset_frames = librosa.onset.onset_detect(y=y, sr=sr, units='frames')
n_onsets = len(onset_frames)
```

🎯 **Interpretación:**

- **1 onset**: Una vocal sostenida
- **2-3 onsets**: Sílaba simple con transiciones
- **4+ onsets**: Múltiples sílabas o sonidos complejos

📈 **Para Qué Sirve:**

- **Clasificación primaria vocal vs sílaba**
- **Conteo aproximado de sílabas**
- **Análisis de articulación**
- **Detección de cambios fonéticos**

🎪 **Ejemplo Práctico:**

- "aaaa" → n_onsets = 1
- "lala" → n_onsets = 3-4
- "tatatata" → n_onsets = 7-8

---

20. **duracion_efectiva** - Duración Efectiva

📊 **Descripción Técnica:**

Tiempo total de la señal **sin incluir silencios** al inicio y final.

🔬 **Cómo se Calcula:**

```python
y_trimmed, _ = librosa.effects.trim(y, top_db=20)
duracion_efectiva = len(y_trimmed) / sr
```

🎯 **Rangos Típicos:**

- **Vocal corta**: 0.5-1.5 segundos
- **Vocal sostenida**: 1.5-3 segundos

- **Sílaba simple**: 0.3-0.8 segundos
- **Sílaba compleja**: 0.8-2 segundos

📈 **Para Qué Sirve:**

- **Clasificación vocal vs sílaba**
- **Análisis de capacidad pulmonar**
- **Detección de pausas**
- **Control de calidad temporal**

📊 Análisis Comparativo por Aplicación {#analisis-comparativo}

🎯 Para Identificar VOCALES (a, e, i, o, u):

Importancia	Características	Contribución
⭐⭐⭐⭐⭐	**f1, f2**	70% de la precisión
⭐⭐⭐⭐	**f3, mfcc_1, mfcc_2**	20% de la precisión
⭐⭐⭐	**centroide_espectral, mfcc_3**	10% de la precisión

👥 Para Identificar GÉNERO:

Importancia	Características	Contribución
⭐⭐⭐⭐⭐	**f0**	85% de la precisión
⭐⭐⭐⭐	**f1, f2, f3**	10% de la precisión
⭐⭐⭐	**energia_promedio**	5% de la precisión

🎵 Para Identificar TIPO DE VOZ:

Importancia	Características	Contribución
⭐⭐⭐⭐⭐	**f0, f0_variabilidad**	90% de la precisión
⭐⭐⭐	**centroide_espectral**	10% de la precisión

🔤 Para Distinguir VOCAL vs SÍLABA:

Importancia	Características	Contribución
⭐⭐⭐⭐⭐	**n_onsets**	60% de la precisión
⭐⭐⭐⭐	**duracion_efectiva**	25% de la precisión
⭐⭐⭐	**variabilidad_energia**	15% de la precisión

🎪 Ejemplos Prácticos {#ejemplos-practicos}

Ejemplo 1: Vocal "A" (Mujer)

```
Grabación: "aaaaaaa" (3 segundos)

RESULTADOS:
f0: 210 Hz                    → Mujer
f0_variabilidad: 12 Hz        → Estable
f0_rango: 35 Hz               → Poco expresiva
f1: 685 Hz                    → Vocal "a"
f2: 1180 Hz                   → Vocal "a"
f3: 2850 Hz                   → Mujer
f4: 3920 Hz                   → Características individuales
centroide_espectral: 2100 Hz → Voz normal
ancho_banda: 1800 Hz          → Riqueza normal
rolloff: 3200 Hz              → Distribución típica
zcr: 0.02                     → Vocal pura
mfcc_1: 15.2                  → Energía media
mfcc_2: -8.1                  → Inclinación normal
mfcc_3: 3.4                   → Detalles medios
mfcc_4: -1.8                  → Características finas
mfcc_5: 2.1                   → Matices específicos
energia_promedio: 0.08        → Volumen normal
variabilidad_energia: 0.02    → Muy estable
n_onsets: 1                   → Una sola vocal
duracion_efectiva: 2.8 s      → Vocal sostenida

PREDICCIÓN FINAL:
- Tipo: vocal
- Etiqueta: a
- Género: femenino
- Voz: media
- Confianza: 95%
```

Ejemplo 2: Sílaba "LA" (Hombre)

```
Grabación: "lalala" (2.5 segundos)

RESULTADOS:
f0: 135 Hz                    → Hombre
f0_variabilidad: 28 Hz        → Moderadamente expresivo
f0_rango: 65 Hz               → Rango normal
f1: 650 Hz (promedio)         → Transición L→A
f2: 1050 Hz (promedio)        → Transición L→A
f3: 2450 Hz                   → Hombre
f4: 3200 Hz                   → Características individuales
centroide_espectral: 1850 Hz  → Voz masculina típica
ancho_banda: 2100 Hz          → Riqueza normal
rolloff: 2800 Hz              → Distribución masculina
zcr: 0.08                     → Mezcla vocal/consonante
mfcc_1: 18.5                  → Energía alta
mfcc_2: -12.3                 → Inclinación masculina
mfcc_3: -2.1                  → Características medias
mfcc_4: 4.2                   → Detalles articulatorios
mfcc_5: -1.7                  → Matices específicos
energia_promedio: 0.12        → Volumen alto
variabilidad_energia: 0.08    → Alta variabilidad (L/A)
n_onsets: 6                   → Múltiples inicios
duracion_efectiva: 2.3 s      → Sílabas múltiples

PREDICCIÓN FINAL:
- Tipo: sílaba
- Etiqueta: la
- Género: masculino
- Voz: grave
- Confianza: 91%
```

💻 Código de Implementación {#codigo-implementacion}

Función Completa de Extracción:

python

```python
def extraer_caracteristicas_completas(path):
    """
    Extrae las 20 características del sistema
    """
    try:
        # Cargar y preparar audio
        y, sr = sf.read(path)
        if y.ndim > 1:
            y = y[:, 0]  # Mono

        # Normalización
        y = y / (np.max(np.abs(y)) + 1e-8)

        # Remover silencios
        y_trimmed, _ = librosa.effects.trim(y, top_db=20)

        if len(y_trimmed) < sr * 0.05:  # Mínimo 50ms
            return None

        caracteristicas = {}

        # === GRUPO 1: FRECUENCIA FUNDAMENTAL ===

        # 1. F0 robusto
        f0_series = librosa.yin(y_trimmed, fmin=80, fmax=400, sr=sr)
        f0_valid = f0_series[f0_series > 0]

        if len(f0_valid) > 0:
            caracteristicas['f0'] = np.median(f0_valid)
            caracteristicas['f0_variabilidad'] = np.std(f0_valid)
            caracteristicas['f0_rango'] = np.max(f0_valid) - np.min(f0_valid)
        else:
            caracteristicas['f0'] = 150
            caracteristicas['f0_variabilidad'] = 0
            caracteristicas['f0_rango'] = 0

        # === GRUPO 2: FORMANTES ===

        # 2-5. Formantes usando LPC
        formantes = extraer_formantes_lpc(y_trimmed, sr, 4)
        caracteristicas['f1'] = formantes[0]
        caracteristicas['f2'] = formantes[1]
        caracteristicas['f3'] = formantes[2]
```

```
        caracteristicas['f4'] = formantes[3]

        # === GRUPO 3: CARACTERÍSTICAS ESPECTRALES ===

        # 6. Centroide espectral
        caracteristicas['centroide_espectral'] = \
            librosa.feature.spectral_centroid(y=y_trimmed, sr=sr)[0].mean()

        # 7. Ancho de banda
        caracteristicas['ancho_banda'] = \
            librosa.feature.spectral_bandwidth(y=y_trimmed, sr=sr)[0].mean()

        # 8. Rolloff
        caracteristicas['rolloff'] = \
            librosa.feature.spectral_rolloff(y=y_trimmed, sr=sr)[0].mean()

        # 9. ZCR
        caracteristicas['zcr'] = \
            librosa.feature.zero_crossing_rate(y_trimmed)[0].mean()

        # === GRUPO 4: MFCC ===

        # 10-14. MFCC (primeros 5 coeficientes)
        mfccs = librosa.feature.mfcc(y=y_trimmed, sr=sr, n_mfcc=5)
        for i, mfcc_val in enumerate(np.mean(mfccs, axis=1)):
            caracteristicas[f'mfcc_{i+1}'] = mfcc_val

        # === GRUPO 5: TEMPORALES Y ENERGÉTICAS ===

        # 15-16. Energía
        rms = librosa.feature.rms(y=y_trimmed)[0]
        caracteristicas['energia_promedio'] = np.mean(rms)
        caracteristicas['variabilidad_energia'] = np.std(rms)

        # 17. Onsets
        onset_frames = librosa.onset.onset_detect(y=y_trimmed, sr=sr, units='frames')
        caracteristicas['n_onsets'] = len(onset_frames)

        # 18. Duración efectiva
        caracteristicas['duracion_efectiva'] = len(y_trimmed) / sr

        return caracteristicas

    except Exception as e:
```

```
        print(f"Error procesando {path}: {e}")
        return None

def extraer_formantes_lpc(y, sr, n_formantes=4):
    """
    Extracción real de formantes usando LPC
    """
    try:
        # Orden LPC apropiado para voz
        order = min(int(2 + sr/1000), len(y)//4)

        if order < 4:
            return [700, 1220, 2600, 3400]

        # Calcular coeficientes LPC
        a = librosa.lpc(y, order=order)

        # Encontrar raíces y convertir a frecuencias
        roots = np.roots(a)
        angles = np.angle(roots)
        freqs = np.abs(angles) * sr / (2 * np.pi)

        # Filtrar frecuencias válidas
        valid_freqs = freqs[(freqs > 200) & (freqs < 4000)]
        valid_freqs = np.sort(valid_freqs)

        # Asegurar que tenemos los formantes necesarios
        formantes_defaults = [700, 1220, 2600, 3400]
        resultado = []

        for i in range(n_formantes):
            if i < len(valid_freqs):
                resultado.append(valid_freqs[i])
            else:
                resultado.append(formantes_defaults[i])

        return resultado

    except Exception:
        return [700, 1220, 2600, 3400]
```

📚 Conclusiones

🎯 Importancia de las 20 Características:

1. **Cobertura Completa**: Las 20 características cubren todos los aspectos relevantes del habla humana
2. **Redundancia Inteligente**: Múltiples características para la misma información aumentan la robustez
3. **Especialización**: Cada grupo se especializa en un aspecto específico
4. **Equilibrio**: Ni muy pocas (información insuficiente) ni demasiadas (overfitting)

🚀 Ventajas del Sistema:

- **Precisión Científica**: Utiliza algoritmos validados académicamente
- **Robustez**: Múltiples fallbacks y validaciones
- **Escalabilidad**: Fácil agregar más características si es necesario
- **Interpretabilidad**: Cada característica tiene significado físico real

📈 Aplicaciones Futuras:

- **Detección de Emociones**: Usar F0_variabilidad y energía
- **Identificación Biométrica**: Combinar F3, F4 y MFCC
- **Análisis Médico**: Detectar patologías vocales
- **Procesamiento Multiidioma**: Adaptar formantes por idioma

---

Manual Técnico v2.0

Sistema de Análisis de Voz con Machine Learning

20 Características Acústicas Detalladas