

# Informe

Proyecto Computación  
Distribuida



ionic

django



UTEM

## Versión 1.1

Manual actualizado al 19 de Julio del 2019.

Documento exclusivo para el desarrollo del Proyecto de Computación Distribuida.

Manual desarrollado por **Fabián Alvarado, Constanza Jazme**  
y **Mauricio González**



# Tabla de contenidos

## 1. Generalidades

- 1.1. Descripción
- 1.2. Uso de Herramientas

## 2. Vistas

- 2.1. Vistas Generales
- 2.2. Primera Vista
- 2.3. Segundo Vista
- 2.4. Tercera Vista

## 3. Enfoque y Funcionamiento de la API

- 3.1. Frecuencia Fundamental
- 3.2. MFCC
- 3.3. Modelos
- 3.4. Algoritmos

---

# 1. Generalidades

# 1. Generalidades

---

## 1.1 Descripción

El objetivo de este proyecto es desarrollar un sistema distribuido, donde se desarrolle una aplicación móvil que, mediante la voz, permitirá determinar si una persona es ***mujer*** u ***hombre***, además a que rango de edad pertenece:

- Niño [0-14 años]
- Adolescente [15-29 años]
- Adulto [30-59 años]
- Anciano [60 años o más]

# 1. Generalidades

---

## 1.2 Uso de Herramientas



**GitHub.** Usado como fuente de repositorio que almacena todo el código de la Aplicación Móvil como del servicio API-REST



**Django REST Framework.** Es una de las apps de terceros más usadas para django, necesaria para crear la API-REST sobre django.

**Ionic.** Framework usado para la construcción de la aplicación móvil, usando distintos complementos (en especial cordova) para poder acceder a los recursos del Móvil y poder compilar en emuladores.



**Python.** Lenguaje de Programación de código legible, usado como base para la creación de la API-REST (django está escrito en python).



**Docker.** Utilizado para crear un contenedor portable con los requerimientos necesarios para ejecutar el software.

---

## 2. Vistas

## 2. Vistas

### 2.1 Vistas Generales





## 2. Vistas

### 2.2 Primera Vista (Home)



Vista inicial de la aplicación, que permite grabar y almacenar audios, y los coloca en una lista donde da múltiples opciones como escucharlo, borrarlo o analizarlo.

Al pulsar la tercera opción (botón verde) se enviará el audio seleccionado a la API-REST creada (segunda vista), y esta después de analizar el audio, retorna la información (tercera vista).

## 2. Vistas

### 2.2 Segunda Vista (Análisis)

Vista de espera para el usuario donde se le informa que su audio esta en procesamiento, el cual está siendo enviado al servicio API-REST y espera la respuesta con los datos obtenidos del audio.



## 2. Vistas

---

### 2.3 Tercera Vista (Resultados)



Vista que muestra los datos obtenidos sobre el audio, entrega la información del género y rango etario que se pudo analizar del audio enviado.

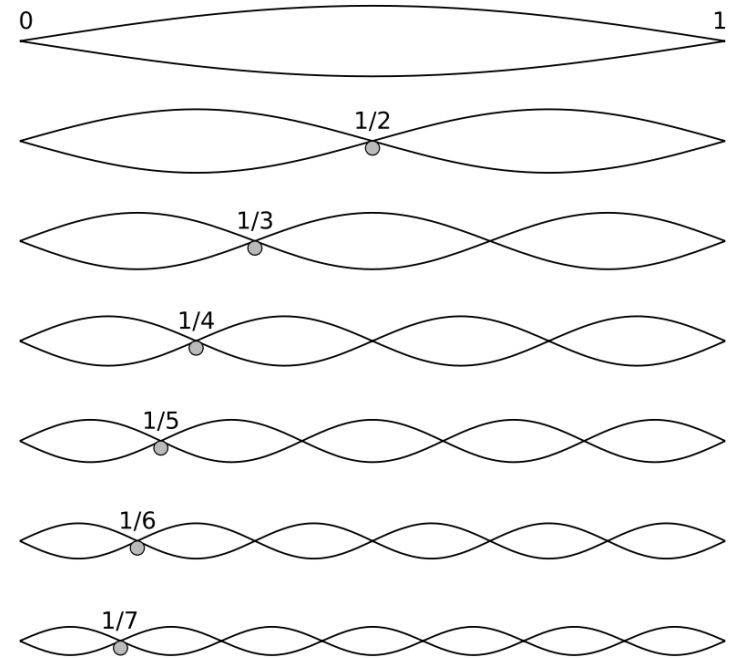
---

## 3. Enfoque y Funcionamiento de la API.

## 3. Enfoque y Funcionamiento de la API

### 3.1 Frecuencia Fundamental( $F_0$ )

La Frecuencia Fundamental es la frecuencia más baja de todas las ondas ( $F_0$ ).



## 3. Enfoque y Funcionamiento de la API

---

### 3.2 MFCC

Los MFCC son Coeficientes Cepstrales en las Frecuencias de Mel, el objetivo de la función **get\_MFCC** es caracterizar el filtro y eliminar la fuente lo cual es desarrollado en cinco etapas:

1. Se transforma la señal de voz en pequeños tramos
2. A los tramos obtenidos se le aplica la Transformada de Fourier discreta para obtener un espectro.
3. Aplicar el bando de filtros correspondiente a la Escala Mel al espectro anterior.
4. Generar el logaritmo de todas las energías de la frecuencia mel.
5. Aplicar la transformada de coseno discreta a estos algoritmos.

De esta manera la funcion **get\_MFCC** retorna las características necesarias.

## 3. Enfoque y Funcionamiento de la API

---

### 3.3 Modelos

En base a datos de entrenamiento para cada género, el código utilizado en el proyecto permite entregar los modelos de género utilizando GMMs (Método de los momentos generalizado), este código es ejecutado una vez para hombre y mujer generando un modelo para cada uno.

### 3. Enfoque y Funcionamiento de la API

---

#### 3.4 Algoritmo

Una vez se recibe la petición de la aplicación móvil, se toma el audio que viene dentro de esta y se verifica el formato en que viene, ya que el algoritmo para el reconocimiento de género y edad solo acepta archivos con extensión “.wav”. En la API se da la posibilidad de enviar 4 tipos de archivos ya sean “.mp3” , “.ogg”, “.m4a” o “.wav”, los cuales son convertidos al formato correcto.

Luego se procede con el algoritmo que reconoce género, consultando los modelos entrenados y comparando posibilidades, se escoge al que tenga la probabilidad o “likelihood” más alta. Después, se hace uso de la librería `mysbsolution.praat` para obtener las frecuencias fundamentales (F0) de la grabación y dar una estimación sobre el rango etario correspondiente según género.

Finalmente, el resultado es devuelto como JSON y con la siguiente estructura:

```
data_response={  gender=gender,    age=age,    error=número}
```

*\*Si el **número** es 1, significa que hubo un problema con el archivo, o que esté dañado*





## Versión 1.1

Manual actualizado al 19 de Julio del 2019.

Documento exclusivo para el desarrollo del  
Proyecto de Computación Distribuida.



ionic

django



Manual desarrollado por **Fabián Alvarado**, **Constanza Jazme** y **Mauricio González**