

**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE COMPROBACIÓN DE LOS
MENSAJES DE ANUNCIO ALOJADOS EN LOS MSS'S DE LA EMPRESA
TELEFONICA**

JUAN SEBASTIAN ROMERO ULLOA

**UNIVERSIDAD SANTO TOMÁS DE AQUINO
FACULTAD DE INGENIERÍA
INGENIERÍA DE TELECOMUNICACIONES
BOGOTÁ D.C.
2019**

**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE COMPROBACIÓN DE LOS
MENSAJES DE ANUNCIO ALOJADOS EN LOS MSS'S DE LA EMPRESA
TELEFONICA**

Presentado por:

**JUAN SEBASTIAN ROMERO ULLOA
CÓDIGO: 2165974**

**Trabajo opción de grado Pasantías en la empresa Telefónica en el área de
operaciones del Core de voz CICO con el propósito de obtener el título de
Ingeniero de Telecomunicaciones**

**Director: Víctor Manuel Castro
Ingeniero Electrónico**

**UNIVERSIDAD SANTO TOMÁS DE AQUINO
FACULTAD DE INGENIERÍA
INGENIERÍA DE TELECOMUNICACIONES
BOGOTÁ D.C.
2018**

RECTOR GENERAL
Padre Juan Ubaldo López Salamanca, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL
Padre Luis Francisco Sastoque Poveda, O.P.

VICERRECTOR ACADÉMICO GENERAL
P. Mauricio Antonio Cortés Gallego O.P

SECRETARIA GENERAL
Ingrid Lorena Campos Vargas

SECRETARIA DE DIVISIÓN
E. C. Luz Patricia Rocha Caicedo

DECANO FACULTAD DE INGENIERÍA DE TELECOMUNICACIONES
Ingeniero Germán Macías Muñoz

Nota de Aceptación.

Firma Ingeniero. Víctor Manuel Castro
Tutor Asignado

Firma del Jurado

Firma del Jurado

Fecha

Contenido

ÍNDICE DE TABLAS	11
ÍNDICE DE FIGURAS	12
TERMINOS Y DEFINICIONES.....	15
INTRODUCCIÓN	20
1. CAPÍTULO I.....	22
1.1 PLANTEAMIENTO DEL PROBLEMA	22
1.2 JUSTIFICACIÓN	24
1.3 OBJETIVOS.....	25
1.3.1 Objetivo General	25
1.3.2 Objetivos Específicos.....	25
2. CAPÍTULO II.....	26
2.1.0 MARCO TEÓRICO	26
2.1.1 Arquitectura VoLTE.....	26
2.1.1.1 VoLTE UE.....	26
2.1.1.2 RAN (Radio Access Network)	26
2.1.1.3 Arquitectura logica.....	26
2.1.2 E-UTRAN (Evolved Universal Terrestrial acces network e-node b)	30
2.1.3 Elementos del Core IMS ePacket Core (Evolved Packet Core)	30
2.1.4 Tecnología 3G	32
2.1.5 3GPP (Third Partnership Project)	36
2.2 Técnicas de Análisis de Audio	36
2.2.1 Análisis de muestras de Audio digital.....	36
2.2.2 Huella acústica digital	37
2.3 Uso único de la Metodología Ágil de programación.....	38
2.3.0 Metodología Xtreme Programming XP (Programacion extrema).....	38
3. CAPÍTULO III.....	43
3.1 Análisis posterior y requerimientos previos según metodología XP	43
3.2 Planificación de la solución según metodología XP	44
3.3 Diseño de la solución según metodología XP	44
3.3.1 Diseño del software	44

3.3.1.1	Diseño del algoritmo para realizar llamada desde pc	46
3.3.1.2	Diseño del algoritmo para realizar grabación de llamada	48
3.3.1.3	Uso del algoritmo de análisis de audio de Shazam	50
3.3.1.4	Perfil de uso del software	51
3.3.2	Diseño de la base de datos	51
3.3.2.1	Diseño de las tablas	51
3.3.2.2	Diseño de los campos de la base de datos	52
3.3.2.3	Diseño de interfaz para presentar los datos	53
3.3.2.4	Diagrama de secuencia	53
3.4	Desarrollo de la solución según metodología XP	55
3.4.1	Desarrollo del software	55
3.4.1.1	Interfaz visual	55
3.4.2	Herramientas de Terceros	59
3.4.2.1.1	ADB Shell Android	59
3.4.2.1.1.1	Uso de ADB Shell Android en el proyecto	60
3.4.2.1.2	Librería MusicG (Google)	61
3.4.2.1.3	Uso de la Librería MusicG (Google)	62
3.4.2.1.4	Google Charts (Tablas)	63
3.4.3	Desarrollo de la base de datos	63
3.4.3.1	Desarrollo de la interfaz de consulta y páginas web	65
4.	CAPÍTULO IV	69
4.1	Conclusiones	69
	BIBLIOGRAFÍA	71
	ÍNDICE DE TABLAS	7
	ÍNDICE DE FIGURAS	8
	TERMINOS Y DEFINICIONES	9
	INTRODUCCIÓN	12
1.	CAPÍTULO I	14
1.1	PLANTEAMIENTO DEL PROBLEMA	14
1.2	JUSTIFICACIÓN	15
1.3	OBJETIVOS	16
1.3.1	Objetivo General	16

1.3.2	<u>Objetivos Específicos</u>	16
2.	<u>CAPÍTULO II</u>	17
2.1.0	<u>MARCO TEÓRICO</u>	17
2.1.1	<u>Arquitectura VoLTE</u>	17
2.1.1.1	<u>VoLTE UE</u>	17
2.1.1.2	<u>RAN (Radio Access Network)</u>	17
2.1.1.3	<u>Arquitectura logica</u>	17
2.1.2	<u>E-UTRAN (Evolved Universal Terrestrial acces network e-node b)</u>	19
2.1.3	<u>Elementos del Core IMS ePacket Core (Evolved Packet Core)</u>	19
2.1.4	<u>Tecnología 3G</u>	20
2.1.5	<u>3GPP (Third Partnership Project)</u>	23
2.2	<u>Técnicas de Análisis de Audio</u>	23
2.2.1	<u>Análisis de muestras de Audio digital</u>	23
2.2.2	<u>Huella acústica digital</u>	24
2.3	<u>Uso único de la Metodología Ágil de programación</u>	25
2.3.0	<u>Metodología Xtreme Programming XP (Programacion extrema)</u>	25
3.	<u>CAPÍTULO III</u>	29
3.1	<u>Análisis posterior y requerimientos previos según metodología XP</u>	29
3.2	<u>Planificación de la solución según metodología XP</u>	30
3.3	<u>Diseño de la solución según metodología XP</u>	31
3.3.1	<u>Diseño del software</u>	31
3.3.1.1	<u>Diseño del algoritmo para realizar llamada desde pc</u>	32
3.3.1.2	<u>Diseño del algoritmo para realizar grabación de llamada</u>	34
3.3.1.3	<u>Uso del algoritmo de análisis de audio de Shazam</u>	36
3.3.1.4	<u>Perfil de uso del software</u>	37
3.3.2	<u>Diseño de la base de datos</u>	37
3.3.2.1	<u>Diseño de las tablas</u>	37
3.3.2.2	<u>Diseño de los campos de la base de datos</u>	38
3.3.2.3	<u>Diseño de interfaz para presentar los datos</u>	39
3.3.2.4	<u>Diagrama de secuencia</u>	39
3.4	<u>Desarrollo de la solución según metodología XP</u>	41
3.4.1	<u>Desarrollo del software</u>	41

3.4.1.1	<u>Interfaz visual</u>	41
3.4.2	<u>Herramientas de Terceros</u>	45
3.4.2.1.1	<u>ADB Shell Android</u>	45
3.4.2.1.1.1	<u>Uso de ADB Shell Android en el proyecto</u>	46
3.4.2.1.2	<u>Librería MusicG (Google)</u>	47
3.4.2.1.3	<u>Uso de la Librería MusicG (Google)</u>	48
3.4.2.1.4	<u>Google Charts (Tablas)</u>	48
3.4.3	<u>Desarrollo de la base de datos</u>	49
3.4.3.1	<u>Desarrollo de la interfaz de consulta y páginas web</u>	51
4.	<u>CAPÍTULO IV</u>	54
4.1	<u>Conclusiones</u>	54
	<u>BIBLIOGRAFÍA</u>	56
	<u>ÍNDICE DE TABLAS</u>	7
	<u>ÍNDICE DE FIGURAS</u>	8
	<u>INTRODUCCIÓN</u>	10
1.	<u>CAPÍTULO I</u>	12
1.1	<u>PLANTEAMIENTO DEL PROBLEMA</u>	12
1.2	<u>JUSTIFICACIÓN</u>	13
1.3	<u>OBJETIVOS</u>	14
1.3.1	<u>Objetivo General</u>	14
1.3.2	<u>Objetivos Específicos</u>	14
2.	<u>CAPÍTULO II</u>	15
2.1.0	<u>MARCO TEÓRICO</u>	15
2.1.1	<u>Arquitectura VoLTE</u>	15
2.1.1.1	<u>VoLTE UE</u>	15
2.1.1.2	<u>RAN (Radio Access Network)</u>	15
2.1.1.3	<u>Arquitectura logica</u>	15
2.1.2	<u>E-UTRAN (Evolved Universal Terrestrial acces network e-node b)</u>	16
2.1.3	<u>Elementos del Core IMS ePacket Core (Evolved Packet Core)</u>	17
2.1.4	<u>Tecnología 3G</u>	18
2.1.5	<u>3GPP (Third Partnership Project)</u>	20
2.2	<u>Técnicas de Análisis de Audio</u>	21

2.2.1	Técnica de Análisis de Audio Mediante Espectro Electromagnético	21
2.2.2	Huella acústica digital	21
2.3	Uso único de la Metodología Ágil de programación	22
2.3.0	Metodología Xtreme Programming XP (Programacion extrema)	22
3.	CAPÍTULO III	27
3.1	Análisis posterior y requerimientos previos según metodología XP	27
3.2	Planificación de la solución según metodología XP	27
3.3	Diseño de la solución según metodología XP	28
3.3.1	Diseño del software	28
3.3.1.1	Diseño del algoritmo para realizar llamada desde pc	29
3.3.1.2	Diseño del algoritmo para realizar grabación de llamada	31
3.3.1.3	Uso del algoritmo de análisis de audio de Shazam	32
3.3.1.4	Perfil de uso del software	33
3.3.2	Diseño de la base de datos	33
3.3.2.1	Diseño de las tablas	33
3.3.2.2	Diseño de los campos de la base de datos	34
3.3.2.3	Diseño de interfaz para presentar los datos	35
3.3.2.4	Diagrama de secuencia	35
3.4	Desarrollo de la solución según metodología XP	37
3.4.1	Desarrollo del software	37
3.4.1.1	Interfaz visual	37
3.4.2	Herramientas de Terceros	41
3.4.2.1.1	ADB Shell Android	41
3.4.2.1.1.1	Uso de ADB Shell Android en el proyecto	42
3.4.2.1.2	Librería MusicG (Google)	43
3.4.2.1.3	Uso de la Librería MusicG (Google)	44
3.4.2.1.4	Google Charts (Tablas)	44
3.4.3	Desarrollo de la base de datos	45
3.4.3.1	Desarrollo de la interfaz de consulta y páginas web	47
4.	CAPÍTULO IV	50
4.1	Conclusiones	50
	BIBLIOGRAFÍA	51

ÍNDICE DE TABLAS

<u>Tabla 1, Roles y Características []</u>	39
<u>Tabla 2, Agrupación de fases en función del ámbito que aplican []</u>	42
<u>Tabla 3 Roles dentro de la solución []</u>	44
<u>Tabla 4. Diseño de tablas [Ninguna tabla tiene clave primaria ni dependencia].</u>	51
<u>Tabla 5. Campos de las tablas</u>	52
<u>Tabla 6. Sugerencias de salida del programa por consola. Fuente: [Autor]</u>	57
<u>Tabla 1, Roles y Características []</u>	26
<u>Tabla 2, Agrupación de fases en función del ámbito que aplican []</u>	28
<u>Tabla 3 Roles dentro de la solución []</u>	30
<u>Tabla 4. Diseño de tablas [Ninguna tabla tiene clave primaria ni dependencia].</u>	37
<u>Tabla 5. Campos de las tablas</u>	38
<u>Tabla 6. Sugerencias de salida del programa por consola. Fuente: [Autor]</u>	43
<u>Tabla 1, Roles y Características []</u>	25
<u>Tabla 2, Agrupación de fases en función del ámbito que aplican []</u>	27
<u>Tabla 3 Roles dentro de la solución []</u>	29
<u>Tabla 4. Diseño de tablas [Ninguna tabla tiene clave primaria ni dependencia].</u>	36
<u>Tabla 5. Campos de las tablas</u>	37
<u>Tabla 6. Sugerencias de salida del programa por consola. Fuente: [Autor]</u>	42
<u>Tabla 1, Roles y Características []</u>	22
<u>Tabla 2, Agrupación de fases en función del ámbito que aplican []</u>	25
<u>Tabla 3 Roles dentro de la solución []</u>	27
<u>Tabla 4. Diseño de tablas.</u>	32
<u>Tabla 5. Campos de las tablas</u>	33
<u>Tabla 6. Sugerencias de salida del programa por consola. Fuente: [Autor]</u>	38

ÍNDICE DE FIGURAS

<u>Figura 1. Arquitectura Logica VoLTE. Fuente []</u>	<u>29</u>
<u>Figura 2. Arquitectura sugerida Evolved Packet Core (EPC). Fuente [].....</u>	<u>31</u>
<u>Figura 3. Arquitectura IMS. Fuente Telefónica []</u>	<u>34</u>
<u>Figura 4. Shazam App. Fuente Google PlayStore []</u>	<u>38</u>
<u>Figura 5. Metodología XP. Fuente []</u>	<u>39</u>
<u>Figura 6. Metodología XP Fases de ciclo de vida. Fuente Autor</u>	<u>41</u>
<u>Figura 7. Algoritmo básico del software. [Fuente: Autor]</u>	<u>45</u>
<u>Figura 8. Algoritmo para realizar llamada. [Fuente: Autor].....</u>	<u>47</u>
<u>Figura 9. Algoritmo de grabación en PC. [Fuente: Autor]</u>	<u>49</u>
<u>Figura 10. Algoritmo de análisis de Audio. [Fuente: Shazam herramienta de tercero]</u>	<u>50</u>
<u>Figura 11. Ejecución de Software. Fuente: [Autor].....</u>	<u>54</u>
<u>Figura 12. Visualización de datos en interfaz web. Fuente: [Autor]</u>	<u>54</u>
<u>Figura 13. Interfaz visual Sugerida. Fuente[Autor]</u>	<u>55</u>
<u>Figura 14. JFrame y botones activos. Fuente: [Autor]</u>	<u>56</u>
<u>Figura 15. JFrame y botones inactivos. Fuente: [Autor]</u>	<u>56</u>
<u>Figura 16. Salida sugerida de Consola. Fuente: [Autor].....</u>	<u>58</u>
<u>Figura 17. Confirmación de ingreso de datos sugerida. Fuente: [Autor]</u>	<u>58</u>
<u>Figura 18. Interfaz sugerida para lectura y/o consulta de base de datos. Fuente: [Autor]</u>	<u>58</u>

Figura 19. Shell ADB Android. Fuente: [Autor]	60
Figura 20. Página de alojamiento de la librería MusicG. Fuente:[Google]	62
Figura 21 Importación de librería MusicG. Fuente [Autor]	62
Figura 22. Importación de librería MusicG dentro del código. Fuente: [Autor]	62
Figura 23. Galería de Gráficos y esquemas. Fuente: []	63
Figura 24. Tablas de la base de datos. Fuente: [Autor]	64
Figura 25. Estructura de las tablas de la base de datos. Fuente: [Autor]	64
Figura 26. Ejemplo de datos consultados de la tabla "Barranquilla". Fuente: [Autor]	65
Figura 27. Página web de consulta, central de Entrerrios. Fuente: [Autor]	66
Figura 28. Página web de consulta, central de Siberia. Fuente: [Autor]	66
Figura 29. Página web de consulta, central de Pereira. Fuente: [Autor]	67
Figura 30. Página web de consulta, central de Barranquilla. Fuente: [Autor]	67
Figura 1. Arquitectura Logica VoLTE. Fuente: []	18
Figura 2. Arquitectura sugerida Evolved Packet Core (EPC). Fuente: []	19
Figura 3. Arquitectura IMS. Fuente Telefónica []	21
Figura 4. Shazam App. Fuente Google PlayStore []	25
Figura 5. Metodología XP. Fuente: []	25
Figura 6. Metodología XP Fases de ciclo de vida. Fuente Autor	28
Figura 7. Algoritmo básico del software. [Fuente: Autor]	31
Figura 8. Algoritmo para realizar llamada. [Fuente: Autor]	33
Figura 9. Algoritmo de grabación en PC. [Fuente: Autor]	35
Figura 10. Algoritmo de análisis de Audio. [Fuente: Shazam herramienta de tercero]	36
Figura 11. Ejecución de Software. Fuente: [Autor]	40
Figura 12. Visualización de datos en interfaz web. Fuente: [Autor]	40
Figura 13. Interfaz visual Sugerida. Fuente[Autor]	41
Figura 14. JFrame y botones activos. Fuente: [Autor]	42
Figura 15. JFrame y botones inactivos. Fuente: [Autor]	42
Figura 16. Salida sugerida de Consola. Fuente: [Autor]	44
Figura 17. Confirmación de ingreso de datos sugerida. Fuente: [Autor]	44
Figura 18. Interfaz sugerida para lectura y/o consulta de base de datos. Fuente: [Autor]	44
Figura 19. Shell ADB Android. Fuente: [Autor]	46
Figura 20. Página de alojamiento de la librería MusicG. Fuente:[Google]	47
Figura 21 Importación de librería MusicG. Fuente [Autor]	48
Figura 22. Importación de librería MusicG dentro del código. Fuente: [Autor]	48
Figura 23. Galería de Gráficos y esquemas. Fuente: []	49
Figura 24. Tablas de la base de datos. Fuente: [Autor]	50
Figura 25. Estructura de las tablas de la base de datos. Fuente: [Autor]	50
Figura 26. Ejemplo de datos consultados de la tabla "Barranquilla". Fuente: [Autor]	51
Figura 27. Página web de consulta, central de Entrerrios. Fuente: [Autor]	52
Figura 28. Página web de consulta, central de Siberia. Fuente: [Autor]	52
Figura 29. Página web de consulta, central de Pereira. Fuente: [Autor]	53
Figura 30. Página web de consulta, central de Barranquilla. Fuente: [Autor]	53

<u>Figura 1. Arquitectura Logica VoLTE. Fuente []</u>	17
<u>Figura 2. Arquitectura sugerida Evolved Packet Core (EPC). Fuente []</u>	18
<u>Figura 3. Arquitectura IMS. Fuente Telefónica []</u>	20
<u>Figura 4. Shazam App. Fuente Google PlayStore []</u>	24
<u>Figura 5. Metodología XP. Fuente []</u>	24
<u>Figura 6. Metodología XP Fases de ciclo de vida. Fuente Autor</u>	27
<u>Figura 7. Algoritmo básico del software. [Fuente: Autor]</u>	30
<u>Figura 8. Algoritmo para realizar llamada. [Fuente: Autor]</u>	32
<u>Figura 9. Algoritmo de grabación en PC. [Fuente: Autor]</u>	34
<u>Figura 10. Algoritmo de análisis de Audio. [Fuente: Shazam herramienta de tercero]</u>	35
<u>Figura 11. Ejecución de Software. Fuente: [Autor]</u>	39
<u>Figura 12. Visualización de datos en interfaz web. Fuente: [Autor]</u>	39
<u>Figura 13. Interfaz visual Sugerida. Fuente[Autor]</u>	40
<u>Figura 14. JFrame y botones activos. Fuente: [Autor]</u>	41
<u>Figura 15. JFrame y botones inactivos. Fuente: [Autor]</u>	41
<u>Figura 16. Salida sugerida de Consola. Fuente: [Autor]</u>	43
<u>Figura 17. Confirmación de ingreso de datos sugerida. Fuente: [Autor]</u>	43
<u>Figura 18. Interfaz sugerida para lectura y/o consulta de base de datos. Fuente: [Autor]</u>	43
<u>Figura 19. Shell ADB Android. Fuente: [Autor]</u>	45
<u>Figura 20. Página de alojamiento de la librería MusicG. Fuente:[Google]</u>	46
<u>Figura 21 Importación de librería MusicG. Fuente [Autor]</u>	47
<u>Figura 22. Importación de librería MusicG dentro del código. Fuente: [Autor]</u>	47
<u>Figura 23. Galería de Gráficos y esquemas. Fuente: []</u>	48
<u>Figura 24. Tablas de la base de datos. Fuente: [Autor]</u>	49
<u>Figura 25. Estructura de las tablas de la base de datos. Fuente: [Autor]</u>	49
<u>Figura 26. Ejemplo de datos consultados de la tabla "Barranquilla". Fuente: [Autor]</u>	50
<u>Figura 27. Página web de consulta, central de Entrerrios. Fuente: [Autor]</u>	51
<u>Figura 28. Página web de consulta, central de Siberia. Fuente: [Autor]</u>	51
<u>Figura 29. Página web de consulta, central de Pereira. Fuente: [Autor]</u>	52
<u>Figura 30. Página web de consulta, central de Barranquilla. Fuente: [Autor]</u>	52
<u>Figura 1. Arquitectura Logica VoLTE. Fuente []</u>	16
<u>Figura 2. Arquitectura sugerida Evolved Packet Core (EPC). Fuente Autor y []</u>	17
<u>Figura 3. Arquitectura IMS. Fuente Telefónica []</u>	19
<u>Figura 4. Shazam App. Fuente Google PlayStore []</u>	22
<u>Figura 5. Metodología XP. Fuente []</u>	23
<u>Figura 6. Metodología XP Fases de ciclo de vida. Fuente Autor</u>	25
<u>Figura 7. Algoritmo básico del software. [Fuente: Autor]</u>	29
<u>Figura 8. Algoritmo para realizar llamada. [Fuente: Autor]</u>	30
<u>Figura 9. Algoritmo de grabación en PC. [Fuente: Autor]</u>	31
<u>Figura 10. Algoritmo de análisis de Audio. [Fuente: Shazam herramienta de tercero]</u>	32
<u>Figura 11. Ejecución de Software. Fuente: [Autor]</u>	36
<u>Figura 12. Visualización de datos en interfaz web. Fuente: [Autor]</u>	36

<u>Figura 13. Interfaz visual Sugerida. Fuente:[Autor]</u>	37
<u>Figura 14. JFrame y botones activos. Fuente: [Autor]</u>	38
<u>Figura 15. JFrame y botones inactivos. Fuente: [Autor]</u>	38
<u>Figura 16. Salida sugerida de Consola. Fuente: [Autor]</u>	40
<u>Figura 17. Confirmación de ingreso de datos sugerida. Fuente: [Autor]</u>	40
<u>Figura 18. Interfaz sugerida para lectura y/o consulta de base de datos. Fuente: [Autor]</u>	40
<u>Figura 19. Shell ADB Android. Fuente: [Autor]</u>	42
<u>Figura 20. Página de alojamiento de la librería MusicG. Fuente:[Google]</u>	43
<u>Figura 21 Importación de librería MusicG. Fuente [Autor]</u>	44
<u>Figura 22. Importación de librería MusicG dentro del código. Fuente: [Autor]</u>	44
<u>Figura 23. Galería de Gráficos y esquemas. Fuente: []</u>	45
<u>Figura 24. Tablas de la base de datos. Fuente: [Autor]</u>	46
<u>Figura 25. Estructura de las tablas de la base de datos. Fuente: [Autor]</u>	46
<u>Figura 26. Ejemplo de datos consultados de la tabla "Barranquilla". Fuente: [Autor]</u>	47
<u>Figura 27. Página web de consulta, central de Entrerrios. Fuente: [Autor]</u>	48
<u>Figura 28. Página web de consulta, central de Siberia. Fuente: [Autor]</u>	48
<u>Figura 29. Página web de consulta, central de Pereira. Fuente: [Autor]</u>	49
<u>Figura 30. Página web de consulta, central de Barranquilla. Fuente: [Autor]</u>	49

TERMINOS Y DEFINICIONES

<u>Termino</u>	<u>Definición</u>
<u>3GPP</u>	<u>3rd Generation Partnership Project</u>
<u>A-SBC</u>	<u>Access Session Border Controller</u>
<u>ACR</u>	<u>Anonymous Call Rejection</u>
<u>AMBR</u>	<u>Aggregate Maximum Bit Rate</u>
<u>AMR</u>	<u>Adaptive Multi-Rate</u>
<u>API</u>	<u>Application Programming Interface</u>
<u>APN</u>	<u>Access Point Name</u>
<u>BGCF</u>	<u>Border Gateway Control Function</u>

<u>CAMEL</u>	<u>Customised Application for Mobile network Enhanced Logic</u>
<u>CAP</u>	<u>CAMEL Application Part</u>
<u>CDR</u>	<u>Charging Data Record</u>
<u>CSCF</u>	<u>Call Server Control Function</u>
<u>CSFB</u>	<u>Circuit Switched Fall Back</u>
<u>DL</u>	<u>DownLink</u>
<u>DNS</u>	<u>Domain Name System</u>
<u>e2e</u>	<u>end to end</u>
<u>ENUM</u>	<u>E.164 Number Mapping</u>
<u>EPC</u>	<u>Evolved Packet Core</u>
<u>EPS</u>	<u>Evolved Packet System</u>
<u>eSRVCC</u>	<u>Enhanced Single Radio Voice Call Continuity</u>
<u>E-UTRAN</u>	<u>Evolved Universal Terrestrial Access Network</u>
<u>GPRS</u>	<u>General Packet Radio Service</u>
<u>GSM</u>	<u>Global System for Mobile communications</u>
<u>HLR</u>	<u>Home Location Register</u>
<u>HSPA</u>	<u>High Speed Packet Access</u>
<u>I-CSCF</u>	<u>Interrogating Call Session Control Function</u>
<u>I-SBC</u>	<u>Interconnect Session Border Controller</u>
<u>IMEI</u>	<u>International Mobile Equipment Identity</u>
<u>IMS</u>	<u>IP Multimedia Subsystem</u>
<u>IMSI</u>	<u>International Mobile Subscriber Identity</u>
<u>IOT</u>	<u>Interoperability Testing</u>

<u>IP</u>	<u>Internet Protocol</u>
<u>IP-CAN</u>	<u>IP-Connectivity Access Network</u>
<u>ISUP</u>	<u>ISDN User Part</u>
<u>LTE</u>	<u>Long Term Evolution</u>
<u>MAC</u>	<u>Medium Access Control</u>
<u>MAP</u>	<u>Mobile Application Part</u>
<u>ME</u>	<u>Mobile Equipment</u>
<u>MME</u>	<u>Mobility Management Entity</u>
<u>MMS</u>	<u>Multimedia Messaging Service</u>
<u>MSISDN</u>	<u>Mobile Subscriber ISDN Number</u>
<u>NAT</u>	<u>Network Address Translation</u>
<u>P-CSCF</u>	<u>Proxy Call Session Control Function</u>
<u>PCC</u>	<u>Policy and Charging Control</u>
<u>PGW</u>	<u>Packet Data Network Gateway</u>
<u>QoS</u>	<u>Quality of Service</u>
<u>RAN</u>	<u>Radio Access Network</u>
<u>RLC</u>	<u>Radio Link Control</u>
<u>RTP</u>	<u>Real-time Transport Protocol</u>
<u>S-CSCF</u>	<u>Serving Call Session Control Function</u>
<u>SBC</u>	<u>Session Border Controller</u>
<u>SGW</u>	<u>Serving Gateway</u>
<u>SIGCOMP</u>	<u>Signalling Compression</u>
<u>SIP</u>	<u>Session Initiation Protocol</u>
<u>SIP-I</u>	<u>SIP with encapsulated ISUP</u>
<u>SMS</u>	<u>Short Message Service</u>
<u>SON</u>	<u>Self-Organising Networks</u>

<u>SRVCC</u>	<u>Single Radio Voice Call Continuity</u>
<u>TAS</u>	<u>Telephony Application Server</u>
<u>TAI</u>	<u>Tracking Area Identity</u>
<u>TCP</u>	<u>Transmission Control Protocol</u>
<u>TDD</u>	<u>Time Division Duplex</u>
<u>TDM</u>	<u>Time Division Multiplexing</u>
<u>UE</u>	<u>User Equipment</u>
<u>UL</u>	<u>Uplink</u>
<u>UMTS</u>	<u>Universal Mobile Telecommunications System</u>
<u>URI</u>	<u>Uniform Resource Identifier</u>
<u>USIM</u>	<u>Universal Subscriber Identity Module</u>
<u>UTRAN</u>	<u>Universal Terrestrial Access Network</u>
<u>VLR</u>	<u>Visitor Location Register</u>
<u>VoHSPA</u>	<u>Voice over HSP</u>
<u>VoLTE</u>	<u>Voice over LTE</u>
<u>XCAP</u>	<u>XML Configuration Access Protocol</u>

|

INTRODUCCIÓN

Las telecomunicaciones han trascendido a través de los siglos “acercando” cada vez más a aquellos a quienes necesitan de ellas. Es por esto que cada día es indispensable innovar y reinventar en las mismas además de darles el siguiente salto para beneficio nuestro. [La historia de las telecomunicaciones](#) Partiendo parte desde la electricidad, pasando por la invención del teléfono, el correo convencional, la televisión, las señales de radiofrecuencia, la internet y por último las generaciones de tecnología celular que han abierto muchos horizontes para el despliegue de nuevas invenciones e incluso la implementación de todo lo anterior en base a una sola tecnología unificada, lo cual es el objetivo principal de las tecnologías actuales.

La tecnología celular [móvil](#) nos ha permitido avanzar más rápidamente en las interacciones sociales, al igual que el aumento y el compartir del conocimiento en general. Es por esto que las que se usan actualmente otorgan la facilidad de llevar datos y multimedia a todas partes de una forma sencilla y cómoda. Para ello fue necesario evolucionar desde [la primera -arquitecturageneración 2G-](#)^[JA1] la cual nos permitía conectar charlas de voz con un punto de inicio y un punto final, ~~ésteeste~~ tipo de generación celular nos permitía un máximo de 14,4 Kbps y fue el primero en adoptar el sistema GSM (Sistema Global de Comunicaciones Móviles) lo que lo llevo a evolucionar en su correspondiente 3G, la cual contaba ya con una velocidad mayor. Gracias a la implementación de la tecnología W-CDMA se logró llegar a una velocidad de 2 Mbps, lo que a su vez logró la implementación de multimedia y la apertura de vías a la implementación y navegación de internet desde un dispositivo celular.

La evolución móvil 4G, trae con ella muchas ventajas para la navegación multimedia, ya que se pensó para ser implementada en conjunto con todos los servicios que se podrían prestar, como las conferencias, video llamadas y navegación por internet. ~~Se tiene pensada~~ [Los operadores, tienen pensado](#)^{[JA2][JSRU3]} ~~para~~ superar los 20 Mbps como velocidad en movimiento y hasta 1Gbps en estado estático o sin movimiento, sin embargo, esta tecnología es la que se tiene actualmente y presenta algunas fallas pues no está completamente desplegada en el mundo. Una ventaja [que cabe](#) a resaltar de esta generación es la integración con el conjunto de servicios y paquetes que entregan las compañías de telecomunicaciones utilizando señalizaciones especiales que les permite una mayor

facilidad al momento de gestionar sus recursos, entre estos se encuentran algunos como SIP, CAMEL, SS7 y MAP^[JA4]^[JSRU5].

La integración de los servicios más reconocidos está dada dentro de un conjunto de dispositivos llamados Core IMS (Subsistema multimedia IP), el cual ha estado constantemente evolucionando y se ha integrado sistemas como internet, voz, multimedia, televisión, servicios de primer nivel^[JA6] y todos migrados sobre el protocolo de internet (IP) lo que permite a todos los proveedores una mayor conectividad, a los usuarios mayor conectividad y facilidad al momento de ubicar posibles fallas que se presenten en el sistema.

La migración de la tecnología a 5G permite a las comunicaciones un gran salto hacia el futuro, y dentro de este se encuentra todos los servicios en base al protocolo IP en el cual se encuentran servicios como VoLTE, IPTV, IoT entre otros. En este caso la mayor importancia recae sobre VoLTE, despliegue que nos permite realizar llamadas de voz mediante el protocolo de internet siempre y cuando el dispositivo lo soporte.⁵ Entre los dispositivos que permiten esta tecnología se encuentra el Core IMS sobre el cual recae la responsabilidad de alojar a los usuarios para su continua comunicación.

La jefatura Core voz CICO es un área dentro de las áreas técnicas del Operador Telefónica encargada de administrar, monitorear y encargarse de todas aquellas funciones relacionadas con el tráfico de voz,⁵ ~~siendo éste un~~ La unidad la conforma un equipo técnico de ingenieros, que pueden atender casos relacionados desde fallos de llamadas, hasta elementos relacionados con casos de equipos que fuesen hechos para el tráfico mismo de voz.

1. CAPÍTULO I

1.1 PLANTEAMIENTO DEL PROBLEMA

La empresa de telecomunicaciones Telefónica es uno de los mayores impulsores de la economía digital del país, hoy en día presta diferentes servicios de telecomunicaciones a empresas y usuarios haciendo de ésta una marca conocida a nivel mundial en el medio. Uno de los servicios más característicos que presta la empresa es el de la telefonía y conectividad móvil, presente en Colombia en “957 municipios con telefonía móvil y en 341 ofrece la tecnología 4G ~~LTE~~” [1].

~~Dentro del~~ QoS (Quality of Service) ~~se encuentra el~~ tiene el objetivo ~~objetivo~~ de entregar una óptima calidad de servicio a los usuarios al igual que los servicios que ellos contratan, para ello se existe la jefatura Core Voz CICO la cual está encargada de atender casos que se presenten en toda la línea de voz. El equipo está enfocado a dar solución dentro del sistema IMS correspondiente a voz, lo que lleva a la problemática presentada relacionada con la tecnología IMS, la cual implementa dentro de su arquitectura para lograr llevar datos y multimedia a sus usuarios [JA7][JSRU8] los dispositivos denominados MGW (Media Gateway) con ayuda de los

(centrales) MSS y permiten enrutar los mensajes de anuncios con el número que tiene asociado los dígitos del usuario, posterior a esto el mensaje es asociado con un tipo de servicio o caso el cual se llama y es mostrado finalmente al usuario en la llamada hacia el Core. El dispositivo llamado como MSS permite además alojar los ~~mensajes de anuncio en conjunto con los MGW's, y estos a su vez son los que informan a los usuarios del contexto dentro del cual se encuentran en el sistema y les permiten acceder a opciones como desde el saldo disponible, menús que permitan recargar el saldo del teléfono, hasta grabaciones que indiquen el estado del servicio actual. Sin embargo, estas grabaciones no siempre puede que se encuentren bien alojadas y adecuadas dentro del MSS, pues pasan por varios procesos de conversión hasta llegar a su fase final de alojamiento en él.~~ Las grabaciones pueden tener un porcentaje de error al ser llevadas de un proceso a otro y para ello es necesario comprobar que tanto la calidad de entendimiento de la grabación como el tiempo y el audio se lleven a cabo correctamente y sean óptimas para su escucha, por eso se plantea: ¿Cómo comprobar la calidad de las grabaciones de los mensajes de anuncios en los MSS directamente desde un dispositivo móvil celular?

1.2 JUSTIFICACIÓN

Dentro de las empresas de telecomunicaciones se encuentran [n funciones](#) desde los principios de calidad y responsabilidad hasta la aplicación de resolución a todo tipo de casos y fallas haciendo uso de técnicas y prácticas aprendidas a lo largo de los años. La calidad del servicio es indispensable para las empresas y estas utilizan metodologías ágiles para encontrar tanto los problemas como sus correspondientes soluciones, es necesario que todos los empleados y empleadores trabajen en conjunto para brindar un buen servicio.

Las empresas como ~~Telefónica~~, [Telefónica](#) realizan periódicamente cambios en sus dispositivos lo que en buenas prácticas como ITIL se conoce como ventanas del servicio, acordadas en conjunto de los proveedores y les permite detener temporal y periódicamente el servicio para realizar mantenimientos y cambios que beneficien a todos y que además permiten la continuidad del servicio.

En el caso apropiado, se tiene en cuenta el dispositivo MGW que hace parte de la arquitectura de la tecnología actual implementada IMS que está constantemente siendo actualizado y monitorizado para su buena calidad y servicio. Este dispositivo brinda un enrutamiento y permite la conectividad hacia el Core y en este caso brinda alojamiento a los mensajes de información y anuncios que permiten interactuar e informar a los usuarios sobre sus productos y servicios disponibles. Sin embargo, debido a que pasan por muchos procesos para finalmente ser alojados, su composición original puede ser alterada y no podrían presentar la originalidad con la que fueron diseñados y consecuentemente no pueden ser entendidos claramente por los usuarios.

El diseño e implementación de un sistema que permita monitorizar además de comprobar puede permitir al área de Core de Voz CICO tener más control después de cada ventana de mantenimiento o ventana de actualización que se realice sobre el sistema asimismo verificar si los archivos que están encontrados dentro de los MGW's no se encuentran desfasados en tiempo y son entendibles para los usuarios. Para acelerar el proceso es necesario un marcador virtual que se comunique con el celular y permita realizar llamadas desde el ordenador.

1.3 OBJETIVOS

1.3.1 Objetivo General

Diseñar e implementar un sistema de comprobación de los mensajes de anuncios alojados dentro de los MSS's.

1.3.2 Objetivos Específicos

- Desarrollar una metodología de comparación de audio para encontrar similitudes entre las grabaciones y los mensajes alojados en los MSS's.
 - Hacer un sistema de gestión que permita hacer llamadas a cualquier destino desde el ordenador y colgarlas respectivamente.
 - Crear un esquema que permita observar el estado de los mensajes de anuncio desde una interfaz web.
 - Desarrollar un sistema de monitorización para observar el estado de los mensajes de anuncio.
-

4.2. CAPÍTULO II

2.1.0 MARCO TEÓRICO

2.1.1 Arquitectura VoLTE

La arquitectura VoLTE está basada en la arquitectura lógica dada por el 3GPP y define algunas partes específicas como lo son VoLTE UE (VoLTE User Equipment), EPC (Evolved Packet Network) y el Core ~~IMS~~IMS [2].

2.1.1.1 VoLTE UE

El VoLTE UE contiene funcionalidad para acceder a la red de acceso de LTE (RAN LTE) asimismo como a la EPC y permite conectividad en banda ancha, todo se maneja en la misma señalización para hacer uso de los recursos del Core IMS y éste pueda llevarlos a comunicarse con las demás partes del medio.

2.1.1.2 RAN (Radio Access Network)

En este caso el acceso a la red se realiza gracias a las características que tiene LTE en términos de radio acceso como FDD LTE, TDD LTE.

2.1.1.3 Arquitectura logica

A continuación, se muestra la arquitectura logica que propone IMS en conjunto a la red de 4G LTE:

Figura 1. Arquitectura Logica VoLTE. Fuente [3]

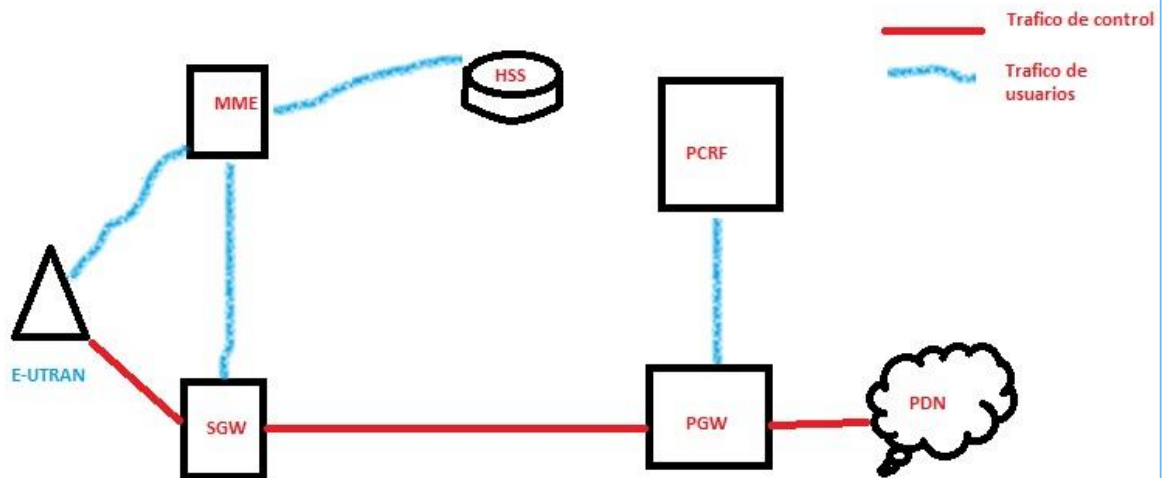
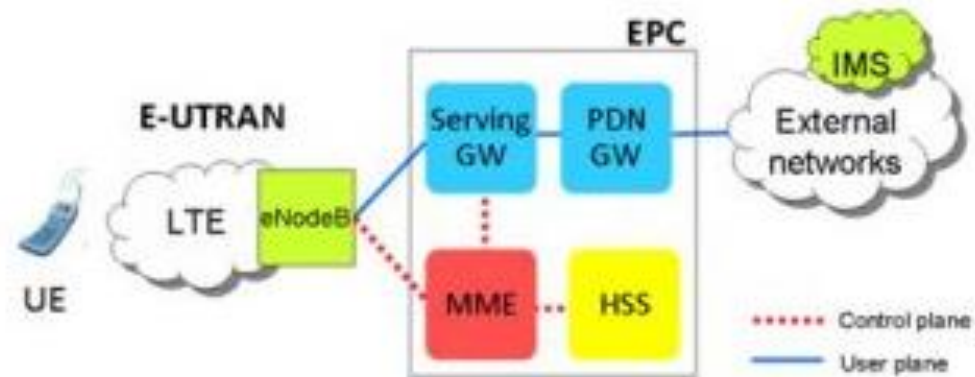
La arquitectura esta creada en base a tres diferentes tipos de partes, la primera siendo el acceso a la red, la segunda la distribución del tráfico, y la tercera está dada como el Core de la red donde el ~~tráfico~~tráfico completo se carga y es procesado para estar a disposición de los demás recursos.

2.1.2 E-UTRAN (Evolved Universal Terrestrial acces network e-node b)

En este caso el acceso al medio está dado por el e-NodeB éste se compone de las interfaces físicas del medio y es el puente entre los dispositivos móviles y la estación base de telecomunicaciones móviles en LTE. Cada interfaz física otorga una dirección MAC la cual es única para cada dispositivo, dentro de la misma interfaz se encuentra el Radio Link Control (RLC), además de permitir la conectividad inalámbrica por radio permite cifrar los datos de los usuarios y permite conectividad de subida (Uplink) y conectividad de bajada (DownLink).

2.1.3 Elementos del Core IMS ePacket Core (Evolved Packet Core)

Dentro del Core IMS se encuentran varios equipos y dispositivos que permiten la conectividad IMS con los recursos del resto de la arquitectura IMS:



[JA10]

Figura 2. Arquitectura sugerida Evolved Packet Core (EPC). Fuente: ~~Autory~~ [4].

2.1.3.1 MME (Mobility Management Entity)

El MME es el dispositivo más importante que permite el control sobre los nodos de radio acceso en LTE. Es la parte encargada de administrar el seguimiento que se realiza sobre los UE (User Entity) aloja a los usuarios locales y les permite alcanzar

los estados idle¹ y paging²_{[JA11][JSRU12]}. En otra parte es responsable además de realizar un bearer o tubo lógico que permite la autenticación de los usuarios con el HSS (Home subscriber server), lo que permite validar los permisos correspondientes de los usuarios o UE con respecto al proveedor de servicio.

2.1.3.2 SGW (Serving Gateway)

El SGW enruta los paquetes del usuario además de permitirle anclarlo a la sesión y asimismo el realiza la tarea de hacer handover a los usuarios dentro de los eNodeb's y dentro del resto de tecnologías como 2G/3G. El SGW está encargado de administrar el nivel de tráfico que manejan los eNodeb's además de ser un único elemento en la arquitectura.

2.1.3.3 HSS (Home Subscriber Server)

El HSS es el elemento más importante_{[JA13][JSRU14]} de la arquitectura, pues almacena, y aloja y entrega los registros y servicios de los usuarios, así como los correspondientes registros de IMS de los mismos y los entrega hacia otros elementos de la red, como el MME y el Core IMS.

2.1.4 Tecnología 3G

La arquitectura básica de la tecnología 3G se muestra en la figura 2:

¹ Estado idle: Cuando el dispositivo móvil es recién encendido y pasa a estar anclado al nodo o celda.

² Paging Channel: avisa al dispositivo cuando entra una llamada entrante.

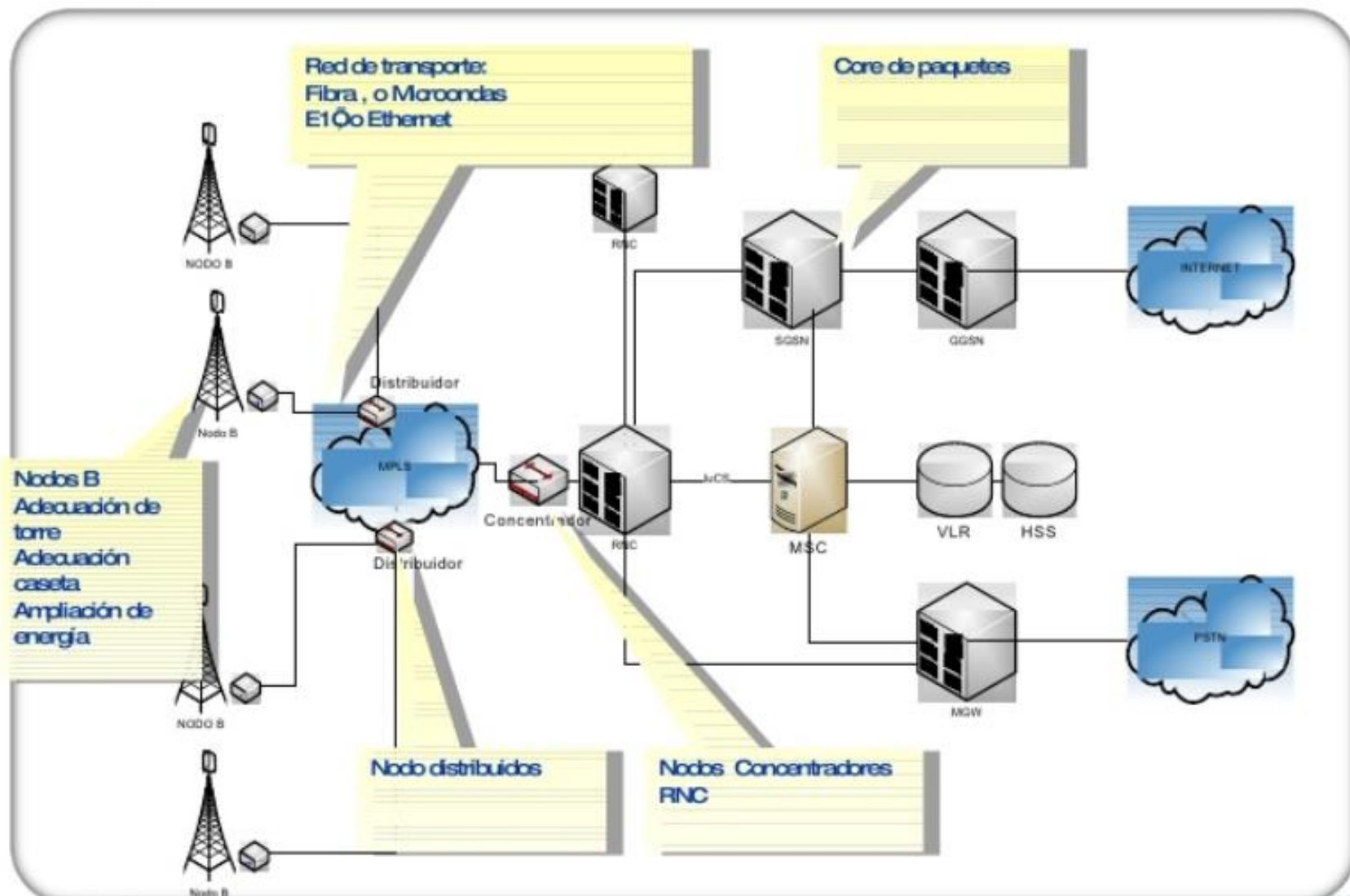


Figura 3. Arquitectura IMS. Fuente Telefónica [5]

Al igual que en la arquitectura IMS cuando se creó la arquitectura 3G se planeó en base a tres principales puntos: Acceso, Distribución y Core.

2.1.4.1 UTRAN

Este ~~elemento a parte~~ de la arquitectura 3G hace parte del acceso hacia la red y permite a los usuarios engancharse, autenticarse, e ingresar donde se encuentran alojados sus datos según sea su operador de servicios. Está compuesto por el dispositivo BSC (Base Station Controller), administrando otros como las BTS que hacen parte del grupo de estaciones móviles y de base. Para realizar su tarea las estaciones base de dos o más frecuencias portadoras multiplexan en el tiempo y transportan el tráfico de datos o señalización de los usuarios. Las BSC pueden administrar un número grande de BTS éstos, se comunican con las estaciones base a través de interfaces Abiss lo que les permite distribuir el tráfico con mayor facilidad al resto de la red GSM. Los registros del sistema de información se realizan sobre dispositivos especializados como HLR (Home Location Register) donde se realiza la base de datos de información de los abonados de la red [6].

2.1.4.2 MSC (Mobile Switching Center)

En la arquitectura 3G se conocen como centrales de switching de voz en las redes móviles. Las centrales tienen comunicación con los RNCs y los BCS, a diferencia de la tecnología 4G las llamadas que hacen uso de los datos, se trazan por un camino diferente haciendo parte del MGW.

2.1.4.3 MGW (Media GateWay)

Este es un dispositivo que hace como traductor entre varios tipos de tecnologías, señalizaciones y datos dentro de las cuales se encuentran 2G, 3G, 4G, NGN (Next Generation Networks). Estos dispositivos además realizan la comunicación multimedia a través de redes de paquetes asíncronos ATM en conjunto del protocolo de internet. Los MGWs permiten la conectividad entre varios tipos de tecnologías y redes y por lo tanto son controlados por dispositivos administradores que entregan control de llamadas y de señalización es por esto ~~que por lo que~~ deben tener la capacidad de traducir entre diferentes tipos de señalizaciones y códecs como lo son

Megaco (h.248), SIP y SS7, lo que les permite funcionar como Smart MGWs. En el caso de 4G y VoLTE, los MGWs establecen una conversión en división por multiplexación de tiempo (TDM) y por RTP (Real Time Protocol) para ser enviados posteriormente hacia los usuarios. Los MGWs utilizan el estándar del grupo 3GPP que son funcionalmente estandarizados y permiten alojar de forma continua los mensajes de anuncio para gestión y ayuda de usuarios.

2.1.5 3GPP (Third Partnership Project)

Es una organización encargada de estandarizar y publicar los procedimientos y protocolos necesarios para llevar a cabo una buena conectividad para tecnologías como: GSM, 3G (UMTS y HSPA), 4G (LTE, LTE advance, y LTE advance Pro) y 5G (NGN) y sistemas como IMS [7].

La organización en primera instancia liberó el Release 6 acorde a la tecnología HSPA+ y a la tecnología LTE la cual es tomada en cuenta para la creación del reléase 7 y permite la liberación y creación del grupo 3GPP2 para acoger sistemas como CDMA. Actualmente se encuentra en el reléase 15 el cual fue liberado en 2018 donde se estandariza la mejora para sistemas IMS para comunicaciones móviles dentro de vehículos en caminos de alto tráfico.

2.2 Técnicas de Análisis de Audio

2.2.1 ~~Técnica de Análisis de de muestras de Audio Mediante Espectro Electromagnético digital o~~

~~El espectro electromagnético hace parte fundamental de las ondas electromagnéticas pues es la representación al pasar del dominio del tiempo al dominio de la frecuencia y con su componente correspondiente de potencia. En este caso, el~~ El espectro audible para los seres humanos se encuentra entre los 20 Hz y 20 KHz (de un oído sano), el espectro se divide en 11 partes de aproximadamente igual proporción llamadas octavas. En el caso del audio digital muestreado, permiten analizar el audio por separado en comparación a hacerlo por completo, pues el audio muestreado se puede clasificar igualmente en octavas: La primera y segunda octava se encuentra dentro del espectro de 16 Hz a 64 Hz, la tercera y cuarta octava

se encuentran dentro de las frecuencias desde 64 Hz a 250 Hz, la quinta, sexta y séptima octava son clasificados como tonos medios dentro de las frecuencias de 250 Hz a 2 KHz la octava acústica se encuentra dentro de 2 KHz hasta 4096 Hz, la novena y décima octava se encuentran dentro del rango de frecuencias 4097 Hz hasta 16 000 Hz a partir de estas frecuencias se consideran sonidos muy agudos y molestos para algunas personas y la undécima octava se encuentra con rango de frecuencias desde 16 000 Hz en adelante y se encuentran frecuencias muy poco audibles [8].^[JA17]^[JSRU18]

2.2.2 Huella acústica digital

Los análisis para detección y comparación de audio se han convertido en algo tanto comercial como forense, lo que ha permitido desarrollar varios tipos de algoritmos que permiten identificar patrones conocidos en ondas de sonido, pasándolas al dominio de la frecuencia y realizando la correspondiente comparación de potencia con respecto a otras. La huella acústica es conocida como el resumen digital de una señal de audio.

Para el oído humano, si dos audios ‘suenan iguales’ o son similares sus huellas acústicas deben coincidir en un porcentaje mayor que en un caso donde no sean similares. Algunas de las características que son tomadas para calcular la huella acústica, son el porcentaje de **retorno a cero**, tempo, espectro estimado, **llanura espectral**, tonos altos y prominentes de la señal acústica y ancho de banda de frecuencia.

2.2.2.1 Espectrograma

Un espectrograma es una firma de audio esencial representada en el dominio de la frecuencia. Se tiene estimado que cualquier señal de audio es capaz de ser pasada a este dominio y es representado por frecuencia | amplitud | tiempo [9].

2.2.2.2 Shazam

Shazam está enfocado en ubicar e identificar los picos de potencia alta que presenta la señal con respecto a las demás partes de la señal, esto le permite reducir altamente el ruido de fondo en identificación de audio, por lo que básicamente es utilizado para detectar grabaciones con respecto a señales de audio base almacenadas en una base de datos previamente cargada. Shazam realiza varias marcas de picos de energía lo que le permite detectar un rango de intensidad de energía, esta técnica le permite al algoritmo detectar picos iguales o similares de energía en ambos archivos de audio y detectar cuales están relativamente cerca unos de otros para la comparación.



Figura 4. Shazam App. Fuente Google PlayStore [10]

2.3 Uso único de la Metodología Ágil de programación

En el proyecto se hará uso únicamente de una metodología ágil, sin embargo, a lo largo del mismo se pueden hacer combinaciones pequeñas con algunas otras. Cabe aclarar que la principal será la metodología XP para la programación y desarrollo del software.

2.3.0 Metodología Xtreme Programming XP (Programacion extrema)

La extreme programming es una metodología que hace parte de una metodología ágil como Scrum que tiene como objetivo principal dedicar control, flexibilidad y precisión a los proyectos sobre los que recae.

Metodología XP – Programación Extrema



Figura 5. Metodología XP. Fuente [11]

Las características principales que definen la metodología XP son considerar al equipo como factor principal de éxito, de igual forma el software o proyecto funciona por encima de una buena documentación, la interacción constante entre el equipo desarrollador y el cliente, su planificación es flexible y totalmente abierta, así como su respuesta a los cambios que se hagan al proyecto dentro de su transcurso.

Tabla 1, Roles y Características [12]

Roles	Características
Cliente	Es el encargado de manejar el proyecto hacia los objetivos trazados.
Programadores	Son los encargados de llevar a cabo el proyecto así como <u>sus como sus</u> tiempos de desarrollo.
Tester	Es el encargado de realizar pruebas al proyecto
Tracker	Es el encargado de darle un seguimiento al proyecto, la programación y sus objetivos.

Coach	Es el encargado de orientar al equipo y saber cómo lograr los objetivos trazados.
Big Boss o Manager	Es el encargado del proyecto, tiene total idea tanto del proyecto como de sus objetivos y estatus.

2.3.0.1 Pruebas del proyecto

Con respecto a la forma de llevar el proyecto se debe llevar a cabo objetivo por objetivo, según lo que el cliente vaya exigiendo a lo largo del desarrollo del proyecto, los objetivos del proyecto se deben cumplir de forma orientada, muchas veces teniendo en cuenta que no se podrían cumplir con los cronogramas establecidos, sin embargo, mientras se cumplan de forma objetiva se acepta, cabe aclarar que no va de sobra tener en cuenta las fechas de entregas periódicas del proyecto pues esto permite tener un control mayor sobre los objetivos y retos que se presenten. El cliente es el que va trazando el curso del proyecto a medida que se van estableciendo, cumpliendo y orientando los objetivos del proyecto, esto permite a los desarrolladores encaminar mejor el futuro del proyecto.

2.3.0.2 Diseño e implementación del proyecto

El diseño e implementación del proyecto se puede realizar mediante la colaboración de otro programador si es necesario, en tal caso se llama programación en conjunto y es permitido compartir parte del código del proyecto, para permitirse correcciones rápidas y transparencia al momento de las compilaciones. Compartir el código de forma grupal permite realizar pruebas automáticas en partes específicas del proyecto y colocar una forma única de hacer los códigos.

2.3.0.3 Características únicas de la metodología XP

Los nombres de los elementos del proyecto no tienen que ser estrictamente 'formales' pueden ser referidos y conocidos por los actores programadores del proyecto. La metodología transparente que se permite con XP esta facilitada para que los programadores saquen a flote más rápidamente los objetivos del proyecto y es efectivo cuando el cliente necesita resultados rápidamente. Esta metodología es mayormente adoptada por las personas que necesitan resultados rápidos y

programadores no tan 'formales' ~~a pesar que~~ a pesar de que esta metodología está siendo adoptada mayormente en las empresas grandes.

2.3.0.4 Ciclo de la metodología XP

La metodología ágil XP cuenta con cuatro fases de ejecución que le permiten desarrollar el proyecto de forma ordenada:

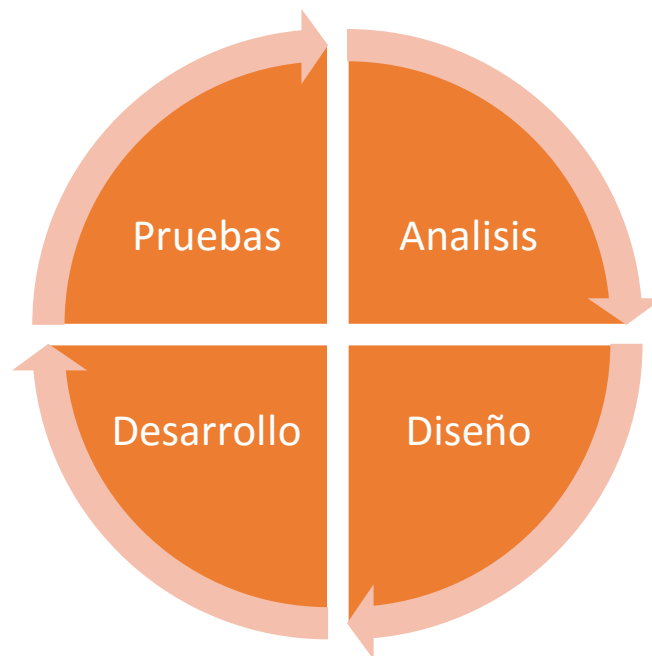


Figura 6. Metodología XP Fases de ciclo de vida. Fuente Autor

Para dar a cabo el proyecto es necesario entender lo que necesita el cliente, para ello se realiza la *fase de exploración*, luego de plantear que es lo que necesita el cliente se procede a realizar la *fase de planificación* donde se estima que cantidad de objetivos se van a tocar a lo largo del proyecto y cuáles serán estos, para comenzar a realizar el proyecto se realiza la *fase de iteraciones* para formalizar la solución que se pensará crear al cliente y se comienza la planificación de proyecto, en la última fase, *la fase de puesta en producción* [13] se desarrolla el proyecto, se realizan correcciones, ajustes, nuevos objetivos y hasta eliminación de algunos otros para realizar la entrega final del proyecto al cliente.

Tabla 2, Agrupación de fases en función del ámbito que aplican [14]

Extreme Programing	Planificación	Historias de usuarios
		Plan de entregas
		Plan de iteraciones
		Reuniones diarias de seguimiento
	Diseño	Simplicidad
		Soluciones 'Spike'
		Recodificación
		Metáforas
	Desarrollo	Disponibilidad del cliente
		Uso de estándares
		Programación dirigida por las pruebas
		Programación en dos manos
		Integraciones permanentes
		Propiedad colectiva del código
		Ritmo sostenido
	Pruebas	Pruebas unitarias
		Protección y corrección de errores
		Pruebas de aceptación

13.3. CAPÍTULO III

13.13.1 Análisis posterior y requerimientos previos según metodología XP

Observando lo anterior mostrado en el marco teórico, y lo presentado en el análisis sobre la problemática que se está trabajando sobre la jefatura Core Voz y CICO para la comprobación de los mensajes de anuncio sobre los MGW's, se exponen las soluciones que se pretenden implementar para permitir al área, un trabajo más eficiente ~~y óptimo~~ en cuanto a resolución de problemáticas de nivel técnico o nivel 1. Para llegar a dar solución es necesario analizar los siguientes temas:

- Software que permita comprobar los mensajes de anuncio
- Base de datos que almacene los resultados obtenidos del software
- Documentación que contenga las bases del funcionamiento del software

Dentro del software se encuentran algunos objetivos a cumplir como:

- Grabación de llamada para análisis posterior de audio
- Iniciar llamada desde el PC, que representa el terminal que utiliza el operario [JA19][JSRU20] hacia el dispositivo móvil igualmente manipulado por el operario.
- Búsqueda de algoritmo que permita rápido análisis y comparación entre dos archivos de audio.
- Búsqueda de lenguaje de programación más adecuado para el trabajo.

Dentro de la base de datos se encuentran los siguientes objetivos:

- Parámetros de las tablas
- Organización de los datos de la base de datos

- Interfaz para mostrar los datos a los usuarios finales.

NOTA: Para realizar esto, se tiene pensado implementar la metodología XP.

13.23.2 Planificación de la solución según metodología XP

En la fase de planificación se tiene estipulado mostrar todos y cada uno de los protagonistas dentro del proceso de la solución para ello se explicará de la siguiente manera:

Tabla 3 Roles dentro de la solución [15]

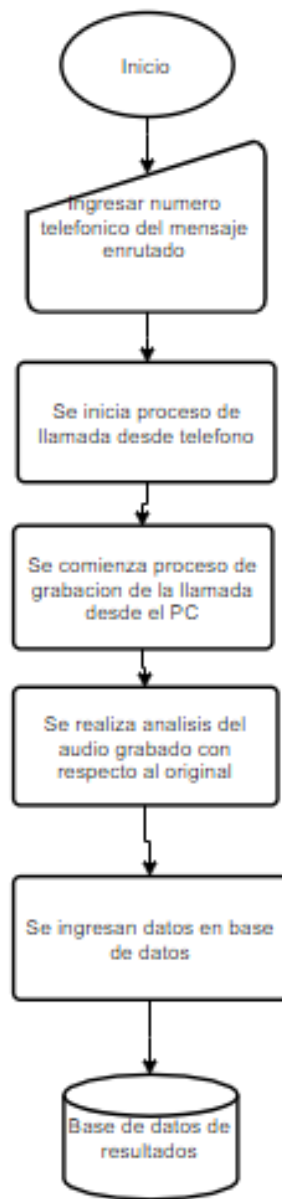
Roles	Características
Cliente	Jefatura Core Voz CICO
Programadores	Autor (JUAN SEBASTIAN ROMERO ULLOA)
Tester	Jefatura Core Voz CICO
Tracker	Jefatura Core Voz CICO
Coach	Tutor (Víctor Castro)
Big Boss o Manager	Autor (JUAN SEBASTIAN ROMERO ULLOA)

Dentro de la temática de planificación se encuentran las entregas periódicas de la solución, que se realizarán cada 15 días en acuerdo con el *Coach* del proyecto, y un prototipo de la solución cada mes en conjunto con el *Cliente*, de igual forma las reuniones de seguimiento con el *Cliente* se llevarán a cabo en los mismos tiempos de las entregas.

13.33.3 Diseño de la solución según metodología XP

El diseño de la solución está establecido para ser dividido en varias partes:

13.3.1 3.3.1 Diseño del software



[JA21][JSRU22]

Figura 7. Algoritmo básico del software. [Fuente: Autor]

En la figura 7 se encuentra el diseño base del software que hará la toma, análisis y guardado de datos, el algoritmo está diseñado para estar en una interfaz simple, lo que permite a los usuarios la fácil interacción y reducir el número de errores con una única salida y una única entrada de datos.

13.3.1.13.3.1.1 Diseño del algoritmo para realizar llamada desde pc

Se van a probar varios diseños para ejecutar la llamada de forma satisfactoria para que ésta pueda ser almacenada y analizada posteriormente. El primer diseño está planeado para ejecutar la llamada directamente desde el celular, realizando la comunicación mediante la web^{[JA23][JSRU24]} interna de la compañía hacia el servidor en el que se aloje el aplicativo para ser llamado por la interfaz del operario. Luego de realizar la llamada se almacena la grabación dentro del celular y se analiza con una aplicación previamente instalada. El segundo diseño está pensado en la facilidad de toma de datos, pues la documentación sobre algoritmos que realicen análisis de audio en aplicativos es muy limitada, por la misma razón es necesario llevar la grabación fácilmente al pc, pues en este existen más librerías y algoritmos de código abierto que dan facilidad y libertad al código. El tercer diseño está pensado en la resolución de los errores que pueden existir sobre los archivos^{[JA25][JSRU26]} errores de grabación de los audios debido al ruido externo, y en filtrado del mismo de este, por ello se tenía pensado convertir las grabaciones en texto plano para únicamente realizar una comparación de dos frases, sin embargo, existe el problema sobre el idioma que reconoce el convertor de audio a texto^{[JA27][JSRU28]}, pues es limitado a inglés en la mayoría de casos, y en el caso que requiere el cliente se necesita un análisis de audio puro sin traducir a texto.-

Estudiando el caso de los tres diseños, sus ventajas y desventajas, el diseño del primer prototipo es el siguiente para realizar la llamada:

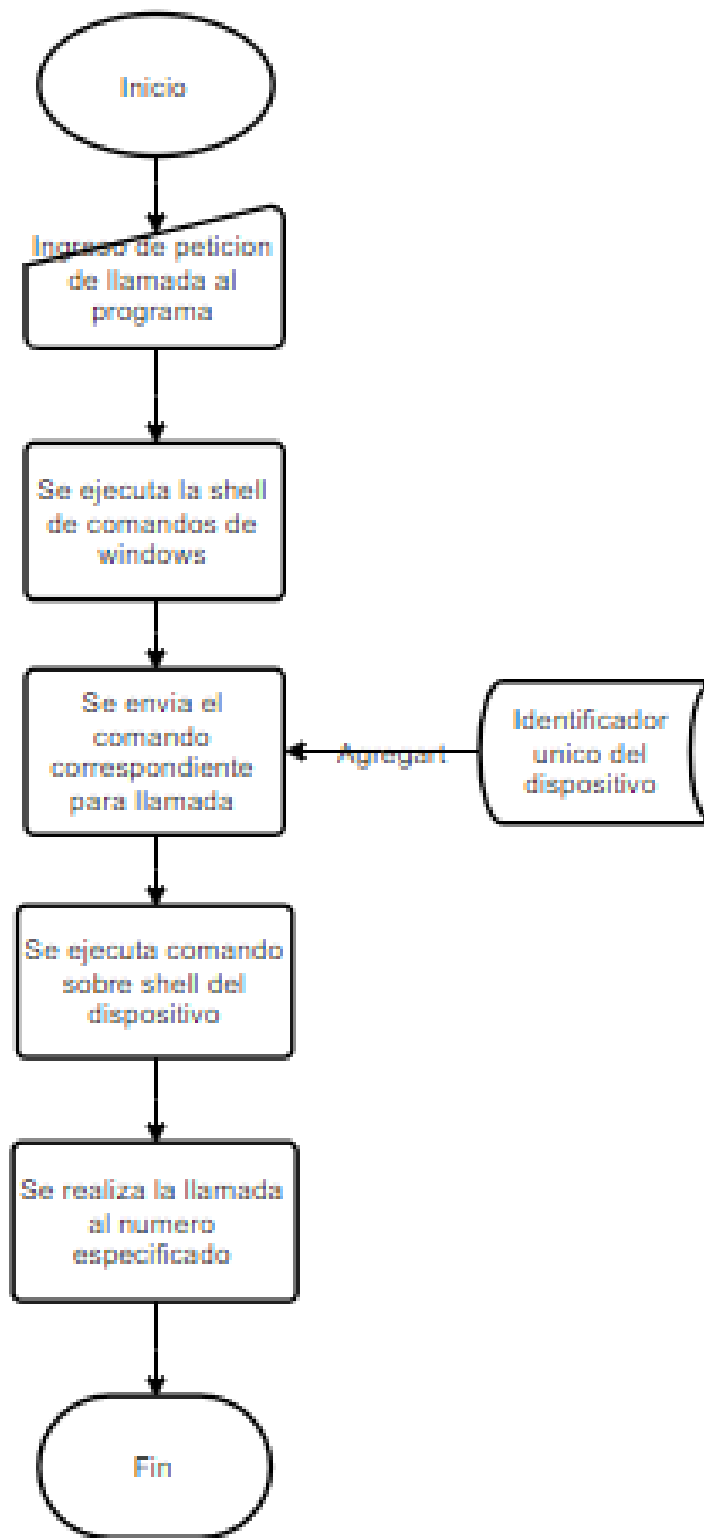


Figura 8. Algoritmo para realizar llamada. [Fuente: Autor]

13.3.1.23.3.1.2 Diseño del algoritmo para realizar grabación de llamada

Para iniciar el proceso de grabación, es necesario identificar parámetros como los bytes de muestra y la frecuencia de muestreo. Luego de identificar los parámetros, se debe abrir la línea del micrófono, pues no se puede iniciar la grabación con la línea de micrófono estando en uso por otro servicio o aplicación del pc. La grabación debe realizarse en formato WAV que como ventaja tiene el peso y la identificación por casi cualquier dispositivo. El diseño básico para la función de la grabación es el siguiente:

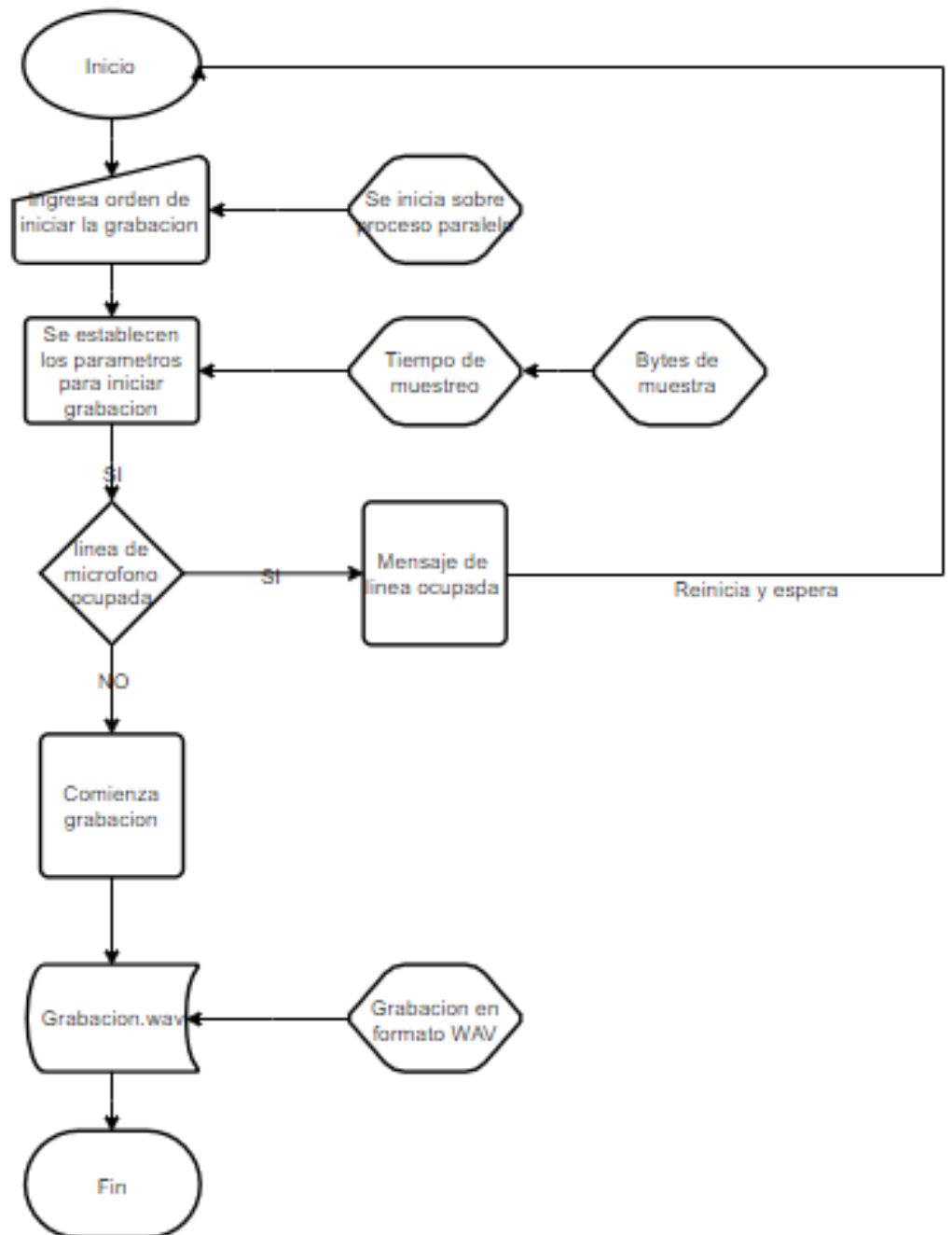


Figura 9. Algoritmo de grabación en PC. [Fuente: Autor]

13.3.1.33.3.1.3 Uso del algoritmo de análisis de audio de Shazam

Según lo visto en la sección de *Marco Teórico*, el análisis de audio puede realizarse por medio de diferentes técnicas, una de esas técnicas es la transformada rápida discreta de Fourier (DFT), analizando los métodos existentes y el estado de arte sobre esta temática, se utilizará para el diseño el algoritmo que utiliza la aplicación *Shazam*, que tiene empaquetada la conversión por DFT en librerías y en código abierto. Para el diseño se tiene pensado el siguiente algoritmo:

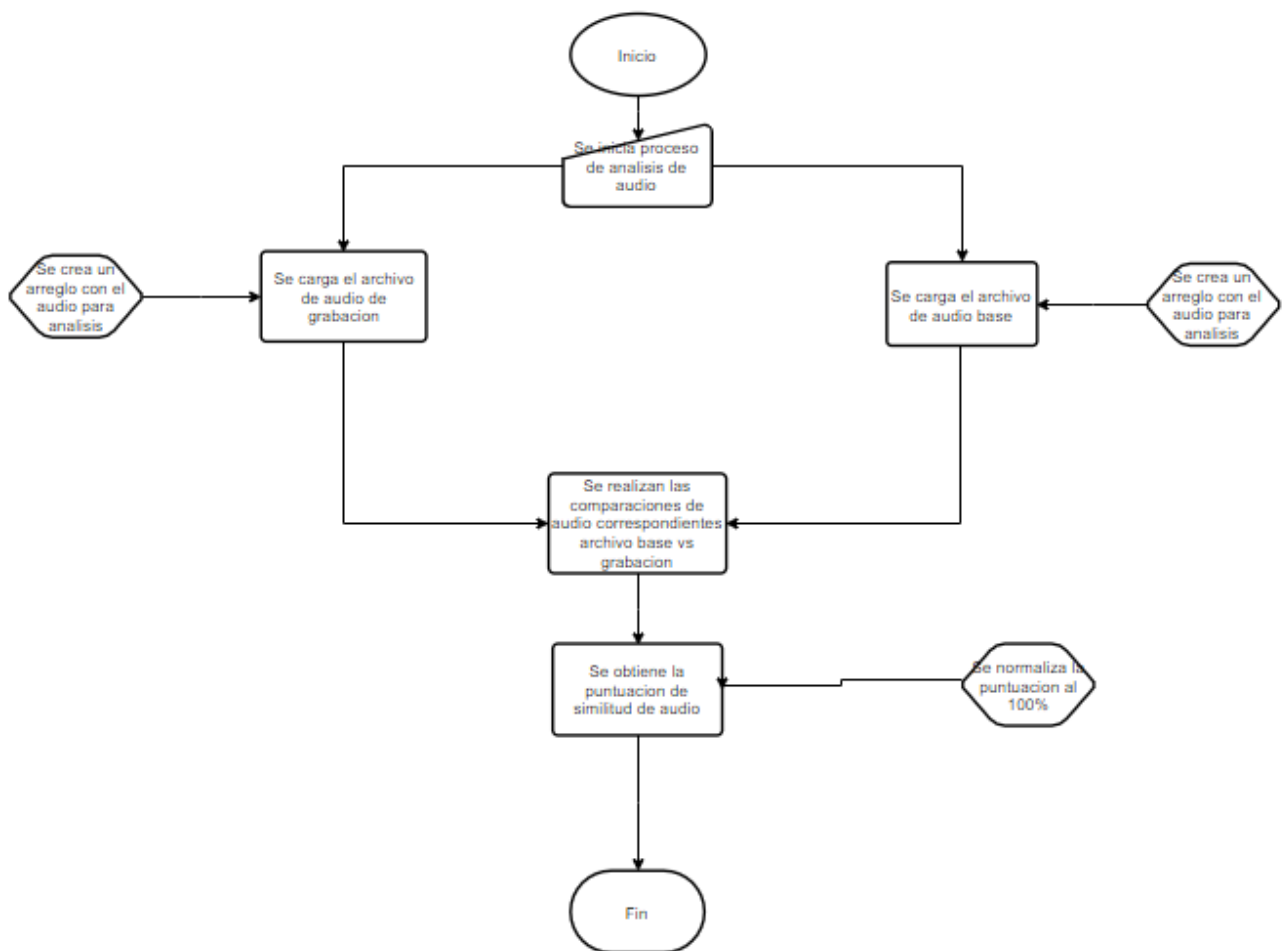


Figura 10. Algoritmo de análisis de Audio. [Fuente: Shazam herramienta de tercero]

13.3.1.4 Perfil de uso del software

El perfil del actor que usará el software es necesario que tenga conocimientos y experiencia en tecnologías 3G y 4G. Además, la persona que utilizara el software debe tener permisos de acceso al MGW para comprobación de los mensajes de anuncio.

13.3.2 Diseño de la base de datos

El diseño de la base de datos se debe realizar de forma adecuada pues las tablas deben tener datos únicos dependiendo de cada MGW. Se piensa almacenar los datos de pruebas de los MGW's individualmente.

13.3.2.1 Diseño de las tablas

Las tablas no tendrán relación, lo que las hace únicas, estarán compuestas de tres columnas básicas:

Tabla 4. Diseño de tablas *[Ninguna tabla tiene clave primaria ni dependencia]*.

Fecha de toma del dato [JA29][JSRU30]	Porcentaje de aprobación	Porcentaje de error

Inicialmente, se ~~tiene pensado que cada~~ asignará una base de datos para que cada MGW tenga su propia tabla, ~~sin embargo,~~ todas tendrán tablas con los mismos parámetros en las columnas. [JA31][JSRU32] El ingreso de datos se realizará por medio de una conexión Java y MySQL. No habrá relación entre las tablas de las bases de datos, pero se podrán hacer consultas para obtener datos individuales según se requiera a lo largo del diseño.

Los campos de las tablas de las bases de datos representan datos individuales por MGW y datos realizados por prueba, cada prueba esta listada por mensaje configurado, y los datos representativos son la aprobación y el porcentaje de error de la prueba:

Tabla 5. Campos de las tablas

Campo	Tipo del campo	Descripción
Fecha	Fecha - Hora	La fecha se realizará colocando un Timestamp del momento que se realice la prueba adjuntando la hora en formato militar o formato de 24h.
Porcentaje Aprobación	Decimal con 2 dígitos de precisión	El porcentaje de aprobación es un campo que ya estará normalizado de 0.00% a 100% y expresa la aprobación que se le da a la evaluación del audio con el algoritmo.
Porcentaje Error	Decimal con 2 decimales	El porcentaje de error del campo que ya estará normalizado desde 0.00% hasta 100% expresará el error que se le da a la evaluación del audio con el algoritmo usado.

Se tiene estimado un total de 10 MGW's, además se dejará establecido el escalamiento a futuro.

~~13.3.2.3~~ 13.3.2.3

datos

Diseño de interfaz para presentar los

Se tiene como diseño utilizar gráficos de barras horizontales, mostrando un porcentaje sobre 100 según sea el caso aplicado. Para los datos de aprobación se utilizará un porcentaje positivo el cual mostrará que tan efectiva fue la prueba sobre ese mensaje en específico, por lo contrario, los datos de ~~error,error~~ serán clasificados con un porcentaje negativo y se mostrarán igualmente al lado de los positivos.

Las bases de datos se realizarán en MySQL el cual es un gestor especializado en bases de datos, que aplica un lenguaje específico. Nos permite interactuar con los datos realizando consultas a dichos datos y realizar conexiones desde casi cualquier dispositivo, por medio de credenciales de acceso.

~~13.3.2.4~~ 13.3.2.4

Diagrama de secuencia

Caso 1: Ejecución de software de reconocimiento de audio

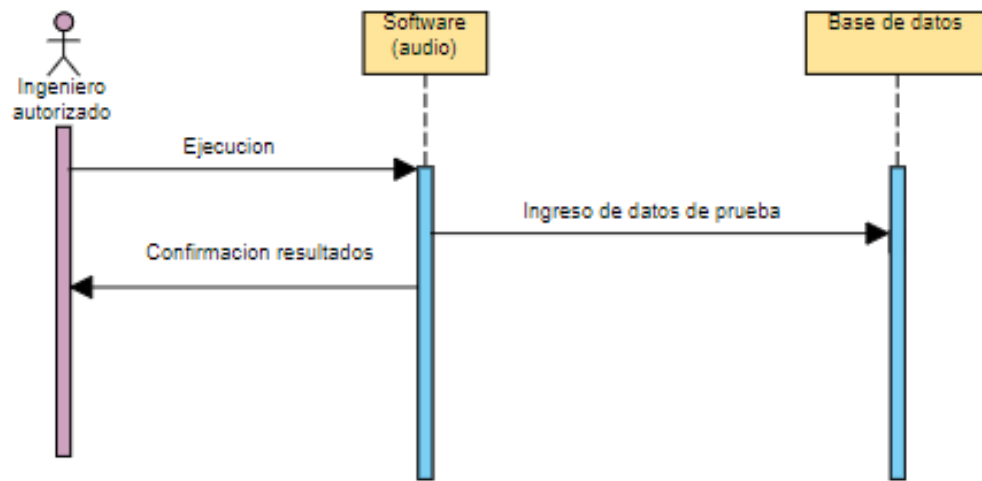


Figura 11. Ejecución de Software. Fuente: [Autor]

Caso 2: Observación de los datos

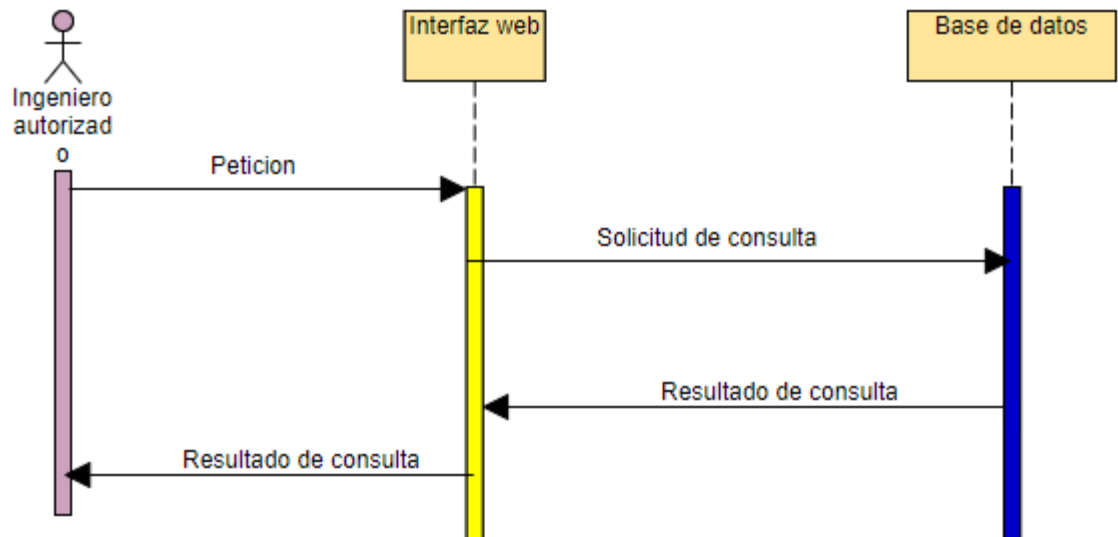


Figura 12. Visualización de datos en interfaz web. Fuente: [Autor]

13.4.13.4 Desarrollo de la solución según metodología XP

13.4.13.4.1 Desarrollo del software

Para el desarrollo del software es necesario saber que el lenguaje de programación que se va a utilizar es JAVA. En base a lo anterior se procederá a utilizar el procedimiento de planeación y diseño que se estableció, pueden surgir algunos cambios debido a cambios del cliente final y/o el tutor a medida que se vaya desarrollando el proyecto.

13.4.1.13.4.1.1 Interfaz visual

Basados en el diseño inicial del programa, la interfaz visual tendrá un aspecto relacionado al siguiente:



Figura 13. Interfaz visual Sugerida. Fuente[Autor]

La interfaz visual consta de un campo principal, el cual es el encargado de capturar el número telefónico, al cual se va a realizar la llamada. Consta igualmente, de un botón que permite realizar la llamada y grabar en un archivo de extensión .WAV de aproximadamente 30s. el botón cambia de estado 'activo' a 'inactivo' cuando se da clic sobre él.

Existe una particularidad cuando se trabaja con modelados en Java y es que se manejan procesos en serie, es decir para que se ejecute uno debe acabarse el otro

anteriormente, en este caso el modelado de los botones y el JFrame se realizan con procesos en serie, al ejecutar funciones que consuman recursos del aplicativo, se congelan los procesos como el modelado del botón y el frame, lo que hace imposible manipular tanto la interfaz como el software. Para solucionar el problema de procesado en serie, se utilizan funciones en paralelo, por eso se adiciona como proceso en paralelo la apertura de la línea del micrófono del PC y la grabación de la misma por 30s.



Figura 14. JFrame y botones activos. Fuente: [Autor]

Mientras los botones y el JFrame se encuentren realizando la función en paralelo se pondrán en estado 'inactivo' para que el usuario no pueda seleccionar otras opciones y colisionar las ejecuciones:



Figura 15. JFrame y botones inactivos. Fuente: [Autor]

En las salidas, se diseñó el software para que se realizara de manera que el usuario lograra observar la actividad que se estuviera ejecutando en la consola de actividades. Para lograrlo existirán varias salidas de pantalla las cuáles serán las siguientes sugeridas:

Tabla 6. Sugerencias de salida del programa por consola. Fuente: [Autor]

Salida	Descripción
Shell ADB	Salida del uso de Shell de Android
Status <u>Estatus</u> de Grabación	Muestra el estado actual de la grabación de la línea de micrófono del PC.
Shell Windows	Finaliza el proceso de Shell ADB que pueda quedar en ejecución.
Salida de ejecuciones: Aprobación	Muestra en salida de consola de Java el porcentaje de aprobación que muestre la ejecución de la prueba de similitud de huella acústica.
Salida de ejecuciones: error	Muestra en salida de consola de Java el porcentaje de error que muestra la ejecución de la prueba de similitud de huella acústica.

En una salida por consola sugerida se observaría de la siguiente manera:

```

Starting: Intent { act=android.intent.action.CALL dat=tel:xxxxxx }

Grabando...
1
Correcto: se termino el proceso "adb.exe" con PID 3124.
Puntaje de similitud :
76,52 % de igualdad
-23,48 % de margen de error

```

Figura 16. Salida sugerida de Consola. Fuente: [Autor]

La ejecución de la salida de los datos hacia la base de datos se realiza por medio de una clase que conecta Java con el gestor administrador MySQL. La salida sugerida para la base de datos confirma al usuario si la conexión a la misma ha sido exitosa o no, de la siguiente manera sugerida:



Figura 17. Confirmación de ingreso de datos sugerida. Fuente: [Autor]

La interfaz para hacer la lectura y/o consulta de las bases de datos se sugiere como la siguiente:

Pruebas realizadas sobre mensaje de anuncio 80			
Fecha	Porcentaje de Aprobación (%)	error (%)	
1 2019-03-28 09:15:24.744	58	-42	
2 2019-03-28 09:16:28.953	70.75	-29.25	
3 2019-03-28 09:22:16.18	66.12	-33.88	
4 2019-03-28 11:53:07.066	76.52	-23.48	

Figura 18. Interfaz sugerida para lectura y/o consulta de base de datos. Fuente: [Autor]

La estructura de la base de datos puede cambiar, sin embargo, se intentará mantener el diseño sugerido. Se establecerá un método de consulta por medio de tablas haciendo uso de las librerías Google Chart Tables las cuales nos permiten agregar ciertas características a nuestras tablas, se realiza una conectividad por medio de PHP en conjunto con Google Chart y utilizando una funcionalidad, se realizaría la muestra de los datos por medio de barras con porcentaje azul y rojo correspondientes al porcentaje de aprobación y porcentaje de error respectivamente.

13.4.23.4.2 **Herramientas de Terceros**

Con el fin de mejorar los procesos del software, se decidió implementar algunos procesos y/o software ya establecidos. Estos procesos permiten al proyecto ir más rápido y más eficaz con respecto a no tener errores, para ello se utilizaron herramientas como las siguientes:

13.4.2.1.13.4.2.1.1 **ADB Shell Android** [JA33][USRU34]

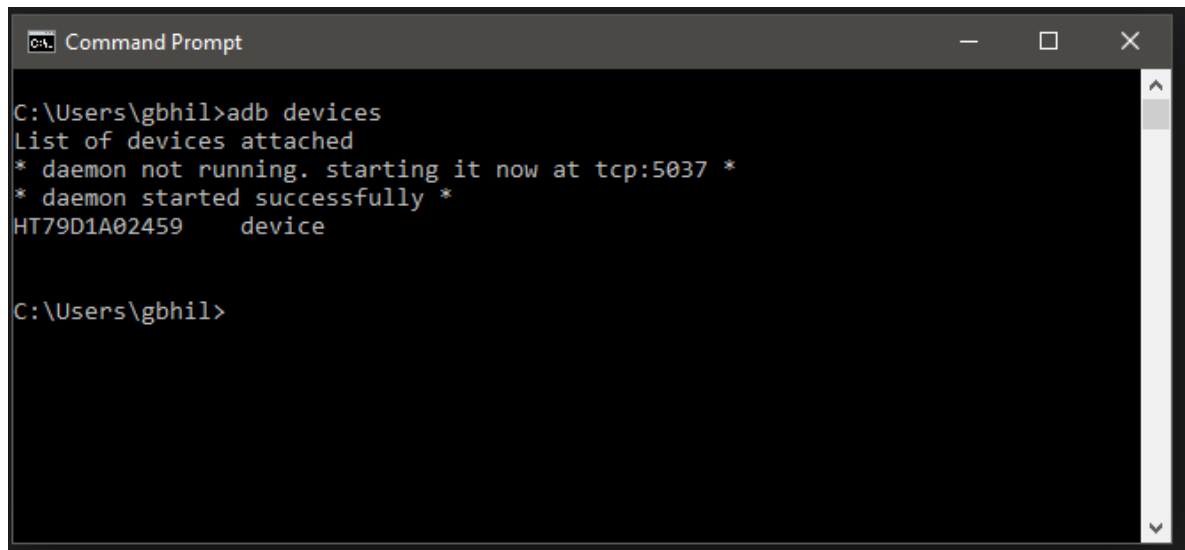
Como ya es sabido Android es un sistema operativo que se encuentra dentro de los dispositivos móviles celulares. Sea cual sea su versión o modificación, todos manejan una Shell base de Android que permite a los desarrolladores crear nuevos aplicativos o probar funcionalidades del teléfono sin necesidad de crear aplicaciones utilizando por ejemplo la herramienta Android Studio. En el caso del proyecto, se va a utilizar el comando para utilizar el dial de marcación y la realización de una llamada así como finalizarla cuando ésta acabe después de cierto tiempo transcurrido.

Para lograr la comunicación con el dispositivo es necesario que el usuario que necesite conectarse decida si lo hará por medio de USB o WI-FI de manera que le permita facilidad para transmitir los comandos.

Luego de generar la conectividad inicial con el dispositivo es necesario que el usuario del dispositivo lo tenga en modalidad de desarrollador, la cual le permite

acceder a las funcionalidades completas del teléfono, para realizar este procedimiento es necesario acceder a Configuraciones del teléfono>>Versión del teléfono>> y allí dar siete toques a la opción >>Versión del sistema operativo. Cabe aclarar que el procedimiento varía según el modelo, marca y versión del sistema operativo Android del teléfono.

Se utilizó esta herramienta en el proyecto y no la que viene por defecto con Android Studio SDK, porque es mucho más liviana y en comparación a rendimiento es mucho más eficaz, únicamente posee una combinación entre fastboot y adb Android Shell, servicios fáciles de utilizar dentro de prácticamente cualquier Shell, hace uso de un paquete con los comandos más necesarios para el dispositivo.



```
C:\Users\gbhil>adb devices
List of devices attached
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *
HT79D1A02459    device

C:\Users\gbhil>
```

Figura 19. Shell ADB Android. Fuente: [Autor]

13.4.2.1.1.13.4.2.1.1.1 Uso de ADB Shell Android en el proyecto

Para uso de la Shell en el proyecto el usuario deberá ingresar el comando “adb –s (ID del celular) shell am start –a android.intent.action.CALL –d tel: (número telefónico)” que le permitirá realizar una llamada al número cualquiera que se desee. De la misma forma, para finalizar la llamada es necesario el comando “adb shell input keyevent KEYCODE_ENDCALL”, para aplicar estos comandos desde Java es necesario utilizar una sentencia “Shell_exec” que permitirá ejecutar comandos de

consola de Windows y correrlos sobre un proceso en paralelo para no afectar el trazado de imagen sobre la interfaz visual de Java. De igual forma es necesario codificar los comandos en formato UTF (Unicode Transformation Format) pues el intérprete que en este caso es la consola de ejecución de Android, lo entienda de forma correcta, incluyendo la mayoría caracteres especiales.

~~13.4.2.1~~23.4.2.1.2 **Librería MusicG (Google)**

La librería MusicG es una librería de código abierto, creada por Google con el propósito de analizar señales de audio. La librería está dispuesta para instalación en lenguaje Java o su implementación en Android. La librería está basada en la aplicación Shazam, y nos permite analizar la similitud entre una base de datos de archivos de audio con respecto a otro. Esta similitud es entregada y normalizada como un puntaje de 0.0 a 100, lo que será utilizado para el aplicativo. Debido a las pocas fuentes de códigos abiertos acerca de un algoritmo que analice audio se decidió utilizar MusicG como única librería en el proyecto.

Google Code Archive

Search this site

Projects

Search

About

Project

Source

Issues

Wikis

Downloads

musicg

Lightweight Java API for audio analysing, Android compatible

About

musicg is a lightweight audio analysis library, written in Java, with the purpose of extracting both high level and low level audio features.

This API allows developers to extract audio features and operate audio data like reading, cutting and trimming easily from an inputstream. It also provides tools for digital signal processing, renders the waveform or spectrogram for research and development purpose.

The API is Android compatible.

Application Demo

WhistleAPI Demo for Android in Downloads here

Surprise! on Android Market: Listen for a whistle and responses with user defined image presentation and sound playing.

Project Information

The project was created on Apr 1, 2012.

- License: [Apache License 2.0](#)
- 114 stars
- git-based source control

Labels:

java

music

android

audio

waveform

spectrogram

fft

whistle

soundapi

musicg

musicdiscovery

Figura 20. Página de alojamiento de la librería MusicG. Fuente:[Google]

13.4.2.1.3.4.2.1.3 Uso de la Librería MusicG (Google)

La librería se descarga desde la misma página web mostrada en la figura 20, luego se carga e importa hacia el entorno de desarrollo como se observa en la figura 21:

Figura 21 Importación de librería MusicG. Fuente [Autor]

```
import com.musicg.fingerprint.FingerprintManager;
import com.musicg.fingerprint.FingerprintSimilarity;
import com.musicg.fingerprint.FingerprintSimilarityComputer;
import com.musicg.wave.Wave;
import java.awt.event.ActionEvent;
```

Figura 22. Importación de librería MusicG dentro del código. Fuente: [Autor]

62

Google Charts son una variedad de librerías creadas para su uso en páginas web. Les permiten a los usuarios implementar código abierto de un gran catálogo de esquemas web y organización multimedia, desde gráficos de barras, mapas, histogramas, hasta modos de organización de tablas. Cada uno de estos gráficos posee facilidad de modificaciones según las necesidades de los usuarios. Su facilidad de implementación y versatilidad para el uso dentro del proyecto permiten ejecución y llamado de scripts de forma remota para no tener la necesidad de alojamiento dentro del directorio.

Se utilizó esta alternativa dentro del proyecto por su facilidad de implementación, únicamente es necesario importar un archivo JavaScript que se puede importar tanto de forma local como remotamente desde la página de Google Charts, con diseños predeterminados y ajustables a cualquier necesidad que dentro del caso del proyecto se importó la configuración de tablas. Su uso puede observarse en la figura 18 con gráficos de barras que representan una escala de 0% a 100%.

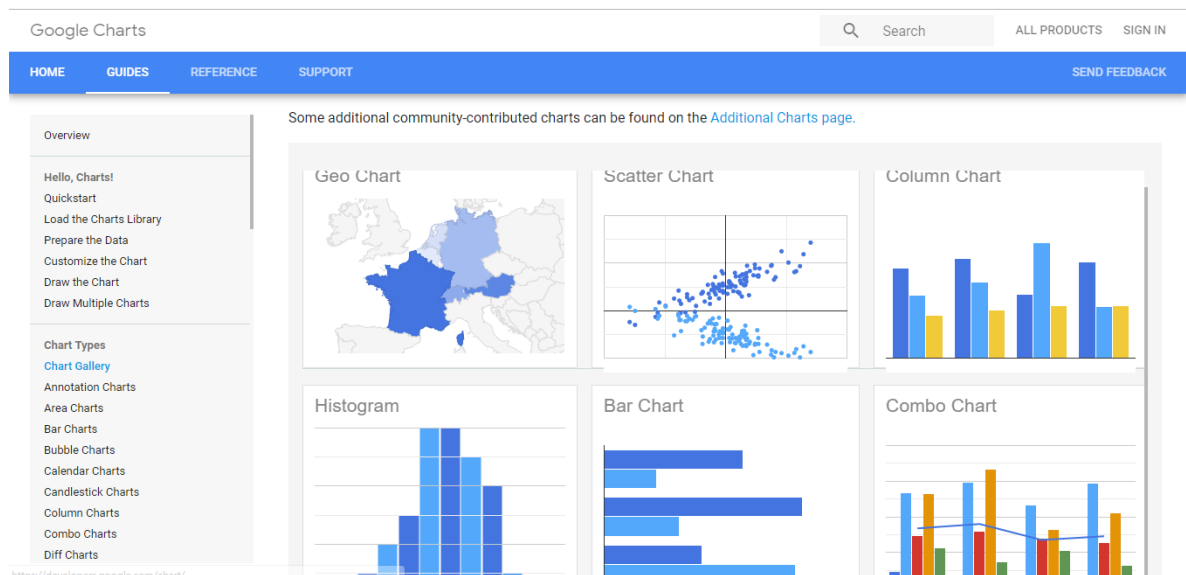


Figura 23. Galería de Gráficos y esquemas. Fuente: [16]

Para el desarrollo de la base de datos con el algoritmo de diseño propuesto es necesario utilizar PHP para describir el código adecuadamente, y lograr conectar la base de datos de forma efectiva y sencilla, al igual que la utilización de la herramienta *Google Chart Tables*. La base de datos se realizará con ayuda del gestor MySQL PHPmyAdmin el cual cuenta con una interfaz visual que permitirá acceder mejor a las herramientas del sitio. Por otro lado, las pruebas inicialmente se realizarán con cuatro tablas principales que almacenarán las pruebas de cuatro centrales y nos permitirán visualizarlas respectivamente.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
pruebasbarranquilla	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	latin1_swedish_ci	16 KB	-
pruebasbogota	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16 KB	-
pruebaspereira	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	16 KB	-
pruebasiberia	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16 KB	-
4 tablas	Número de filas	6	InnoDB	latin1_swedish_ci	64 KB	0 B

Figura 24. Tablas de la base de datos. Fuente: [Autor]

La base de datos está constituida por cuatro tablas principales que representan cada una una central dentro de la red. Cada central almacenará los datos de las pruebas realizados, su estructura será la siguiente:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	Mensaje_aproximado	int(11)			No	Ninguna			Cambiar Eliminar Primaria Único Más
2	Porcentaje	decimal(10,2)			No	Ninguna			Cambiar Eliminar Primaria Único Más
3	error	decimal(10,2)			Si	NULL			Cambiar Eliminar Primaria Único Más
4	Fecha	text	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Primaria Único Más

Figura 25. Estructura de las tablas de la base de datos. Fuente: [Autor]

Como se observa en la figura 253, [JA35] la tabla estará estructurada por cuatro campos o columnas, el primer campo “Mensaje_Aproximado” representa el mensaje de anuncio más similar al cotejamiento realizado por el software mostrando así un valor entero correspondiente al audio. El segundo valor “Porcentaje” muestra el porcentaje de similitud normalizado entregado por el software, este valor esta dado

en decimales con una precisión de 2. El tercer campo, “error” es un porcentaje igualmente entregado por el software que es el error de hacer la prueba de similitud, pues la similitud se realiza 10 veces para mayor precisión, el valor error esta entregado igualmente en decimales y con precisión de 2 al igual que el campo porcentaje esta normalizado. El campo “Fecha” representa la hora y fecha en que se realizó la prueba, este valor es texto entregado por el software con el formato “AAAA-MM-DD HH: Min: SS”. Las tablas no poseen datos primarios o clave para consultar. Los datos se almacenarán de la siguiente forma:

✓ Mostrando filas 0 - 4 (total de 5, La consulta tardó 0.0019 segundos.)

`SELECT * FROM `pruebasbarranquilla``

☐ Perfilando [Ec

☐ Mostrar todo | Número de filas: 25 Filtrar filas:

+ Opciones

Mensaje_aproximado	Porcentaje	error	Fecha
72	66.46	-33.54	2019-04-23 10:17:37.899
72	61.46	-38.54	2019-04-23 10:21:34.212
72	85.42	-14.58	2019-04-23 10:23:59.989
72	53.33	-46.67	2019-04-23 10:29:40.552
72	44.17	-55.83	2019-04-23 10:34:32.421

Figura 26. Ejemplo de datos consultados de la tabla "Barranquilla". Fuente: [Autor]

13.4.3.13.4.3.1 Desarrollo de la interfaz de consulta y páginas web

La interfaz de consulta está dada por una variedad de páginas web todas conectadas entre sí, lo que le permite al usuario una mejor experiencia. La página principal tiene un botón en específico que le permite al usuario descargar el software para instalarlo en su PC, además contiene enlaces hacia las demás páginas que

permiten cambiar los mensajes de anuncio directamente hacia las centrales para hacer las pruebas correspondientes. Para la representación de cada central, hay cuatro páginas web, cada una igualmente con enlaces hacia las demás y con botones que permiten visualizar la base de datos final similar a la figura 18. Un ejemplo de las páginas web de consulta es el siguiente:



Figura 27. Página web de consulta, central de Entrerrios. Fuente: [Autor]

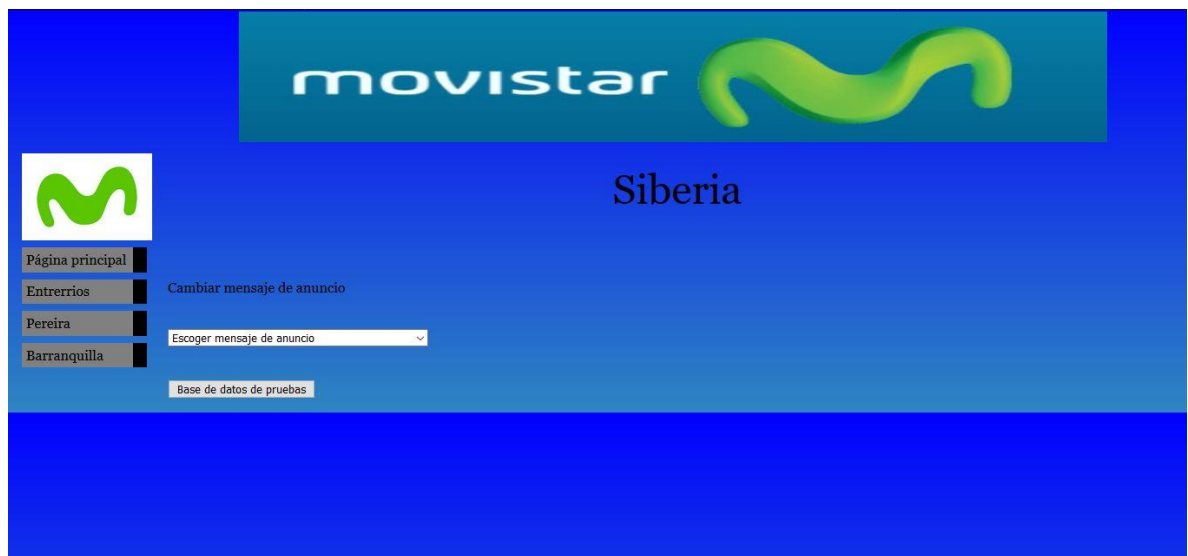


Figura 28. Página web de consulta, central de Siberia. Fuente: [Autor]



Figura 29. Página web de consulta, central de Pereira. Fuente: [Autor]



Figura 30. Página web de consulta, central de Barranquilla. Fuente: [Autor]

En resumen, cada una de las páginas web tiene la misma plantilla y cada una conecta a una diferente base de datos, el proyecto comienza únicamente con 4 centrales, en las cuales se podrá hacer el cambio de mensaje de anuncio con perfil de administrador correspondientemente. El procedimiento no varía del diseño requerido inicialmente, se requiere hacer un cambio de anuncio de mensajería de anuncio, acorde a ello se realiza la prueba, que consiste en realizar la llamada a un número específico con una duración aproximada de 30s, al mismo tiempo que se

realiza la llamada, se ejecuta la grabación de ~~la misma~~esta y su análisis posterior los 30s. Luego del análisis de la huella acústica de los dos archivos se analiza su similitud y se guarda sus datos correspondientemente en una base de datos que puede ser visualizada por medio de una interfaz web correspondiente a dicha central. El software es diseñado para cumplir el propósito de ayudar y hacer más efectivo las ventanas de mantenimiento, pues sucedía que luego de una ventana de mantenimiento sobre una central en ocasiones presentaba el fallo de uno de estos mensajes colocados, o incluso después de la migración o actualización de ~~los mismo~~estos, se lograban ver afectados.

15.4. CAPÍTULO IV

15.14.1 Conclusiones

El proyecto permitió realizar una comprobación sobre los mensajes de anuncio alojados en las MSS lo que a su vez logró hacer una ardua verificación sobre el estado en el que se encontraban los mismos. El software además permitió mediante ~~20 un número considerable de llamadas~~ [JA36] evaluar la calidad de las grabaciones que se encontraban allí alojadas entregándoles a las mismas una puntuación con escala de 0% a 100% siendo 100% un porcentaje de similitud y calidad óptima.

Se desarrolló una metodología para encontrar la similitud entre la grabación realizada y el audio original, para lograrlo se desarrolló un algoritmo que hace uso de varias herramientas de terceros que son de uso libre como el uso de “*ADB Shell*”, “*Fastboot*” y “*Google Charts*” a su vez, permiten enfocar los procesos que se necesitaban optimizar en tiempo y recursos. Es necesario aclarar que el algoritmo de identificación de audio de Google MusicG es una librería de uso libre y fue hecha por la corporación misma, su único uso fue para la identificación de la huella acústica de los audios.

Se hizo una interfaz que permitiera realizar llamadas y posteriormente colgarlas desde el PC, esto se puede expandir a procesos como automatización de llamadas desde el dispositivo móvil y análisis de patrones de voz en las llamadas.

Mediante la visualización de las bases de datos con las tablas de Google se puede observar las pruebas que se han realizado sobre las centrales, de igual forma se puede observar el porcentaje de éxito ~~del mismo de este~~, esto permitió al proyecto una gestión para observar la evolución a partir de la corrección de errores de audio en el futuro.

Se realizó una herramienta de monitorización para intercambiar los mensajes de anuncio sobre las centrales, de igual forma un sistema de gestión para el cambio rápido de los mismos, permitiendo al usuario agilizar procesos de conectividad e ingreso de comandos.

El Proyecto realizado puede expandirse en cualquier horizonte y el análisis de audio en una grabación puede ser clave para entornos como IoT (Internet of things), llamadas remotas y procesos que se necesiten realizar de forma remota utilizando el teléfono móvil, incluso llamadas automáticas programadas, reemplazando los llamados softphone.

Se evidencio que el proyecto fue un éxito, pues cumplió con los estándares y objetivos de diseño propuestos, así como las proyecciones y modificaciones que se realizaban a lo largo del mismo. Su funcionamiento es correcto y todos los datos quedan almacenados en el servidor para su futura monitorización.

BIBLIOGRAFÍA

3GPP ETSI TS 123 002 V15.0.0. (Julio de 2018). Network architecture. (3. T. 15, Ed.)

3GPP ETSI TS 123 228 V15.2.0. (Julio de 2018). IP Multimedia Subsystem (IMS); Stage 2. (3. T. 15, Ed.)

Association GSMA VoLTE Architecture, G. (18 de December de 2014). *VoLTE GSMA Service Description and Implementation Guidelines*.

Diego Calvo. (7 de abril de 2018). *Metodología XP Programación Extrema (Metodología ágil)*.
Obtenido de <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

Google Inc. (23 de Febrero de 2017). *Google Charts*. Obtenido de Google Charts Gallery: <https://developers.google.com/chart/interactive/docs/gallery>

Inc., Google. (22 de octubre de 2008). *Google Play Store*. Obtenido de <https://play.google.com/>

J.M. Hernández. (s.f.). *Estructura y características de la señal acústica. Estudio y caracterización*. Obtenido de <https://jmhweb.wordpress.com/docencia/master-en-biologia-de-la-conservacion-ucm/estructura-y-caracteristicas-de-la-senal-acustica-estudio-y-caracterizacion/>

Johnston, J. D. (1988). Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6.

Metodología XP Programación Extrema (Metodología ágil). (07 de Abril de 2018). Obtenido de <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

Polo, H. C. (2009). *Redes Móviles Celulares*. Universidad Santo Tomás.

Ribeiro-Neto R. Baeza-Yates and B. Modern Information Retrieval. (1999). Modern Information Retrieval.

UIT-T. SECTOR DE NORMALIZACIÓN DE LAS TELECOMUNICACIONES DE LA UIT. (Noviembre de 2006). Subsistema multimedia IP (IMS) IPCablecom2: Presentación general de las Recomendaciones complementarias. *SERIE J: REDES DE CABLE Y TRANSMISIÓN DE PROGRAMAS RADIOFÓNICOS Y TELEVISIVOS, Y DE OTRAS SEÑALES MULTIMEDIA* .

www.3gpp.org/. (2018). Obtenido de www.3gpp.org/about-3gpp/about-3gpp

www.telefonica.co. (13 de Septiembre de 2018). Obtenido de <http://www.telefonica.co/acerca-de-telefonica-colombia>

¹ (www.telefonica.co, 2018)

² (3GPP ETSI TS 123 002 V15.0.0, 2018)

³ (3GPP ETSI TS 123 228 V15.2.0, 2018)

⁴ (Association GSMA VoLTE Architecture, 2014)

⁵ (www.telefonica.co, 2018)

⁶ (UIT-T. SECTOR DE NORMALIZACIÓN DE LAS TELECOMUNICACIONES DE LA UIT, 2006)

⁷ (UIT-T. SECTOR DE NORMALIZACIÓN DE LAS TELECOMUNICACIONES DE LA UIT, 2006)

⁸ (Ribeiro-Neto R. Baeza-Yates and B. Modern Information Retrieval, 1999)

⁹ (J.M. Hernández, s.f.)

¹⁰ (Inc., Google, 2008)

¹¹ (Diego Calvo, 2018)

¹² (Metodología XP Programación Extrema (Metodología ágil), 2018)

¹³ (Metodología XP Programación Extrema (Metodología ágil), 2018)

¹⁴ (Metodología XP Programación Extrema (Metodología ágil), 2018)

¹⁵ (Diego Calvo, 2018)

¹⁶ (Google Inc., 2017)