



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN INFORMÁTICA

Desarrollo de un prototipo de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista.

Año lectivo 2017-2018

TRABAJO DE TITULACIÓN

AUTOR: Landázuri Peña Paul Andrés

DIRECTOR: Abad Espinoza Marco Patricio, Ing.

CENTRO UNIVERSITARIO REGIONAL QUITO

2018



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2018

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniero.

Marco Patricio Abad Espinoza.

DIRECTOR DEL TRABAJO DE TITULACIÓN

De mi consideración:

El presente trabajo de titulación: desarrollo de un prototipo de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista realizado por Landázuri Peña Paúl Andrés, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, abril del 2018

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Landázuri Peña Paúl Andrés, declaro ser autor del presente trabajo de titulación: Desarrollo de un prototipo de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista, de la Titulación de Ingeniería Informática siendo Marco Patricio Abad Espinoza director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición de Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad Intelectual de Investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o Institucional (operativo) de la Universidad.”

f.....
Autor: Landázuri Peña Paúl Andrés
Cédula:1722957949

DEDICATORIA

Dedico este trabajo de titulación, a mis padres quienes me han impulsado a seguir estudiando durante años, quienes me recuerdan siempre, que una buena educación es el pilar del éxito y que siempre que me esfuerce, podré alcanzar las metas que me proponga.

Recuerdo como mi padre siempre nos dice a mis hermanos y a mí, que un trabajo bien realizado dignifica a la persona, y uno se siente orgulloso de lo que hace; cuando, con su trabajo ayuda a que la sociedad y el país se beneficie; olvidándose de la mediocridad y dando todo de sí en cada actividad que uno realice, siempre con una actitud de amor a la patria para mejorarla con cada cosa que uno haga, aunque sea algo pequeño, pero que se lo haga bien; por esto dedico esta tesis a mis padres Otto Landázuri y Janeth Peña y sin olvidar a mis hermanos Santiago, Ricardo, Diego, Nataly quienes me han ayudado cuando los necesitaba.

Y dando una especial dedicación a mi novia Silvia quien me ha apoyado a seguir mis estudios y de quien estoy agradecido por su ayuda incondicional en cada semestre que pase en la Universidad Técnica Particular de Loja.

AGRADECIMIENTO

Agradezco primeramente a mi familia, que me ha ayudado a seguir con mis estudios y me ha apoyado en la vida universitaria; además, agradezco a mi novia quien me ha dado ánimos cuando más lo he necesitado, para conseguir terminar los estudios universitarios.

Agradezco a la universidad y a sus profesores que constantemente nos inculcan sus conocimientos en las guías didácticas preparadas, para que cada semestre adquiramos mis compañeros y yo, el conocimiento necesario para ser buenos profesionales; y, además mencionar que gracias al modo de estudio a distancia con el que cuenta la universidad, pude inculcar hábitos de lectura para que constantemente siga auto educándome y más aún cuando mi carrera de Ingeniería en Informática es muy cambiante y es necesario, estar constantemente actualizado.

Al Dr. Fausto Coello y la Dra. Gioconda Baquero quienes me ayudaron con las pruebas de audiometría.

A mi tutor Ing. Marco Abad quien me ayudo en el desarrollo de mi trabajo de titulación.

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xii
ÍNDICE DE DIAGRAMAS	xiii
ÍNDICE DE ANEXOS	xiv
RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN	3
JUSTIFICACIÓN	5
OBJETIVOS	6
CAPITULO I. MARCO TEÓRICO	7
1.1. Arquitectura de software	8
1.1.1. Arquitecturas para aplicaciones móviles	8
1.1.2. Beneficios de contar con una arquitectura adecuada.	9
1.2. Arquitectura de aplicaciones móviles	9
1.2.1. Aplicaciones móviles auto contenidas.	9
1.2.2. Aplicaciones con conexión a internet	10
1.2.3. Patrones de arquitectura software para desarrollar aplicaciones.	11
1.2.4. Modelos de arquitecturas	14

1.2.5.	Estilos arquitectonicos.....	17
1.3.	Patrones de diseño.....	19
1.3.1.	Componentes de un patrón de diseño.....	19
1.3.2.	Patrones del desarrollo del software para aplicaciones.....	20
1.3.2.1.	Patrones creacionales.....	20
1.3.2.2.	Patrones estructurales.....	21
1.3.2.3.	Patrones de comportamiento.....	21
1.4.	Metodologías para aplicaciones móviles.....	22
1.4.1.	Definición de metodología de desarrollo.....	22
1.4.2.	Tipos de metodologías de desarrollo de aplicaciones móviles.....	22
1.4.2.1.	Metodología XP.....	23
1.4.2.2.	Metodología Scrum.....	26
1.4.3.	Análisis comparativo.....	31
1.5.	Información general de audiometría.....	33
1.5.1.	Generalidades de la audición humana.....	33
1.5.2.	Percepción del sonido.....	34
1.5.3.	Anatomía y fisiología del oído humano.....	35
1.5.3.1.	Oído externo.....	36
1.5.3.2.	Oído medio.....	37
1.5.3.3.	Oído interno.....	38
1.5.4.	Causas más comunes que provocan la pérdida auditiva.....	39
1.5.5.	Definición de audiometría.....	40
1.5.6.	Características de la audiometría.....	41
1.5.7.	Representación e interpretación del audiograma.....	41
1.5.8.	Transductores utilizados en audiometrías vía aérea.....	43
1.5.9.	Decibeles NPS vs Decibeles HL.....	44
1.5.10.	Pruebas de audición en dispositivos móviles.....	45
1.5.10.1.	Uso de dispositivos móviles en audiometría.....	45
1.5.10.2.	Evaluación del umbral auditivo de una persona.....	46

1.5.10.3. Necesidad de calibración de los dispositivos móviles para audiometría. .	46
CAPITULO II.ESTUDIO DE NECESIDADES	47
2.1. Requerimientos del sistema.	48
2.1.1. Médico.	48
2.1.2. Usuario.....	56
2.1.3. Dispositivo móvil (Smartphone o Tablet).	57
CAPITULO III.DISEÑO.....	59
3.1. Arquitectura.....	60
3.2. Datos.	61
3.3. Interfaces.	63
3.4. Implantación.....	63
3.5. Fase de diseño.	66
3.5.1. Vista lógica.	67
3.5.2. Vista de desarrollo.....	69
3.5.3. Vista de procesos.....	69
3.5.4. Diagrama de navegación.	70
3.5.5. Diseño de los bocetos de las pantallas del sistema (prototipo).	70
3.5.5.1. Boceto de la pantalla de IntroAPP.....	70
3.5.5.2. Boceto de la pantalla del menú.....	71
3.5.5.3. Boceto de la pantalla de inicio de prueba de audición.....	72
3.5.5.2. Boceto de la pantalla de información.....	75
3.5.5.3. Boceto de la pantalla de ayuda.....	76
CAPITULO IV.CONSTRUCCIÓN	77
4.1. Librerías.	78
4.1.1. Uso de la librería Java y XML.	78
4.1.2. Uso de la librería GraphView.	78
4.1.3. Uso de las librerías de la plataforma Matlab.	79
4.2. Desarrollo.....	80
4.2.1. Fase de planificación.....	80

4.2.2.	Funcionalidades de la aplicación.	81
4.2.3.	Atributos de calidad.	83
4.2.4.	Restricciones.	83
4.3.	Fase de codificación.	84
4.3.1.	Requerimientos de desarrollo.	84
4.3.2.	Desarrollo del módulo de prueba audición.	85
4.3.2.1.	Descripción la clase Java graficar.	85
4.3.2.2.	Descripción de la clase Java prueba auditiva.	85
4.3.2.3.	Descripción de la clase Java configuración.....	85
4.3.2.4.	Archivo XML Activity_Menu.xml (menú principal).	85
4.4.	Fase de pruebas.	86
4.4.1.	Elaboración y ejecución del plan de pruebas.	87
4.4.2.	Análisis de resultados.....	90
4.5.	Documentación.	91
CAPITULO V.VALIDACIÓN		92
5.1.	Pruebas realizadas.	93
CONCLUSIONES		100
RECOMENDACIONES		101
BIBLIOGRAFIA.....		102
ANEXOS		106

ÍNDICE DE FIGURAS

Figura 1. Modelo vista controlador	12
Figura 2. Modelo vista controlador	13
Figura 3. Modelo vista vista modeló.....	14
Figura 4. Modelo de “4+1” vistas.....	15
Figura 5. Modelo cliente-servidor	18
Figura 6. Flujo de características de la programación extrema.	24
Figura 7. Ciclo de desarrollo ágil Scrum.	29
Figura 8. Ciclo principal de Scrum	29
Figura 9. Ciclo de desarrollo Mobile-D.....	30
Figura 10. Diagrama de uso de la metodología XP	33
Figura 11. Curvas de igual sonoridad obtenidas por Fletcher y Munson.	35
Figura 12. Anatomía y composición del oído humano.	36
Figura 13. Estructura básica del oído externo.	37
Figura 14. Partes principales del oído medio.	38
Figura 15. Composición del oído interno.	39
Figura 16. Señales que se usan para diagramas audiométrico	42
Figura 17. Diagrama audiométrico con señales	42
Figura 18. Audiograma de niveles y audiograma de banana auditiva	43
Figura 19. Especificaciones técnicas de auriculares intra-aurales	44
Figura 20. Equivalencia entre dB NPS vs dB HL.....	45
Figura 22. Sistemas operativos más usados en el mundo.....	60
Figura 23. Esquema de aplicaciones autocontenidas	61
Figura 24. StarUML Modelo 4+1 vistas.....	62
Figura 25. Plataforma de desarrollo.....	63
Figura 26. Ventana de creación de dispositivos virtuales.	64
Figura 27. Ventana para escoger las especificaciones del dispositivo que se desea virtualizar.....	65

Figura 28. Máquina virtual de una Tablet.	65
Figura 29. Prototipo pantalla de IntroAPP.	71
Figura 30. Prototipo pantalla menú.	71
Figura 31. Prototipo pantalla datos de usuario.	72
Figura 32. Prototipo pantalla de recomendación.	73
Figura 33. Prototipo pantalla prueba auditiva.	74
Figura 34. Prototipo pantalla respuesta.	74
Figura 35. Prototipo pantalla configuración.	75
Figura 36. Prototipo pantalla información.	75
Figura 37. Prototipo pantalla ayuda.	76
Figura 38. Grafica de ejemplo con GraphView.	79
Figura 39. Compilacion de la Aplicación.....	87
Figura 40. Pruebas realizadas en una Samsung Galaxy Tab 3 T211	88
Figura 41. Limpieza auditiva	93
Figura 42.Resultados del paciente 1.....	95
Figura 43.Resultados del paciente 2.....	96
Figura 44.Resultados del paciente 3.....	96
Figura 45.Resultados del paciente 4.....	97
Figura 46.Resultados del paciente 5.....	98
Figura 47.Resultados del paciente 6.....	98
Figura 48.Resultados del paciente 7.....	99

ÍNDICE DE TABLAS

Tabla 1. Modelos de arquitectura para la creación de vistas	15
Tabla 2. Comparaciones entre Metodología XP vs Metodología Mobile-D vs Metodología SCRUM.....	31
Tabla 3. Plan de iteraciones	51
Tabla 4. Identificación de los actores.....	51
Tabla 5. Especificación caso de uso N°1.....	53
Tabla 6. Especificación caso de uso N°2.....	53
Tabla 7. Especificación caso de uso N°3.....	54
Tabla 8. Especificación caso de uso N°4.....	55
Tabla 9. Resultados de la encuesta.....	56
Tabla 10. Requerimientos mínimos del dispositivo	57
Tabla 11. Versión de Android y su porcentaje de dispositivos activos	58
Tabla 12. Matriz de Stakeholders del proyecto	80
Tabla 13. Resumen de las tareas de los involucrados.....	81
Tabla 14. Interacción con la interface de “prueba de audición”.	88
Tabla 15. Análisis de resultados de la primera prueba funcional.....	90
Tabla 16. Resultados prueba de audiometría.....	94
Tabla 17. Requerimientos mínimos de Android Studio.	118

ÍNDICE DE DIAGRAMAS

Diagrama 1. Caso de uso.....	52
Diagrama 3. Diagrama de clases.....	68
Diagrama 4. Diagrama de componentes, pruebas guardadas.....	69
Diagrama 5. Diagrama de actividades, inicio prueba audiometría tonal.....	69
Diagrama 6. Diagrama de navegación.	70

ÍNDICE DE ANEXOS

Anexo. A	107
Anexo. B	112
Anexo. C	118
Anexo. D	120
Anexo. E	126
Anexo. F	140
Anexo. G	151
Anexo. H	155
Anexo. I	161

RESUMEN

Este trabajo de titulación tiene el propósito de desarrollar un prototipo de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista.

Para el desarrollo de la aplicación se utilizó la metodología de desarrollo XP (programación extrema), se inició el desarrollo del proyecto con el levantamiento de los requerimientos de usuario y del especialista en audiometría, mediante una encuesta y una entrevista respectivamente, luego se realizó el diseño con el modelado 4+1 vistas de Kruchten y para la diagramación se usó el lenguaje de modelado unificado (UML), el desarrollo se realizó con la plataforma Android Studio y para la generación de tonos puros se utilizó librerías específicas de Matlab y finalmente se realizó la implementación de la aplicación con la herramienta de virtualización de Android Studio.

El funcionamiento de la aplicación se verificó mediante pruebas de aplicación móvil en un celular Samsung J2 Prime vs un aparato audiométrico convencional y se obtuvieron los mismos resultados en cuanto al nivel de pérdida de audición de los pacientes, pero existieron mínimas variaciones en los decibeles de ciertas frecuencias con respecto al audiómetro convencional.

PALABLAS CLAVES: Audiometría de tonos, dispositivo móvil, aplicación móvil, diagrama audiométrico, pérdida de audición, tonos puros.

ABSTRACT

This titling work has the purpose of developing a final prototype of a mobile audiometry application for the consumer with the possibility of sending the information to a specialist.

For the development of the application the XP methodology was used (extreme programming), the development of the project started with the elicitation of the requirements from the user and the audiometric specialist, by means of a survey and an interview respectively, then the design with the modeling 4 + 1 views of Kruchten and for the layout the unified modeling language (UML) was used, the development was made with the Android Studio platform and for the generation of pure tones we used specific libraries of Matlab and finally made the implementation of the application with the virtualization tool of Android Studio.

The operation of the application was verified by mobile application tests on a Samsung J2 Prime phone vs a conventional audiometric device and the same results were obtained regarding the level of hearing loss of the patients, but there were minimal variations in the decibels of certain frequencies with respect to the conventional audiometer.

KEYWORDS: Tone audiometry, mobile device, mobile application, audiometric diagram, hearing loss, pure tones

INTRODUCCIÓN

Según datos de la (“OMS | Sordera y pérdida de la audición,” 2017), más de 360 millones de personas en todo el mundo tienen problemas de pérdida de audición, la mayoría de ellas ocasionadas por mala limpieza del conducto auditivo o factores como la escucha excesiva de música a alto volumen, por el uso de smartphones o reproductores de música portátiles. Las pérdidas auditivas pueden ser leves, moderadas, moderadas severas, severas y profundas.

En atención a esta problemática, se ha propuesto el presente trabajo de titulación, que se enfocará en el desarrollo de una aplicación móvil, cuyo funcionamiento será similar al aparato audiométrico de tonos puros, usado por los especialistas para identificar los distintos problemas de audición, su operación se realiza mediante la escucha por el paciente de tonos en diferentes frecuencias, lo cual permite determinar la pérdida de audición en cada uno de los oídos, generando como resultado un diagrama audiométrico donde se refleja el nivel de audición del paciente.

Esta aplicación será de gran utilidad tanto para especialistas como para los pacientes y la sociedad en general que quiera realizarse exámenes para conocer el estado de su salud auditiva; pues podrán tener acceso a una solución efectiva de forma rápida y de bajo costo, que según datos de la (“OMS | Sordera y pérdida de la audición,” 2017) el costo es una de las principales razones por las cuales las personas no se realizan exámenes a tiempo para evitar que los problemas se agraven. Por otra parte, permitirá que las personas que tienen hábitos inadecuados de salud auditiva tengan una alerta temprana para que se pueda modificar dichos hábitos.

El presente trabajo tiene por objetivo desarrollar de un prototipo de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista.

Para la consecución del objetivo detallado la metodología utilizada consistió en el estudio de los distintos problemas auditivos y el funcionamiento de un equipo de audiometría, con los que

trabaja la audiometría convencional, luego se iniciara con el desarrollo de la aplicación móvil, y se usara la metodología de Desarrollo de Software Programación Extrema (XP).

Este trabajo de titulación está dividido en 5 capítulos:

Capítulo 1. En esta fase se desarrolló el “Marco Teórico”, el cual comprende arquitecturas de software, patrones de diseño y metodologías de desarrollo para dispositivos móviles y la información general de audiometría, la cual resulta imprescindible para entender el funcionamiento y complejidad del aparato auditivo.

Capítulo 2. En esta siguiente fase se desarrolló el “Estudio de Necesidades”, donde se determinó requerimientos de usuario y de un especialista en audiometría además de los requerimientos de hardware para la aplicación.

Capítulo 3. En esta fase se desarrolló el “Diseño de la aplicación” el cual esta subdividido en arquitectura, datos, interfaces, implantación.

Capítulo 4. En esta fase se desarrolló la “Construcción” la cual incluye librerías, desarrollo, fase de codificación, fase de pruebas, documentación. De las cuales se obtiene las librerías que se usó para desarrollar la aplicación con todas las fases que se siguen en la metodología de desarrollo XP.

Capítulo 5. En esta última fase se desarrolló la “Validación”, en la cual se realizaron pruebas entre el aparato de audiometría convencional vs la aplicación móvil.

Cabe recalcar que para la realización del presente trabajo la mayor parte de dificultades se presentaron en la fase de validación, ya que se requiere de un número grande de personas para garantizar una óptima validación, misma que incurre en costos elevados a la hora de realizar pruebas con los equipos convencionales.

JUSTIFICACIÓN

De acuerdo a la (“OMS | Sordera y pérdida de la audición,” 2017), más del 5% de las todas las personas en el mundo padecen de pérdida de audición discapacitante; es decir, alrededor de 360 millones de personas; de los cuales 32 millones son niños y 328 millones son adultos.

Al menos el 60% de todos los casos de pérdida de audición en niños se debe a causas que pueden ser prevenibles, si se realiza los exámenes necesarios a su debido tiempo.

Especialistas en temas de audición entrevistados, indican que mucha gente solo se acerca a realizarse un examen cuando ya tiene un problema y en ocasiones es muy tarde para realizarle un tratamiento.

Este trabajo de titulación tiene el propósito de ayudar a las personas a obtener un diagnóstico temprano de una forma fácil y a bajo costo; ya que el diagnóstico de un problema auditivo y el examen con un aparato audiométrico es muy costoso, lo cual imposibilita que todas las personas puedan acceder a ellos.

Mediante el uso de esta aplicación las personas pueden determinar, si tienen o no algún problema auditivo, obteniendo información real del nivel de pérdida auditiva, ya que genera una ficha detallada de los niveles de pérdida de audición y recomendaciones para tratar el problema; lo cual es de gran ayuda para que un especialista pueda tratar el caso.

Por otro lado, esta aplicación contribuirá a concientizar al usuario de los problemas que ocasionan no tener un buen cuidado auditivo.

OBJETIVOS

Objetivo General

Desarrollo de un prototipo final de una aplicación móvil de audiometría para el consumidor con posibilidad de envío de la información a un especialista.

Objetivos Específicos

- Estudiar los diferentes tipos de problemas de audición que se pueden identificar mediante la audiometría de tonos y de esta manera poderlos detallar en la aplicación, para que el usuario tenga claro que hace la aplicación.
- Estudiar el funcionamiento de un equipo de audiometría y con esta información desarrollar la aplicación.
- Diseñar la aplicación móvil de audiometría para obtener una aplicación funcional.
- Desarrollar la aplicación móvil de audiometría, para tener una aplicación de igual funcionamiento que el equipo audiométrico convencional.
- Implementar la aplicación móvil de audiometría en un dispositivo móvil, para verificar su funcionamiento en condiciones reales.
- Evaluar la aplicación móvil de audiometría, para obtener resultados de funcionamiento de la aplicación vs el equipo audiométrico

CAPITULO I.
MARCO TEÓRICO

En el primer capítulo se expone los diferentes aspectos teóricos que son la base para iniciar el análisis, aspectos relacionados con: arquitecturas de aplicaciones móviles, patrones de diseño y metodologías de desarrollo de software específicas para aplicaciones móviles.

1.1. Arquitectura de software.

La arquitectura de aplicaciones de software comprende el proceso de definición de una solución estructurada, misma que cumple con todos los requisitos técnicos y operativos, a la vez que optimiza los atributos de calidad comunes; tales como: la seguridad, el rendimiento y la capacidad de administración. Implica la toma de una serie de decisiones basadas en una gran cantidad de factores, cada una de las cuales puede tener un impacto significativo en el rendimiento, la calidad, la capacidad de mantenimiento y el éxito general de la aplicación.

"La arquitectura del software abarca el conjunto de decisiones importantes sobre la organización de un sistema de software, incluida la selección de los elementos estructurales y sus interfaces mediante los cuales se compone el sistema; comportamiento como se especifica en colaboración entre esos elementos; composición de estos elementos estructurales y de comportamiento en subsistemas más grandes; y un estilo arquitectónico que guía a esta organización. La arquitectura de software también implica funcionalidad, facilidad de uso, flexibilidad, rendimiento, reutilización, comprensibilidad, restricciones económicas y tecnológicas, concesiones y preocupaciones estéticas. "(Itectu, n.d.)

La arquitectura de software permite identificar los componentes que realizan alguna tarea computacional, sus interfaces y la forma como se comunican entre ellos, de una manera abstracta. Toda arquitectura debe ser implementable en una arquitectura física.

"La arquitectura del software de un programa o sistema informático es la estructura o estructuras del sistema, que comprende elementos de software, las propiedades visibles externamente de esos elementos y las relaciones entre ellos. La arquitectura se ocupa del lado público de las interfaces; los detalles privados de los elementos los detalles que tienen que ver únicamente con la implementación interna no son arquitectónicos ".(Itectu, n.d.)

1.1.1. Arquitecturas para aplicaciones móviles.

Una buena arquitectura es imprescindible para que todo este organizado de manera correcta y con ello la aplicación también lo esté, caso contrario aparecerán los siguientes problemas:

- Falta de cobertura del código por pruebas unitarias.
- Dificultad para depurar una clase por la gran cantidad de funciones.
- Imposibilidad de seguimiento de la lógica dentro de esa gran clase.
- Dificultad para que un nuevo programador pueda mantener y agregar nuevas funciones a su trabajo.

1.1.2. Beneficios de contar con una arquitectura adecuada.

Si se pretende construir una aplicación de alta calidad, es necesario realizar una arquitectura adecuada y efectiva; misma que ofrece los siguientes beneficios:

- **Simplicidad:** Es necesario separar y definir un único rol, que sea claro para cada componente de la aplicación; ya que permitirá saber qué hace y que hay dentro de cada clase.
- **Estabilidad:** Los códigos que se escriban deben ser comprobables, para posteriormente pruebas unitarias.
- **Mantenimiento de bajo costo:** Permite realizar un seguimiento de las partes lógicas importantes; así como, agregar o eliminar funciones.

1.2. Arquitectura de aplicaciones móviles.

Las aplicaciones móviles se dividen en dos grupos que son:

- Aplicaciones móviles auto contenidas
- Aplicaciones con conexión a internet.

La implementación de estas depende del tipo de información que se pretenda mostrar; tomando siempre en cuenta la escalabilidad y las ventajas de desarrollar una u otra.

1.2.1. Aplicaciones móviles auto contenidas.

En este tipo de aplicaciones móviles, el contenido es estático; sus imágenes, información y menús, casi nunca cambia o rara vez lo hace. La razón es que todo su contenido se encuentra autocontenidas dentro de la misma aplicación y no necesita valerse de una herramienta o un servicio para que pueda funcionar.

Estas aplicaciones son del tipo nativo; es decir, que puede hacer uso de todas las herramientas y utilidades con que cuenta el dispositivo móvil. Un claro ejemplo es el desarrollo de una calculadora para los sistemas operativos: Android, IOS o Windows Phone, donde la aplicación no tiene la necesidad de cambiar los números debido a que información puede ser suministrada a través de una manual en alguna función u opción del menú. Así mismo, si necesita brindar herramientas como por ejemplo tema; se puede predeterminedir colores, tamaños y demás cosas al respecto usando las herramientas con que ya cuenta el dispositivo móvil.

Ventajas

- Desarrollo relativamente sencillo.
- No requiere de gastos en servicios de hosting, bases de datos u otros.

Desventajas

- Si se requiere actualizar la información en la aplicación, es necesario compilar el código nuevamente y subirla a la plataforma.
- Si la información que es utilizada por la aplicación es extensa, es probable que la aplicación necesite cambios periódicos, lo que podría volverla inviable.

1.2.2. Aplicaciones con conexión a internet.

En este tipo de aplicaciones la información se encuentra alojada en un back-end, mismo que cuenta con una base de datos, que contiene la información, la cual, accedida a través de un API de servicios web (SOAP o REST), comunican a la base de datos con el front-end de la aplicación, que es donde se puede visualizar la información solicitada.

Aplicaciones de redes sociales, como Facebook o Twitter, aplicaciones de mensajería instantánea como WhatsApp; requieren el servicio de Internet para poder acceder a información específica o realizar alguna acción.

Este tipo de aplicaciones se clasifican en:

- Con descarga de información estática
- Con descarga de información dinámica.

Aplicaciones móviles con descarga de información estática

Cuenta con un archivo en formato XML o cualquier otro formato que se utilice, y que se modifica de forma manual cada vez que se requiera cambiar información.

Ventajas

- No es necesario publicar una versión nueva cada vez que se requiera dar a conocer una nueva información.

Desventajas

- Requiere un gasto para implementar una base de datos, hosting, entre otros.
- Si la información modificada es errónea, será necesario corregir dicha información hasta que sea correcta.

Aplicaciones móviles con información dinámica

En este tipo de aplicaciones la información esta almacenada en una base de datos la cual está alojada en un servidor y que se accede a esta a través de servicios API los cuales comunican a la base con la interfaz de la aplicación.

Ventajas

- Los cambios se realizan de manera fácil.
- Permite editar la información en la base de datos a través de una plataforma específica.
- Contando con permisos especiales, se puede realizar tareas desde cualquier lugar.

Desventajas

- Debido a la necesidad de contratación de personal para el desarrollo web y la contratación de alguna de empresa de hosting, el costo es superior.
- El desarrollo de la aplicación es más complejo.

1.2.3. Patrones de arquitectura software para desarrollar aplicaciones.

A continuación, se menciona a los patrones de arquitectura de software más utilizados para aplicaciones móviles:

- Modelo - Vista - Controlador (MVC)
- Modelo - Vista - Presentador (MVP)
- Modelo - Vista - ViewModel (MVVM)

- Vista - Presentador (VP)

“En el caso del MVC, en el patrón de diseño se separan los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes. Cuando la lógica de negocio realiza un cambio, se requiere que esta a su vez sea la que actualiza la vista.”(GILBER ARANGUREN, n.d.)

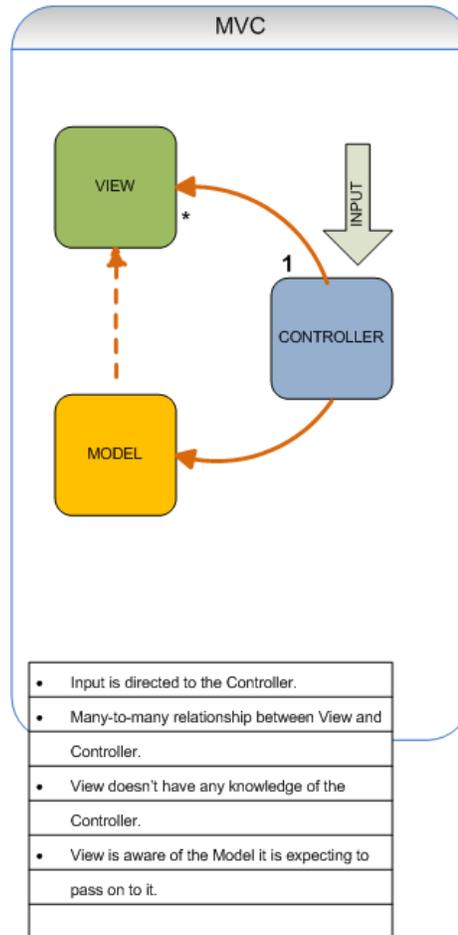


Figura 1. Modelo vista controlador

Fuente: (Tomy Rhymond, 2011)

Elaborado por: (Tomy Rhymond, 2011)

El MVP, nace por la necesidad de ayudar a realizar pruebas automáticas de la interfaz gráfica, con lo cual, la idea es codificar la interfaz de usuario y hacerla lo más simple posible, teniendo la menor cantidad de código posible, de forma que no sea necesario probarlo. En lugar de esto, toda la lógica de la interfaz de usuario se hace en una clase separada, a la cual se la conoce como “Presentador”, esta no dependerá de los componentes de la interfaz gráfica, por lo cual, será más fácil realizar pruebas.

La idea es que la clase “Presentador” haga la función de intermediario, entre el modelo de datos y la vista (la interfaz gráfica de usuario). La vista tiene métodos en los que le pasan los datos que debe pintar ya “mascados”. Únicamente se debe meter esos datos en los componentes gráficos. También métodos “get” para obtener el contenido de los componentes. (“Patrón MVP - Blog de Imaginanet. Artículos especializados en soluciones y desarrollo web.,” 2012)

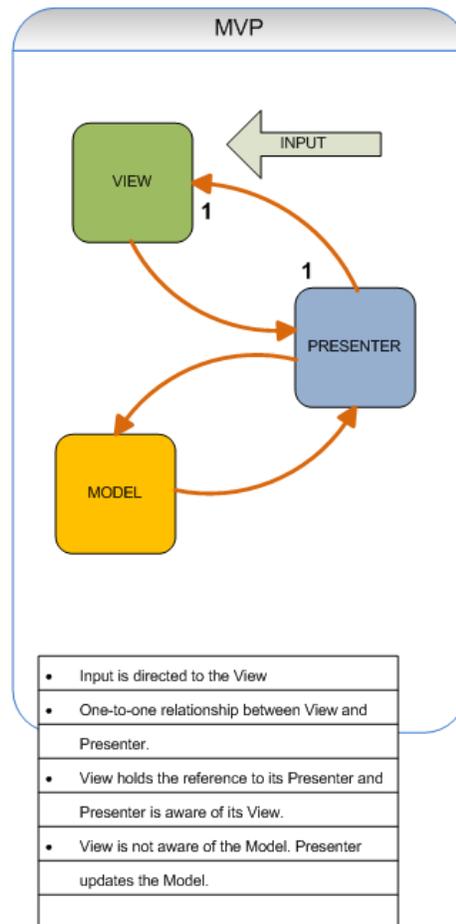


Figura 2. Modelo vista controlador

Fuente:(Tomy Rhymond, 2011)

Elaborado por: (Tomy Rhymond, 2011)

MVVM “significa Modelo Vista Vista Modeló, el patrón de diseño se separa la interfaz de usuario y los datos de la aplicación, pero en lugar de controlar de forma manual los cambios en la vista o en los datos, estos se actualizan automáticamente cuando hay un cambio en ellos; es decir, que si por ejemplo la vista actualiza un dato que está presentando se actualiza también el modelo automáticamente y viceversa.” (GILBER ARANGUREN, n.d.)

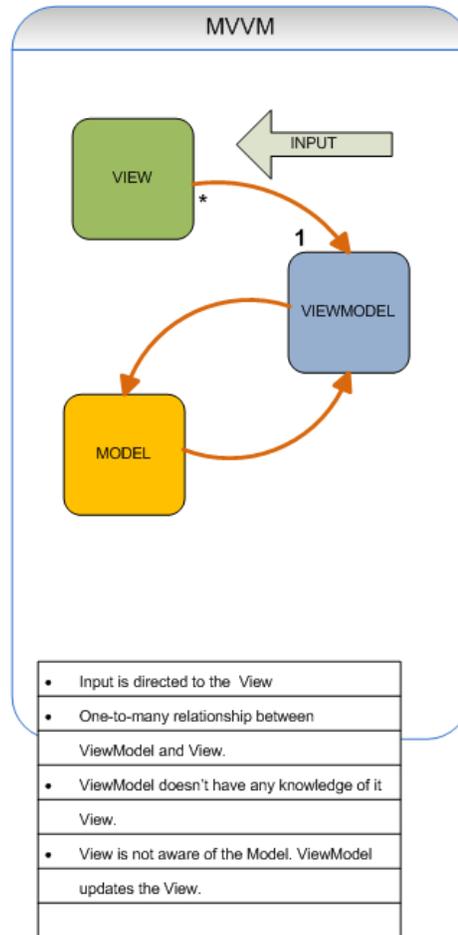


Figura 3. Modelo vista vista modeló
Fuente:(Tomy Rhymond, 2011)
Elaborado por: (Tomy Rhymond, 2011)

1.2.4. Modelos de arquitecturas.

La necesidad de descomponer un problema en varios elementos que nos ayudan a comprenderlo de mejor manera, no es algo nuevo, ya desde publicaciones realizadas en la década de los 80, en donde se afirma que la descomposición de un sistema en estructuras es esencial para su entendimiento y a estas estructuras se las denomina vistas, como se cita a Perry y Wolf en Limón Cordero: “El arquitecto del software necesita un número de vistas diferentes de la arquitectura de software para varios usos y usuarios...son requeridas para enfatizar y entender diferentes aspectos de la arquitectura”. (Ayala, 2014)

Existen varias propuestas que trabajan con el concepto de vistas, así tenemos:

Tabla 1. Modelos de arquitectura para la creación de vistas

Autor(es)	Nombre	Año
Kruchten	4+1 vistas	1995
Hofmeister, Nord y Soni	Siemens1	1999
Clements	SEI2	2003

Fuente: Guía didáctica de Arquitectura de Aplicaciones
Elaborado por: (Ayala, 2014)

El modelo de 4+1 vistas.

Este modelo describe la arquitectura del software usando cinco vistas concurrentes. Tal como se muestra en la **Figura. 4**, cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema.

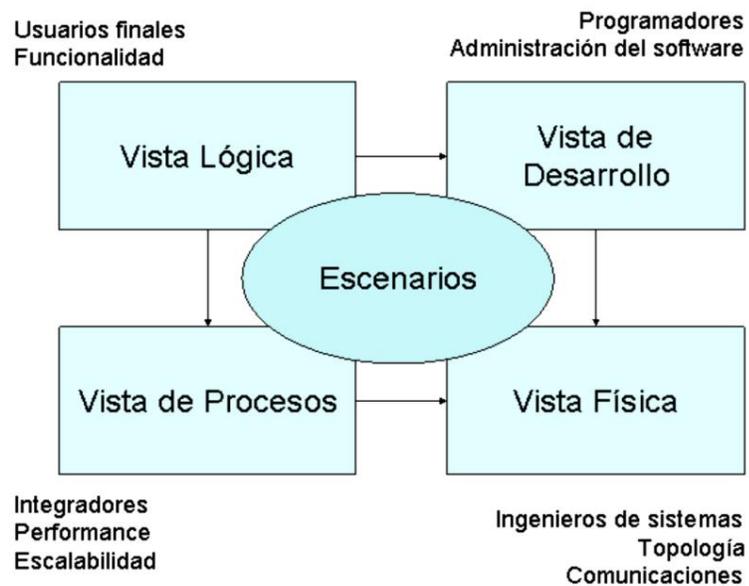


Figura 4. Modelo de “4+1” vistas

Fuente: (Kruchten, 1995)

Elaborado por: (Kruchten, 1995)

Vista lógica

La vista lógica apoya principalmente los requisitos funcionales; es decir, lo que el sistema debe ofrecer en términos de servicios a sus usuarios. El sistema se descompone en una serie de abstracciones clave, tomadas principalmente del dominio del problema, en forma de objetos o clases de objetos. Aquí se utilizan los principios de abstracción, encapsulamiento y herencia. Esta descomposición sirve no solo para potenciar el análisis funcional, sino que también se utiliza para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema.

Vista de procesos

A la arquitectura de procesos se la describe en varios niveles de abstracción, donde cada uno de los niveles representa distintos intereses. Al nivel más alto de la arquitectura de procesos se lo puede ver como un conjunto de redes lógicas de programas que se comunican, ejecutándose de forma independiente, y que se encuentran distribuidos en un conjunto de recursos de hardware, lo cuales están conectados por un bus, una LAN o WAN. Varias redes lógicas pueden ser usadas para apoyar la separación de la operación del sistema que trabaja en línea, del sistema que lo hace fuera de línea; así como, para apoyar la coexistencia de versiones de software de simulación o de prueba.

A la agrupación de tareas que forman una unidad ejecutable, se las conoce como proceso. Los procesos son el nivel al que la arquitectura puede ser controlada tácticamente para comenzar, recuperar, reconfigurar, y detener. Los procesos se pueden replicar para mejorar la disponibilidad o aumentar la distribución de la carga de procesamiento.

Vista de desarrollo

Se centra en la organización real de los módulos de software en el ambiente de desarrollo. El software se empaqueta en pequeñas partes, bibliotecas de programas o subsistemas, que pueden ser desarrollados por uno o unos cuantos desarrolladores. Los subsistemas se organizan en una jerarquía de capas, donde cada una de ellas ofrece una interfaz estrecha y bien definida hacia las capas superiores.

Esta vista se representa en diagramas de módulos o subsistemas que muestran las relaciones “exporta e importa”. Solamente cuando todos los elementos del software han sido identificados es que se puede describir completamente a la arquitectura de desarrollo. Sin embargo, es factible listar las reglas que rigen la arquitectura de desarrollo partición, agrupamiento y visibilidad antes de conocer todos los elementos.

Vista física

La arquitectura física considera en primer lugar los requisitos no funcionales del sistema, como la disponibilidad, confiabilidad (tolerancia a fallas), performance (throughput), y escalabilidad. El software se ejecuta sobre una red de computadores o nodos de procesamiento. Los elementos identificados (redes, procesos, tareas y objetos) deben ser mapeados en los nodos. El mapeo del software en los nodos necesita ser altamente flexible con un impacto mínimo en el código fuente como tal.

Vista de escenarios

Mediante la utilización de un conjunto de casos de uso es que se ilustra la descripción de la arquitectura, o con el uso de escenarios, lo cual genera una quinta vista. Los escenarios describen secuencias de interacciones entre objetos, y entre procesos; estos sirven para identificar y validar el diseño de la arquitectura. También sirven como punto de partida para probar prototipos de arquitectura.

1.2.5. Estilos arquitectonicos.

Los estilos arquitectónicos tienen muchas definiciones y conceptos definidos por los grandes desarrolladores de la ingeniería de software. Por ejemplo “un estilo arquitectónico define una familia de sistemas en términos de patrón de organización estructural. Específicamente, un estilo arquitectónico determina el vocabulario de componentes y conectores que puede ser usado, así como un conjunto de restricciones de cómo pueden ser combinados”.

Los estilos arquitectónicos están basados en los patrones de arquitecturas que se usen. Los estilos agrupan clases, englobando una serie de estilos arquitectónicos que comparten características. Generalmente los estilos proveen guías para crear una clase amplia de arquitectura, donde los patrones se enmarcan en darle solución a problemas más pequeños y más específicos dentro de un estilo dado.

Estilos de llamada y retorno.

- **Modelo vista controlador:** Conocido más bien como Model View Controller (por sus siglas en inglés, MVC). Este patrón separa el modelado del dominio, la presentación y las acciones basadas en la entrada de datos por el usuario en tres clases diferentes. El modelo está encargado de administrar el comportamiento y los datos del dominio de aplicación, respondiendo a requerimientos de información sobre su estado y a instrucciones de cambiar su estado (normalmente desde el controlador); la vista que controla la visualización de la información y el controlador que interpreta las acciones del ratón y del teclado, informando al modelo y/o a la vista para que cambien su estado según les sea factible.
- **Ciente-servidor:** Es una arquitectura compuesta por dos componentes fundamentales, uno de estos elementos es el cliente que puede estar compuesto por una o varias computadoras conectadas en Red que solicitan servicio al otro componente, que es el servidor, que no es más que otra computadora en la misma Red.

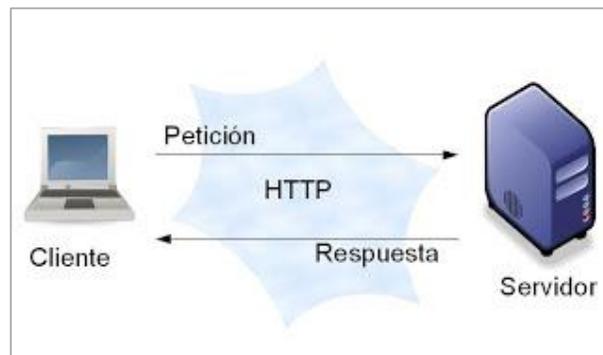


Figura 5. Modelo cliente-servidor
 Fuente: (Yoandy Campos Diaz, 2008)
 Elaborado por: (Yoandy Campos Diaz, 2008)

Este tipo de arquitectura permite que los usuarios puedan acceder a los mismos datos al establecer conexión con el servidor, este brinda un acceso transparente a las aplicaciones que se ejecutan en el, controlando el acceso a los recursos. Esta arquitectura se basa en procesos distribuidos, procesos de pedidos y respuestas. En la actualidad se hace uso de esta tecnología, por ejemplo, la Internet en el amplio sentido de la palabra está basada en esta tecnología. Esta arquitectura es recomendable para aplicaciones Web.

- **Arquitectura en capas:** Es una organización jerarquizada, donde una capa le brinda servicios única y exclusivamente a su capa superior y es provista de su capa inferior. Las interacciones entre capas generalmente son invocadas por métodos, donde normalmente las capas inferiores no pueden utilizar las funcionalidades de los niveles superiores. Este tipo de estilo permite desarrollar la aplicación a través de módulos, facilitando a la vez la corrección de errores y brindar un mejor soporte al sistema una vez finalizado.
- **Arquitectura orientada a objetos:** En este estilo los componentes se basan en los principios orientados a objetos (OO). En una forma simple un diseño orientado a objetos permite diseñar sistemas para encapsular los datos y los objetos proveen explícitamente interfaces a otros objetos, las interfaces están separadas de las implementaciones. Pero la principal característica de este estilo es que se puede modificar la implementación de un objeto sin afectar la interfaz. Hay muchas variantes en este estilo ya que algunos sistemas permiten que los objetos sean tareas concurrentes, otros admiten que los objetos contengan diferentes interfaces.

Estilos de código móvil

- **Arquitectura de máquinas virtuales:** En la nueva estrategia arquitectónica de Microsoft la máquina virtual por excelencia tiene gran relación con el Common Language Runtime (CLR), El CLR admite programación orientada a objetos (C#, F#, C++, Java, Python) y otros paradigmas puros y templados. Con una máquina virtual común el proceso evita la redundancia de compiladores compitiendo por recursos, unificando depurador y profilers.

Estilo orientado a servicios.

- **Una arquitectura orientada a servicios (Service Oriented Architecture SOA).** Esta es una arquitectura en la que se pueden establecer conexiones para consumir servicios que provee un servidor determinado, estos servicios pueden ser consumidos por terceras personas ya que esta arquitectura permite brindar información en interfaces estándares. En esta arquitectura hay tres componentes que juegan un papel importante que son: el proveedor del servicio, el registro del servicio y el consumidor del servicio. Generalmente a estos servicios son conocidos como webservices (servicios web) usando como formato principal para el intercambio de datos el estándar XML.

1.3. Patrones de diseño.

Constituyen la descripción de un problema que ocurre de manera reiterada en el entorno y que se pueden usar para diseñar aplicaciones móviles. Este es un pequeño resumen de soluciones de diseño que se han demostrado su funcionalidad más de una vez y pueden ser usadas como guía para desarrollar una aplicación móvil.

1.3.1. Componentes de un patrón de diseño.

El patrón de diseño debe tener los siguientes elementos:

- **Nombre:** Permite describir al patrón, para ello generalmente se utilizan una o dos palabras para ello.
- **Problema:** Describe cuándo aplicar el patrón. Explica el problema y en el contexto en el que se halla, este puede incluir una serie de condiciones que deben darse para que tenga sentido el patrón.

- **Solución:** Describe los elementos que forman el diseño, las relaciones, responsabilidades y colaboraciones.
- **Consecuencias:** Constituyen los resultados, identificando tanto las ventajas como desventajas de aplicar el patrón, son fundamentales para poder evaluar las alternativas de diseño y determinar los costos y beneficios de aplicarlo. Las consecuencias que tiene un patrón contienen su impacto sobre la flexibilidad, extensibilidad y portabilidad de este en un sistema.

1.3.2. Patrones del desarrollo del software para aplicaciones.

Existen muchos patrones de diseño los cuales se clasifican en:

- **Patrones Creacionales**
Inicialización y configuración de objetos.
- **Patrones Estructurales**
Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes. Separan la interfaz de la implementación.
- **Patrones de Comportamiento**
Describen la comunicación entre clases u objetos.

1.3.2.1. *Patrones creacionales.*

- **Fábrica Abstracta (Abstract Factory)**
Se encarga de crear diferentes familias de objetos, como la creación de interfaces gráficas de distintos tipos; entre ellos, ventana, menú, botón entre otros.
- **Método de Fabricación (Factory Method)**
Parte del principio de que las subclases determinan la clase a implementar.
- **Prototipado (Prototype)**
Se basa en la clonación de ejemplares copiándolos de un prototipo.
- **Singleton**
Restringe la instanciación de una clase o valor de un tipo a un solo objeto.
- **MVC (Model View Controler)**
Ese patrón plantea la separación del problema en tres capas: la capa **model**, que representa la realidad; la capa **controler**, que conoce los métodos y atributos del modelo, recibe y realiza lo que el usuario quiere hacer; y la capa **vista**, que muestra un aspecto del modelo y es utilizada por la capa anterior para interactuar con el usuario.

1.3.2.2. *Patrones estructurales.*

- **Adaptador (Adapter)**
Convierte una interfaz en otra.
- **Puente (Bridge)**
Desacopla una abstracción de su implementación permitiendo modificarlas independientemente.
- **Objeto Compuesto (Composite)**
Es utilizado para construir objetos complejos a partir de otros más simples, para lo cual utiliza la composición recursiva y una estructura de árbol.
- **Envoltorio (Decorator)**
Permite añadir dinámicamente funcionalidad a una clase existente, evitando heredar sucesivas clases para incorporar la nueva funcionalidad.
- **Fachada (Facade)**
Permite simplificar la interfaz para un subsistema.
- **Peso Ligero (Flyweight)**
Elimina la redundancia o la disminuye cuando existe una gran cantidad de objetos con información idéntica.
- **Apoderado (Proxy)**
Un objeto se aproxima a otro.

1.3.2.3. *Patrones de comportamiento.*

- **Cadena de responsabilidad (Chain of responsibility)**
El objetivo es permitir que más de un objeto tenga la posibilidad de atender una petición.
- **Orden (Command)**
Encapsula una petición como un objeto dando la posibilidad de “deshacer” la petición.
- **Intérprete (Interpreter)**
Interpreta el lenguaje para una gramática simple y sencilla.
- **Iterador (Iterator)**
Sin exponer la estructura interna, define una interfaz que declara los métodos necesarios para acceder secuencialmente a una colección de objetos.
- **Mediador (Mediator)**
Permite la interacción de varios objetos, sin generar acoples fuertes en esas relaciones. Coordina las relaciones entre sus asociados.

- **Recuerdo (Memento)**
Almacena el estado de un objeto y lo restaura posteriormente.
- **Observador (Observer)**
Notifica acerca de cambios de estado de un objeto.
- **Estado (Server)**
Se utiliza cuando el comportamiento de un objeto cambia dependiendo del estado de este.
- **Estrategia (Strategy)**
Maneja la selección de un algoritmo.
- **Método plantilla (Template Method)**
Algoritmo con varios pasos suministrados por una clase derivada.
- **Visitante (Visitor)**
Operaciones aplicadas a elementos de una estructura de objetos heterogénea.

1.4. Metodologías para aplicaciones móviles.

El desarrollo de software para móviles como el desarrollo de software convencional tiene grandes similitudes. Sin embargo, existen algunas diferencias como la corta duración de sus desarrollos, una gran competencia por las innovaciones, cambios frecuentes en las plataformas de desarrollo y hardware de los dispositivos; razón por la cual, es necesario tener en cuenta todos estos factores antes de elegir una metodología concreta de desarrollo.

1.4.1. Definición de metodología de desarrollo.

“Una metodología es un conjunto de procedimientos, herramientas, técnicas y documentos auxiliares para implementar nuevos sistemas de información. Una metodología está formada por fases, las cuales pueden subdividirse en sub-fases, que servirá para planificar, gestionar, controlar y evaluar el proyecto de desarrollo.” (Avison & Fitzgerald, 2006)

1.4.2. Tipos de metodologías de desarrollo de aplicaciones móviles.

Existen muchas metodologías de desarrollo cada una con puntos débiles y fuertes.

Algunos de los métodos más usados para desarrollo de software móvil son:

- Desarrollo ágil (Programación Extrema XP, Scrum, etc.)
- Mobile-D

Una de las características importantes de la gran mayoría de los desarrollos móviles es su corta duración.

Las metodologías ágiles como lo menciona (rebaza, n.d.) son adecuadas para el desarrollo de aplicaciones móviles, ya que el cambio constante del entorno hace que constantemente se deba adaptar a nuevos dispositivos, cambios en las plataformas de desarrollo o en el entorno de desarrollo como tal. Por este motivo se requiere una respuesta rápida y ágil al cambio más que al seguimiento de un plan de desarrollo.

El desarrollo ágil basa su desarrollo en iteraciones, donde en cada iteración se realizan todas las fases de ciclo de desarrollo.

Gracias a esta metodología las interacciones en el proceso y las herramientas son más controlables y es posible una fluida comunicación si existen varios desarrolladores trabajando en una aplicación. Una aplicación se puede realizar en periodos de desarrollo cortos alrededor de un mes o seis meses. Con el propósito de una realimentación rápida es posible realizar varias actualizaciones según se van entregando funcionalidades.

1.4.2.1. **Metodología XP.**

Programación extrema o Xp es la metodología más recomendada para desarrollo de aplicaciones donde la cantidad de desarrolladores es de 1 a 2. Esta metodología fue formulada por Kent Beck.

Según (“Extreme Programming: A Gentle Introduction.,” 2013) la Programación Extrema es la más recomendada de los procesos ágiles de desarrollo de software, por su enfoque en la adaptabilidad. Debido a que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable y hasta deseable del desarrollo de proyectos.

Por lo antes mencionado el presente trabajo se fundamenta en la metodología XP, con el objeto de conseguir un desarrollo ágil y dinámico.

La Programación Extrema se basa en:

- **Desarrollo en iteraciones:**

En cada iteración se agregan nuevas funcionalidades, o se corrigen errores generando así distintas versiones.

- **Pruebas unitarias continuas:**
Orientadas a comprobar que la aplicación mantenga sus funcionalidades.
- **Programación en parejas:**
Generalmente se realiza en pareja con el objeto de generar código de mejor calidad con menos errores.
- **Interacción entre los desarrolladores y el usuario:**
Permite una mayor comprensión de los problemas o necesidades de los usuarios y las soluciones que puedan brindar los desarrolladores.
- **Refactorización del código:**
Busca hacer el código más fácil y mantenible, pero debe garantizar su correcto funcionamiento manteniendo las pruebas unitarias.
- **Propiedad del código compartida:**
Busca que cualquier integrante del proyecto pueda colaborar modificando código hecho por otro. La existencia de errores se comprueba realizando pruebas.
- **Simplicidad del diseño:** Los diseños simples pero funcionales permiten que posteriores funcionalidades se puedan agregar de manera fácil y rápida.



Figura 6. Flujo de características de la programación extrema.
Fuente: Adaptado de: ("Extreme Programming: A Gentle Introduction.," 2013)
Elaborado por: Autor

Proceso XP

“Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste en los siguientes pasos:” (Sillitti, 2011)

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Se vuelve al paso 1.

Fases de desarrollo de la Programación Extrema

El ciclo de vida ideal de XP consiste en seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. (Sillitti, 2011)

Fase I: Exploración

En esta fase, el cliente plantea de manera general las historias de usuario que son de interés para la primera entrega del producto. Así mismo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

Esta fase toma entre pocas semanas a pocos meses, dependiendo del tamaño y el conocimiento que tengan los programadores con la tecnología.

Fase II Planificación de la Entrega

Aquí el cliente establece la prioridad de cada historia de usuario, con lo cual, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Fase III Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega se compone por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Fase IV Producción

Se realizan pruebas adicionales y revisiones de rendimiento antes de que el sistema sea entregado al cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

Fase V Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente.

Fase VI Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema y se enfoca en aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. Esta fase también puede ocurrir cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

1.4.2.2. Metodología Scrum.

Scrum se centra la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. Entonces, el dueño del producto decide qué elementos del backlog se deben desarrollar en el sprint siguiente. Los miembros del equipo coordinan su trabajo en un diario de stand-up de la reunión. Un miembro del equipo, el “Scrum Master” (equivalente al gerente del proyecto), es el encargado de resolver los problemas que impiden que el equipo trabaje

eficazmente(Scrum, Manuel, Gallego, Ana, & Domingo, n.d.), este equipo generalmente es de diez o menos componentes, aunque Schwaber and Beedle recomiendan equipos de cinco integrantes, dividiendo el equipo principal en equipos más pequeños si fuera necesario.

Aun con las recomendaciones que presentan Schwaber y Beedle (Scrum et al., n.d.), para los proyectos actuales y más concretamente los desarrollos para dispositivos móviles, el equipo de desarrollo se está encontrando con problemas de requisitos variables y tecnologías con características muy diferentes. En esta situación se recomienda que el primer sprint tenga una funcionalidad implementada con la tecnología que está dando problemas, se seguirá con un product backlog¹, priorizando las tareas a ejecutarse, con el objetivo de subir la moral a los desarrolladores y a todo el equipo en general.

Scrum es una metodología que goza de gran popularidad y de la cual podemos encontrar bastantes empresas que gustan de utilizarla, entre ellas podemos encontrar empresas tan importantes como Yahoo o Google. En el año 2000 Rising and Janoff publicaron tres proyectos exitosos utilizando Scrum [21].

Scrum es adecuado para aquellas empresas en las que el desarrollo de los productos se realiza en entornos que se caracterizan por tener:

- 1. Incertidumbre:** Sobre esta variable se plantea el objetivo que se quiere alcanzar sin proporcionar un plan detallado del producto. Esto genera un reto y da una autonomía que sirve para generar una “tensión” adecuada para la motivación de los equipos.
- 2. Auto-organización:** Los equipos son capaces de organizarse por sí solos, no necesitan roles para la gestión, pero tienen que reunir las siguientes características:
 - **Autonomía:** Son los encargados de encontrar la solución usando la estrategia que encuentren adecuada.
 - **Autosuperación:** Las soluciones iniciales sufrirán mejoras.
 - **Auto-enriquecimiento:** Al ser equipos multidisciplinares se ven enriquecidos de forma mutua, aportando soluciones que puedan complementarse.
- 3. Control moderado:** Se establecerá un control suficiente para evitar descontroles. Se basa en crear un escenario de “autocontrol entre iguales” para no impedir la creatividad y espontaneidad de los miembros del equipo.

4. Transmisión del conocimiento: Todo el mundo aprende de todo el mundo. Las personas pasan de unos proyectos a otros y así comparten sus conocimientos a lo largo de la organización.

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman **iteraciones** y que en Scrum se llamarán "**Sprints**".

Ciclo de desarrollo de Scrum

Para entender el ciclo de desarrollo de Scrum es necesario conocer las **5 fases** que definen el ciclo de desarrollo ágil:

1. Concepto: Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.

2. Especulación: en esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas. Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno. Esta fase se repite en cada iteración y consiste, en rasgos generales, en:

- Desarrollar y revisar los requisitos generales.
- Mantener la lista de las funcionalidades que se esperan.
- Plan de entrega. Se establecen las fechas de las versiones, hitos e iteraciones. Medirá el esfuerzo realizado en el proyecto.

3. Exploración: Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.

4. Revisión: El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.

5. Cierre: Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados "mantenimiento", que hará que el producto final se acerque al producto final deseado.

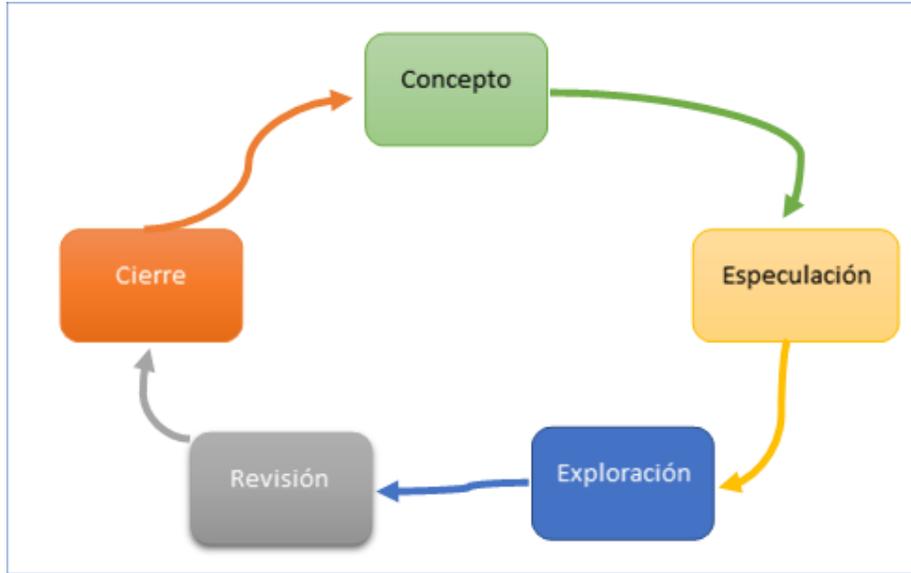


Figura 7. Ciclo de desarrollo ágil Scrum.

Fuente: Adaptado de: (Scrum et al., n.d.)

Elaborado por: Autor

Scrum gestiona estas iteraciones a través de reuniones diarias, uno de los elementos

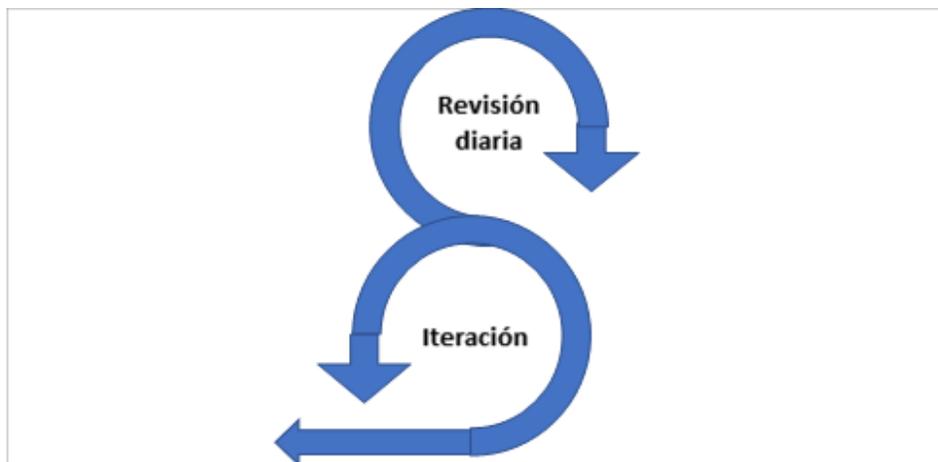


Figura 8. Ciclo principal de Scrum

Fuente: Adaptado de: (Scrum et al., n.d.)

Elaborado por: Autor

1.4.2.3. **Mobile-D.**

El método Mobile-D se desarrolló junto con un proyecto finlandés en el (Instituto de Investigación Finlandés, 2004).

Fue realizado principalmente, por investigadores de la VTT Instituto de Investigación Finlandés y, a pesar de que es un método antiguo, sigue en vigor (se está utilizando en proyectos de éxito y está basado en técnicas que funcionan).

El objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas.

Se trata de método basado en soluciones conocidas y consolidadas: Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP), XP para las prácticas de desarrollo, Crystal para escalar los métodos y RUP como base en el diseño del ciclo de vida.



Figura 9. Ciclo de desarrollo Mobile-D
 Fuente:(Vique, 2010)
 Elaborado por: (Vique, 2010)

Cada fase (excepto la inicial) tiene siempre un día de planificación y otro de entrega. Las fases son:

- **Exploración.** Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.
- **Inicialización.** Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico.
- **Productización o fase de producto.** Se repiten iterativamente las subfases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del test driven development para conseguir la mayor calidad.

- **Fase de estabilización.** Se llevan a cabo las acciones de integración para hay que asegurar que el sistema completo funciona correctamente.
- **Fase de pruebas y reparación.** Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente. (Vique, 2010).

1.4.3. Análisis comparativo.

Se presenta un análisis comparativo entre las principales características, ventajas y desventajas de algunos puntos fundamentales de las metodologías Programación Extrema (XP), Mobile-D y SCRUM; con el objeto de determinar la metodología que más se ajuste a las necesidades del presente proyecto.

Tabla 2. Comparaciones entre Metodología XP vs Metodología Mobile-D vs Metodología SCRUM

COMPARACIONES	XP	MOBILE-D	SCRUM
Desarrollo de software iterativo	✓	✓	✓
Verificación continua de la calidad	✓	✓	✓
Interacción directa con los usuarios	✓	✓	✓
Mínimo número de desarrolladores	✓		
El proceso de diseño es simplificado	✓	✓	✓
Ciclos de iteración cortos menores a dos semanas	✓		
Cambios en proceso desarrollo	✓	✓	✓

Elaborado por: Autor

De acuerdo con el cuadro comparativo podemos ver que la Metodología XP es la más adecuada para el desarrollo de este proyecto ya que provee un sin número de ventajas que más adelante se indicarán.

1.4.4. Descripción de la metodología seleccionada.

Toda metodología debe ser adaptada al contexto del proyecto como son: recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Tomando en cuenta que no existe una metodología general que sea aplique como única solución a cualquier desarrollo. Sin embargo, la Programación Extrema proporciona dos ventajas adicionales en comparación con MOBILE-D y SCRUM para la realización del proyecto ya que ayudan al ahorro de tiempo y recursos.

- Permite la entrega de productos usables con mayor rapidez lo que permite la entrega de versiones continuas al usuario.
- Capacidad de respuesta ante cambios permitiendo la adaptación a nuevos requisitos de la empresa u organización.

Por estas y otras ventajas, la metodología que más se adecua a este proyecto de desarrollo, misma que pertenece a las “Metodologías Ágiles de Desarrollo” es la XP.

La estructura de desarrollo de la metodología XP se visualiza en el siguiente diagrama de la **Figura. 10**, con la cual se muestra que se hace en cada fase.

XP Aplicado

▶ Metodología XP(Extreme Programming)

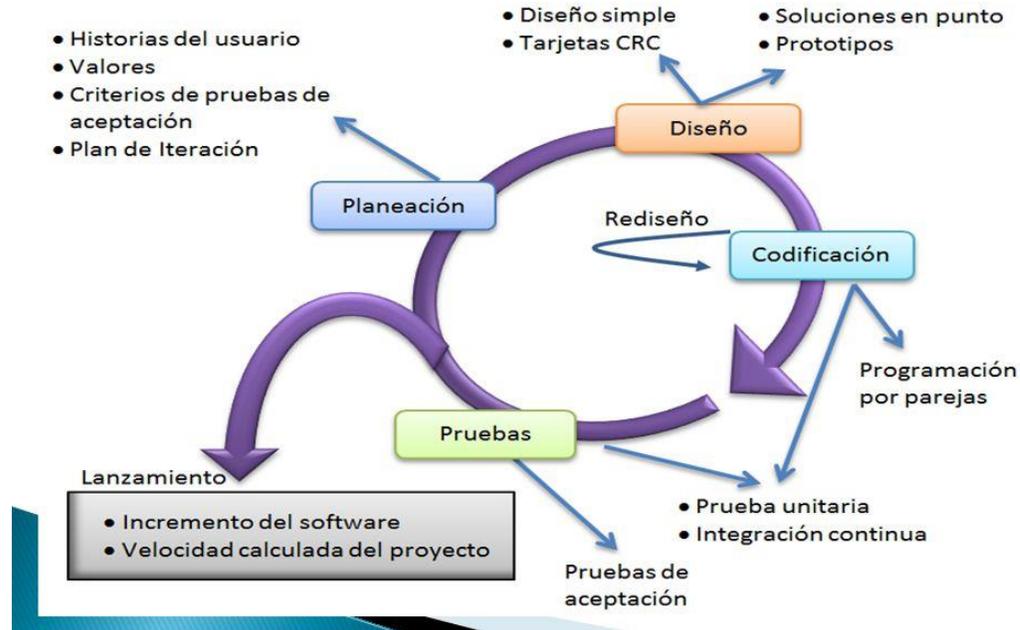


Figura 10. Diagrama de uso de la metodología XP
Fuente: (David Ramirez Torres, 2017)
Elaborado por: tomado de (David Ramirez Torres, 2017)

1.5. Información general de audiometría.

La información que a continuación se describe es fundamental para análisis y desarrollo de la aplicación de audiometría de tonos.

1.5.1. Generalidades de la audición humana.

La audición como un sistema sensitivo permite la percepción de la información que proporciona el medio, adjuntando características como tono, intensidad y timbre, las cuales ayudan a la persona a adaptarse a su entorno. Estas características son: la frecuencia, amplitud y composición armónica o forma de onda. El espectro de frecuencias audibles para el oído humano se encuentra entre los 20 Hertz hasta 20.000 Hertz.

Umbral de audición es el nivel de intensidad mínimo de sonido que es capaz de escuchar el oído humano. Su valor es referenciado en los 0 dB NPS (Nivel de presión sonora) o 20 micro pascales a 1KHz. El umbral de dolor es el nivel de intensidad en el que un sonido produce

sensación de dolor en el oído, su valor medio se sitúa entre los 100-120 dB (NPS) o 20 pascuales.

La sensación sonora de intensidad se agudiza para sonidos débiles y se disminuye para sonidos más fuertes, esto significa que la audición humana no es lineal, sino logarítmica. La sensación auditiva puede ser permanente o temporal. (Sebastián, 1997)

1.5.2. Percepción del sonido.

El mecanismo de la audición humana es un transductor electro mecano acústico muy sensible, y corresponde a ondas sonoras de una gran diversidad de frecuencias, intensidades y formas de onda y estas transforman las fluctuaciones de presión acústica en impulsos nerviosos.

Se ha determinado que la sensación auditiva depende de muchos factores y no es posible relacionarla con el cero de intensidad física, por ello se parte de un cero relativo establecido para medir la audición humana.

Con las curvas de igual sonoridad se establece el cero audible y se deducen las curvas isofónicas, partiendo de varias unidades como el fon y el son.

El fon es una unidad de medida psicofisiológica logarítmica y adimensional que se usa para indicar la sonoridad con la que se percibe un determinado sonido, el fon está referenciado a un sonido senoidal de 1KHz con un nivel de presión sonora de 0 dB NPS. A diferencia del fon, el son es capaz de establecer una relación real de sonoridad de sonidos diferentes y se define como la sonoridad de un sonido senoidal de 1KHz con un nivel de presión sonora de 40 dB NPSL. (Seto & Sarmiento Rodríguez, 1973)

Las unidades de altura tonal del oído a diferentes frecuencias fueron determinadas de manera experimental por Fletcher y Munson (Sebastián, 1997).

Las curvas isofónicas de igual sonoridad calculan la relación entre la frecuencia y el nivel de presión sonora (unidades decibeles) de dos sonidos.

En estas curvas se observa como el oído humano es más sensible a las frecuencias medias y media-altas que a las graves. Además, también se puede observar cómo a medida que aumenta el nivel de presión sonora, las curvas se van haciendo más planas, esto debido a que el tímpano se tensa para poder protegerse de los altos volúmenes de intensidad.

Las gráficas están calibradas de manera que en 1KHz coincidan los dB NPS (unidades objetivas) y los dB subjetivos o más conocidos como Fons.

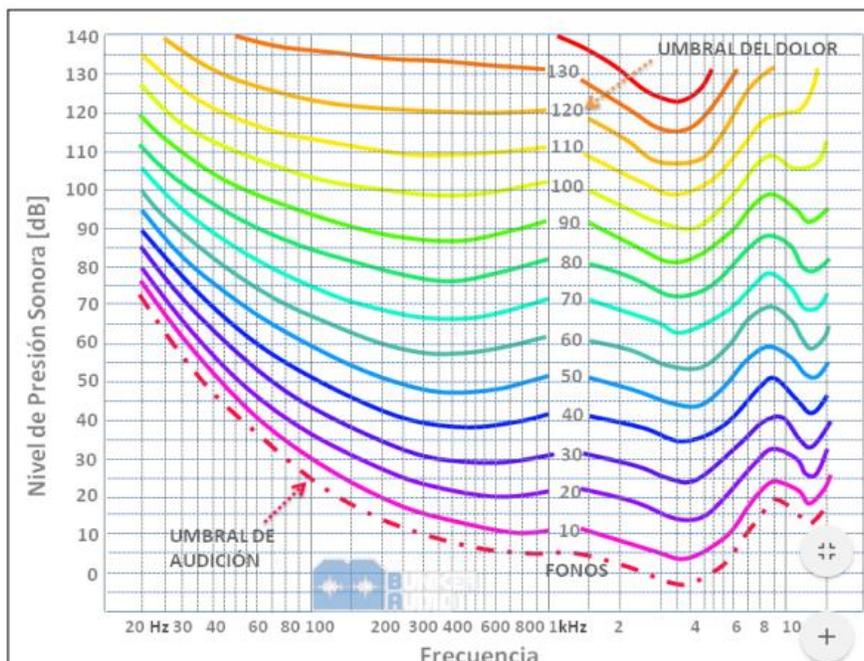


Figura 11. Curvas de igual sonoridad obtenidas por Fletcher y Munson.
Fuente:(Ronald Furet H., 2013)
Elaborado por: (Ronald Furet H., 2013)

1.5.3. Anatomía y fisiología del oído humano.

El oído es un órgano responsable de la audición y el equilibrio, está dividido en tres partes: oído externo, oído medio y oído interno. El proceso de audición comienza cuando las ondas sonoras penetran por el conducto del oído externo, esto ayuda a que distintas ondas con distintas frecuencias puedan ser comprendidas por el cerebro humano y su disminución en decibelios sean identificadas por el oído y a su vez por el cerebro, esto permite que la audiometría tonal pueda identificar los niveles de pérdida de audición y poder realizar diagnósticos preventivos.

Entender estos órganos ayudaran a comprender como se percibe el sonido y cuáles serían los problemas si se realizara el examen de audiometría tonal con el conducto auditivo obstruido.

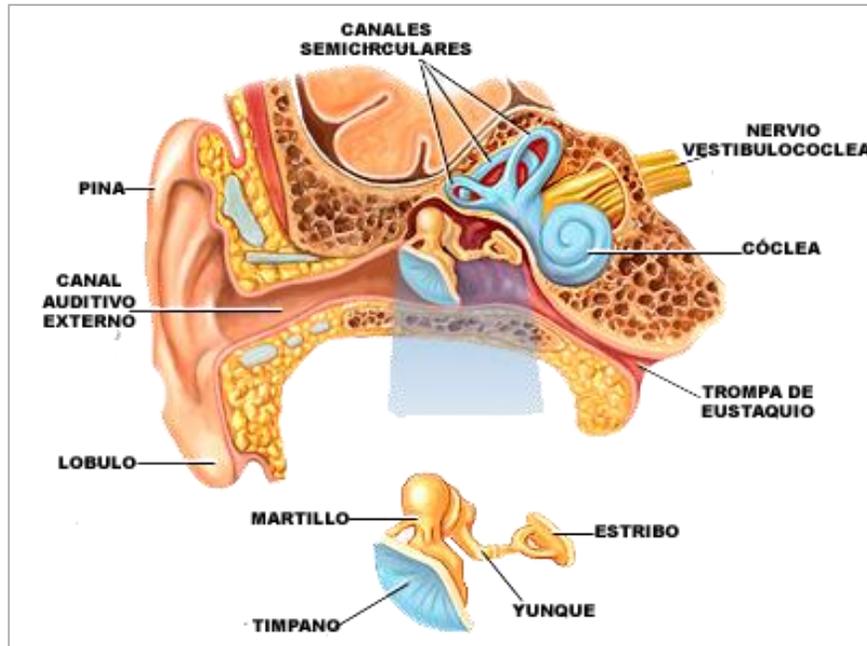


Figura 12. Anatomía y composición del oído humano.

Fuente: (Henry Vandyke Carter, 1853)

Elaborado por: (Henry Vandyke Carter, 1853)

1.5.3.1. **Oído externo.**

El oído externo es un aparato de transmisión, se encuentra en posición lateral al tímpano y está conformado por la oreja o pabellón auricular y el conducto auditivo.

El pabellón del oído recibe, concentra y amplifica la señal de los sonidos, dirigiendo las ondas sonoras al conducto auditivo.

El conducto auditivo tiene una frecuencia de resonancia que se encuentra alrededor de los 3 kHz, el cual hace que se duplique el nivel de presión sonora (NPS) de las vibraciones que llegan al conducto.

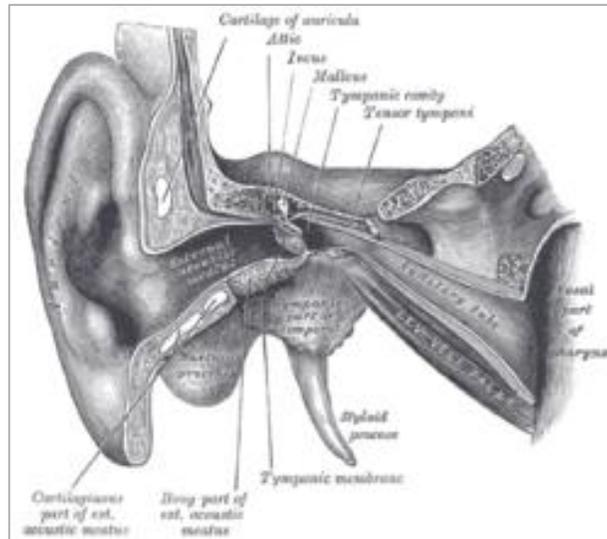


Figura 13. Estructura básica del oído externo.

Fuente: (Henry Vandyke Carter, 1853)

Elaborado por: (Henry Vandyke Carter, 1853)

1.5.3.2. Oído medio.

El oído medio está en comunicación directa con la nariz y garganta, a través de la trompa de Eustaquio, que permite la entrada y salida de aire para equilibrar las diferencias de presión entre este y el exterior.

En el oído medio existe una cadena de tres huesecillos u osículos, denominados martillo, yunque y estribo, que se encuentran unidos entre sí asegurando una conexión entre el tímpano y la ventana oval, permitiendo la transmisión de las vibraciones del tímpano hacia el oído interno.

El martillo que está sujeto al tímpano y recoge las vibraciones sonoras, que luego son llevadas al oído interno por el yunque y el estribo. De ese modo las ondas sonoras llegan en forma de energía mecánica amplificada (Fred Warshofsky, 1977).

La fuerza mecánica amplificada que se transmite del oído medio al interno a través de los osículos se transforma en presión hidráulica que trasmite movimiento al conducto coclear y al órgano de Corti.



Figura 14. Partes principales del oído medio.

Fuente:(Bruce Blaus, n.d.)

Elaborado por: (Blaus, n.d.)

1.5.3.3. *Oído interno.*

El oído interno representa el fin del procesamiento mecánico del sonido, en él se llevan a cabo tres funciones principales: el filtraje de la señal sonora, transducción y la generación de los impulsos nerviosos.

Dentro del oído interno, que se encuentra lleno de líquido, la fuerza que el estribo hace sobre la ventana oval del caracol se convierte en ondas de presión hidráulicas, las cuales recorren rápidamente los conductos vestibular y timpánico, así como el coclear. Al pasar sobre la membrana basilar se crean ondulaciones que estimulan al órgano de Corti (Fred Warshofsky, 1977).

El oído interno es el que convierte el estímulo físico al neurosensorial, está formado por el órgano de Corti con células ciliadas uniformemente distribuidas a lo largo de la membrana basilar. La selectividad tonal se debe a que estas células actúan dependiendo de la posición de la zona de excitación de la membrana basilar.



Figura 15. Composición del oído interno.

Fuente:(Bruce Blaus, n.d.)

Elaborado por: (Bruce Blaus, n.d.)

1.5.4. Causas más comunes que provocan la pérdida auditiva.

La pérdida auditiva se puede dar por el daño de:

OÍDO EXTERNO: normalmente se deben a acumulaciones excesivas de cerumen e infecciones del canal auditivo.

OÍDO MEDIO: los problemas más comunes se deben a la inflamación, la existencia de fluido detrás del tímpano, las perforaciones del tímpano y la otosclerosis (entumecimiento en los huesos del oído medio).

OÍDO INTERNO: la mayoría de los problemas de audición tienen que ver en esta parte del oído, la causa más habitual es el envejecimiento natural, el exponerse a ruidos intensos como por ejemplo al escuchar música durante largo tiempo “La excesiva exposición del oído a estos aparatos y los altos niveles de volumen a los que se escuchan puede provocar hipoacusia (Es la pérdida auditiva de superficial a moderada en uno o ambos oídos; esta pérdida es menor de 70 decibeles.) o, lo que es lo mismo, la pérdida permanente de la sensibilidad auditiva” (Estudiantes: Isidro Pulido y David Rivas & Asesores: Servín Rivas y Rabadán Malda, 2010) (p.8), “las audiometrías, encuestas y condiciones de uso del reproductor se puede relacionar

que hay una pérdida de audición en la cual los reproductores portátiles de audio pueden influir a dicho problema, este es considerado que los niveles de presión máxima superan los 85 dB y los tiempos de exposición con el oído humano sobrepasan los 60 minutos ”(Estudiantes: Isidro Pulido y David Rivas & Asesores: Servín Rivas y Rabadán Malda, 2010) (p.71)

También pueden afectar negativamente a la capacidad auditiva de las personas tomar algún tipo de medicamentos o sufrir fracturas en la cabeza. Estas influencias dañan las finas células ciliadas y afectan a la transmisión de señales a los nervios auditivos.

La pérdida auditiva provocada por algún daño en el oído externo o medio se denomina pérdida auditiva conductiva, cuando los daños se ubican en el oído interno, se denomina pérdida auditiva neurosensorial y si se producen ambos tipos se denomina pérdida auditiva mixta.

La audiometría tonal puede detectar la hipoacusia a una edad temprana y también se puede utilizar cuando se presente una dificultad para oír por cualquier otra causa (Norton, 2010).

1.5.5. Definición de audiometría.

La audiometría es la medición de la agudeza auditiva en relación con las diferentes frecuencias de sonido, la prueba se realiza con un aparato de alta tecnología que consiste en un generador de distintas frecuencias a diferentes niveles de presión sonora, este instrumento emite tonos puros de intensidad variable, sonidos que el ser humano no está acostumbrado a escuchar, ya que no existen como tal en el entorno natural.

La audiometría tonal permite definir el tipo de hipoacusia (es la disminución de la sensibilidad auditiva) que presenta el paciente, así como cuantificar la pérdida auditiva para las distintas frecuencias exploradas, también permite detectar los daños ocasionados por la exposición al ruido.

Al finalizar la audiometría se obtiene un gráfico denominado audiograma, el cual ayuda a obtener una mejor interpretación del examen.

Procedimiento básico para detectar el umbral de audición.

En 1959 Carhart y Jerger investigaron distintos métodos para obtener el umbral de audición, ellos recomendaron el método ascendente-descendente como el más adecuado. Este procedimiento de obtención del umbral de audición, descrito y aplicado por Kaplan y Mercher, consiste en aplicar en los transductores, un tono de un nivel de presión sonora que el paciente

pueda oír claramente a una frecuencia determinada (donde el paciente oprime un botón o levanta la mano) e ir disminuyendo de 10 dB en 10 dB su nivel hasta que el paciente deje de oírlo, momento en el cual se debe aumentar de 5 dB en 5 dB hasta que el paciente detecte el nivel del tono nuevo, al realizar la prueba en repetidas ocasiones, se obtiene un promedio de los datos. (Kaplan,1998).

Esta técnica permite obtener el umbral de audición como aquel nivel mínimo que el paciente detecta en un 50 por ciento de las veces.

1.5.6. Características de la audiometría.

La audiometría es un examen que tiene por objeto cifrar las alteraciones de la audición en relación con los estímulos acústicos, resultados que se muestran en un audiograma.

Los procedimientos específicos pueden variar, pero generalmente implican el estudio de un oído a la vez con el fin de evaluar la capacidad de escuchar susurros, palabras habladas o el sonido del tictac de un reloj. Si es que se obtiene una hipoacusia en la vía aérea se debe examinar la vía ósea, de lo contrario no es necesario.

La audiometría proporciona una medición más precisa de la audición, para evaluar la conducción aérea, se colocan unos audífonos que van conectados al audiómetro, los tonos puros de intensidad controlada son transmitidos al oído uno solo a la vez, se solicita levantar la mano, presionar un botón o indicar de otro modo cuando se escuche un sonido (Kileny,2010).

La duración de este examen varía según el tipo de audiometría que desee, la evaluación inicial puede tomar de 5 a 10 minutos, mientras que una audiometría detallada puede tomar casi una hora, no se necesita preparación especial para este examen y no ocasiona ningún tipo de molestia.

1.5.7. Representación e interpretación del audiograma.

En un audiograma se representan las frecuencias de 125 a 4000 Hz por intervalos en bandas de octavas (abscisas) y en sentido ascendente están ubicadas las pérdidas en decibelios de nivel de audición (ordenadas) en relación con el eje 0, el que representa el umbral normal para las vías óseas y aéreas.

Cada señal está representada por un pequeño círculo para el oído derecho y por una pequeña cruz para el oído izquierdo, así pueden inscribirse ambos oídos en el mismo gráfico, el derecho en color rojo y el izquierdo en color azul, como se muestra en la **Figura. 16** y **17**.

TIPO DE ENSAYO	OÍDO DERECHO	OÍDO IZQUIERDO
•Conducción aérea	○	×
•Ausencia de respuesta	⊗	⊗

Figura 16. Señales que se usan para diagramas audiométrico
 Fuente: (Cortés Aguilera, Enciso Higuera, & Reyes González, 2012)
 Elaborado por: Autor

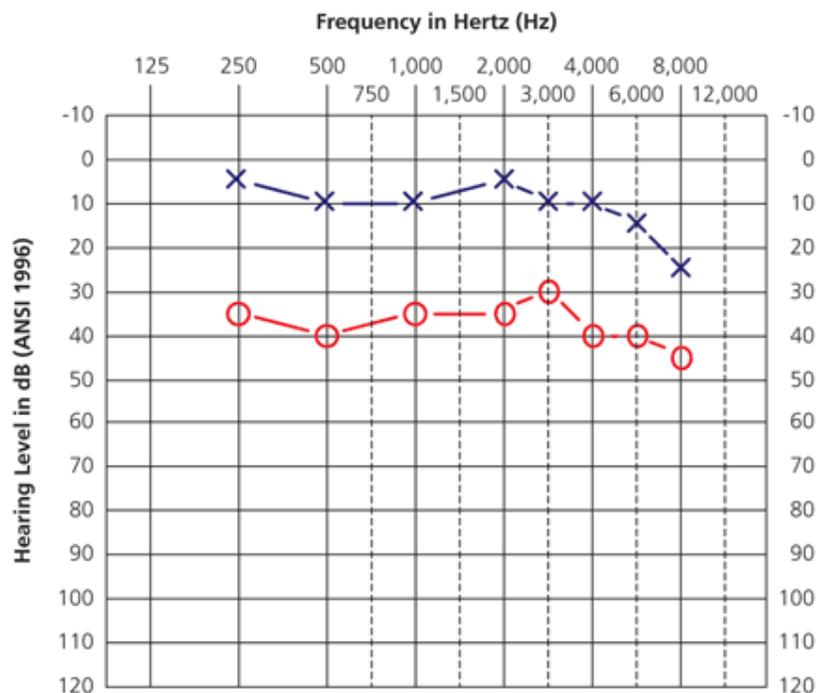


Figura 17. Diagrama audiométrico con señales
 Fuente: (Gerardo Armando Esparza Mendez, 2013)
 Elaborado por: Autor

Umbrales entre -10 y +20 dB HL se consideran en el rango normal, mientras que los umbrales que se encuentran por encima de los +20 dB HL se consideran en un diagnóstico de leve, moderado, severa o profunda, como se muestra en la **Figura. 18**.

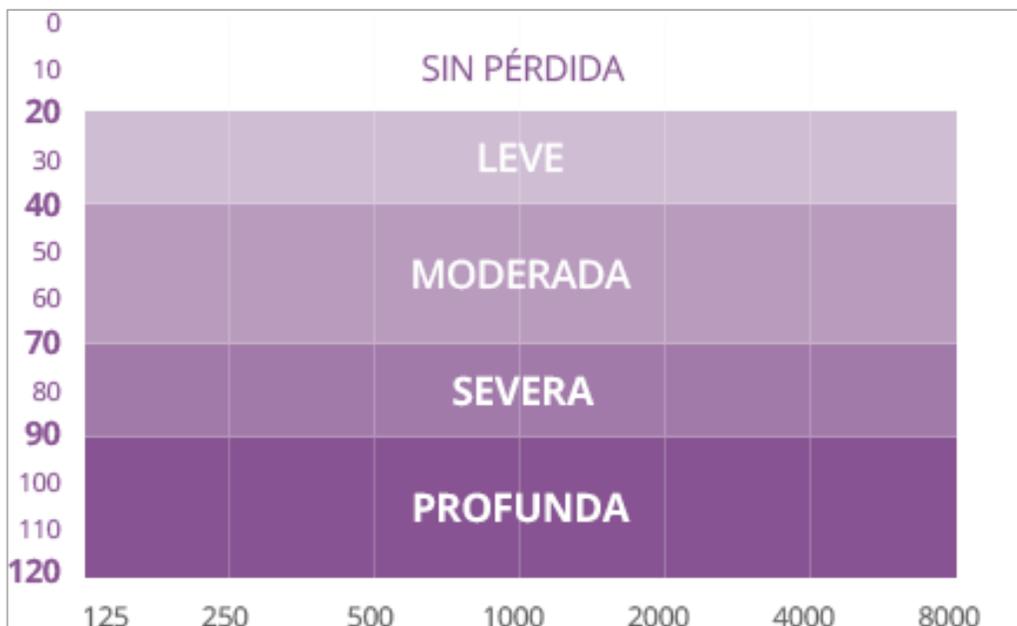


Figura 18. Audiograma de niveles y audiograma de banana auditiva

Fuente:(Taladro, 2012)

Elaborado por: (Taladro, 2012)

1.5.8. Transductores utilizados en audiometrías vía aérea.

Para la realización de las audiometrías de vía aérea, se pueden utilizar los siguientes transductores:

- Auriculares supra-aurales: Son los transductores más utilizados que permiten realizar audiometrías hasta 8 KHz. Este tipo de altavoces cubren la mayor parte del pabellón auditivo, su sensibilidad en bajos y agudos es más profunda y tienen un material aislante de ruido externo.
- Auriculares intra-aurales: se puede disfrutar de un sonido claro y detallado, gracias a su sistema de altavoces dinámicos de alto rendimiento, cuentan con varios juegos de adaptadores de oído (S / M / L tamaños) permitiendo un ajuste más personalizado para la realización de la prueba, así como una excelente atenuación pasiva del ruido ambiente.

Para seleccionar los auriculares que se deben utilizar en las audiometrías, se deben de tomar en cuenta las siguientes características.

Tipo de audífono	Intraural
Color	Opcional
Impedancia	16 Omhs
Respuesta en frecuencia	19-20500 Hz
Nivel de presión sonora (NPS)	120 dB (NPS)
Entrada	3,5 mm ángulo
Principio del transductor	Dinámico
Peso	12 gr

Figura 19. Especificaciones técnicas de auriculares intra-aurales

Fuente:(Sebastián, 1997)

Elaborado por: Autor

1.5.9. Decibeles NPS vs Decibeles HL.

La intensidad es la cantidad de energía que atraviesa una superficie, sus unidades son vatios por metro cuadrado (w/m^2). El nivel de intensidad se mide en decibeles (dB NPS nivel de presión sonora) y la pérdida auditiva se suele medir en dB HL (decibeles de nivel de audición). Una persona cuyos umbrales de audición tengan un valor mayor a 20 dB HL del valor medio, tiene una pérdida auditiva.

En el nivel de intensidad dB NPS (nivel de presión sonora), la presión sonora de referencia en la fórmula logarítmica es de 20 micro pascales, si se necesitarán usar los dB NPS para representar la audición de un paciente con sordera, se tendría una línea curva distinta, pero sería difícil de entenderla, se puede decir que todos los puntos de esa curva son el valor 0 dB NPS, valor donde empieza a oír el humano y por tanto son valores de referencia para la escala de medición de la capacidad auditiva. A partir de esta gráfica se puede construir otra donde las líneas que antes eran curvas se hagan completamente planas para todas las frecuencias y la escala de medición será ahora la denominada en decibeles de nivel de audición (dB HL) siendo el valor de esta línea el 0 dB HL.

Con los dB HL se obtiene una gráfica que sirve para medir directamente la pérdida auditiva de una persona, lo cual es mucho más práctico que si se midiera en parámetros físicos de pérdida absoluta.

Cabe mencionar que el 0 dB HL no significa que no exista presión acústica, ya que el valor de 0 dB HL se ha obtenido estadísticamente y con diferentes presiones sonoras para las distintas frecuencias. Se pueden dar valores desde -5 o -10 dB HL, que indica que hay personas que empiezan a oír desde niveles inferiores de presión sonora a los de la media estadística. La siguiente tabla muestra la relación entre dB NPS y dB HL para las frecuencias estudiadas en las audiometrías.

Frecuencia	dB NPS	dB HL
125 Hz	47,5	0
250 HZ	26,5	0
500 Hz	13,5	0
1000 Hz	7,5	0
2000 Hz	11	0
4000 Hz	10,5	0
8000 Hz	13	0

Figura 20. Equivalencia entre dB NPS vs dB HL
 Fuente:(Sebastián, 1997)
 Elaborado por: Autor

1.5.10. Pruebas de audición en dispositivos móviles.

1.5.10.1. *Uso de dispositivos móviles en audiometría.*

La popularización de dispositivos móviles con reproducción de audio ha crecido tanto en los últimos años, que ahora en el mismo paquete del celular vienen auriculares de buena calidad. Los dispositivos Android cuentan con un nivel de audio de referencia determinado por su software, para poder reproducir tonos en excelente calidad y a su vez poder modificar sus decibelios.

Debido al rápido crecimiento del mercado en diversidad de dispositivos móviles, es apropiado proponer un método que determine automáticamente el nivel de sonido de referencia, sin tener que calibrar cada dispositivo en un laboratorio. El nivel de sonido de referencia predefinido puede realizarse mediante una calibración biológica y hasta que se obtenga un nivel de referencia confiable, se pedirá al usuario que realice la calibración biológica antes de la prueba. Se puede suponer que las pruebas realizadas sobre la base del nivel de sonido de referencia predefinido serán más precisas que las realizadas en una única calibración biológica. Al mismo tiempo, la escalabilidad de la solución propuesta con respecto al número

de modelos de dispositivo es mayor que la calibración de laboratorio, y, por lo tanto, se espera que la disponibilidad de la prueba sea también más alta. (Masalski, Kipiński, Grysiński, & Kręcicki, 2016)

1.5.10.2. ***Evaluación del umbral auditivo de una persona.***

La evaluación del umbral auditivo de conducción de aire se puede realizar en dispositivos electrónicos comunes como computadoras personales, computadoras portátiles, tabletas o teléfonos móviles [1 - 11]. El examen no solo se aplicará en las pruebas de detección, sino que también puede ser útil para el autocontrol en los siguientes trastornos auditivos: pérdida auditiva fluctuante, enfermedad de Menier, tinnitus, pérdida auditiva neurosensorial repentina, pérdida de la audición relacionada con la edad o durante el tratamiento ototóxico [1, 2]. Además, puede ser beneficioso para la evaluación preliminar de pacientes con quejas otológicas [3], al ahorrar el tiempo necesario para exámenes más sofisticados o para pacientes que requieren más atención médica [12] Las pruebas de audición basadas en teléfonos móviles también pueden complementar otros métodos de telesalud en otorrinolaringología, como el ajuste remoto del audífono [13] o la otoendoscopia basada en teléfonos móviles [14]. Los exámenes de audición realizados en dispositivos electrónicos comunes pueden resultar particularmente útiles en estas partes del mundo donde el acceso a los profesionales de la salud auditiva es muy limitado. (Masalski, Kipiński, Grysiński, & Kręcicki, 2016)

1.5.10.3. ***Necesidad de calibración de los dispositivos móviles para audiometría.***

Las pruebas de audición llevadas a cabo en casa por medio de dispositivos móviles requieren una calibración previa del nivel de sonido de referencia. Los dispositivos móviles con auriculares integrados crean la posibilidad de aplicar el nivel predefinido para un modelo en particular como alternativa a la calibración de cada dispositivo por separado. (Masalski et al., 2016).

CAPITULO II.
ESTUDIO DE NECESIDADES

2.1. Requerimientos del sistema.

Para poder obtener los requerimientos del sistema se indago sobre lo que recomienda la metodología de desarrollo XP, la cual recomienda usar técnicas de recolección de requerimientos. Se uso la técnica de cuestionario por que puedo obtener de un grupo grande de personas datos confiables, también se uso la técnica de la entrevista que es de gran utilidad para la obtención de información mediante la comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada, de estas dos técnicas se obtuvo los requerimientos de un grupo de personas y de un especialista en audiometría respectivamente.

A continuación, los requerimientos adquiridos del especialista en audiometría y del grupo de personas.

2.1.1. Médico.

Para iniciar con la obtención de requerimientos de un especialista en audiometría se optó por ir a un instituto especializado de audición y lenguaje dentro del país, además de ser uno de los dueños del centro, trabaja en la Facultad de Ciencias de la Discapacidad de la Universidad Central del Ecuador y tiene un doctorado en especialidad en problemas de comunicación lingüística (audiólogo foniatra). Ver **Anexo A**.

Las historias de usuario obtenidos del especialista fueron las siguientes:

Historial de Usuario	
Numero: 1	Nombre: Información relevante
Usuario: Especialista audiométrico	
Modificación de historia número:	Iteración asignada: 1
Prioridad en negocio: Alta (ALTA/Media Baja)	Puntos estimados: 3
Riesgo en desarrollo: Alta (Alto/Medio/Bajo)	Puntos reales:
Descripción: Yo como especialista audiométrico, necesito que la aplicación permita a los usuarios visualizar información relevante sobre audiometría de tonos, de modo que puedan comprender fácilmente el funcionamiento de la aplicación y los resultados obtenidos.	
Observaciones: Para que el usuario no tenga ningún problema será recomendable que se muestre en el menú la opción de información donde se encontrará lo requerido.	

Historial de Usuario	
Numero: 2	Nombre: Pasos para realizar las pruebas audiometrías
Usuario: Especialista Audiometría	
Modificación de historia Numero:	Iteración asignada: 2
Prioridad en negocio: Alto (ALTA/Media Baja)	Puntos estimados: 10
Riesgo en desarrollo: Bajo (Alto/Medio/Bajo)	Puntos reales:
Descripción: Yo como especialista audiométrico, necesito que la aplicación cuente con pasos precisos para que el usuario pueda realizarse la prueba audiométrica y no cometa error que conlleve a resultados erróneos.	
Observaciones: Para que el usuario entienda como se debe realizar la prueba audiométrica será necesario colocar un cuadro de dialogo antes de iniciar la prueba donde se indique el procedimiento que deberá seguir para realizarse la prueba audiométrica.	

Historial de Usuario	
Numero: 3	Nombre: Frecuencias para realización de prueba audiométrica
Usuario: Especialista audiométrico	
Modificación de Historia Numero:	Iteración Asignada: 3
Prioridad en Negocio: Alta (ALTA/Media Baja)	Puntos estimados: 10
Riesgo en Desarrollo: Alto (Alto/Medio/Bajo)	Puntos Reales:
Descripción: Yo como especialista, necesito que las siguientes frecuencias sean predeterminadas, para tener un examen completo de los oídos al usuario. A continuación, las frecuencias óptimas para un examen normal: (250,500,1000,2000,4000,6000,8000) Hz. Además, cuente con un modo avanzado para especialistas, donde se pueda escoger las frecuencias a las cuales se requiera realizar la prueba y obtener de ellas una información más detallada. A continuación, las frecuencias avanzadas que requeriría un especialista: (125,250,500,750,1000,1500,2000,3000,4000,6000,8000,10000,12000,14000,16000) Hz.	
Observaciones: Para que el usuario o especialista que se encuentre usando la aplicación pueda realizarse la audiometría de tonos en las frecuencias que deseen se va a crear parámetros modificables donde se pueda escoger las frecuencias predeterminadas o las que el usuario pueda requerir.	

Historial de Usuario	
Numero: 4	Nombre: Pasos antes de realizarse una audiometría de tonos
Usuario: Especialista audiométrico	
Modificación de historia número:	Iteración asignada: 4
Prioridad en negocio: Medio (ALTA/Media Baja)	Puntos estimados: 6
Riesgo en desarrollo: Medio (Alto/Medio/Bajo)	Puntos reales:
Descripción:	

Yo como especialista requeriría que la aplicación cuente con información sobre qué se debe hacer antes de realizarse una prueba audiométrica:

- Tipo de auriculares que se debe usar.
- Lugares recomendados para realizarse la prueba o examen.
- Las horas del día óptimas para realizar la prueba.

Observaciones:

Esta información puede estar incluida en una actividad previa a la realización del examen audiométrico con el uso de imágenes e información relevante.

Historial de Usuario	
Numero: 5	Nombre: Datos pedidos por la aplicación
Usuario: Especialista audiométrico	
Modificación de historia Numero:	Iteración asignada: 5
Prioridad en negocio: Media (ALTA/Media Baja)	Puntos estimados: 10
Riesgo en desarrollo: Medio (Alto/Medio/Bajo)	Puntos reales:
Descripción: Yo como especialista requeriría que la aplicación pida información general al usuario sobre: <ul style="list-style-type: none"> • Nombres • Apellidos • Edad • Teléfono • Fecha de su última limpieza de oído que se realizó en un lugar especializado. 	
Observaciones: Para la fecha de la última limpieza de oídos en un lugar especializado, se puede realizar un botón switch donde el usuario activaría si anteriormente se hubiese realizado un examen y proseguiría a colocar la fecha.	

Historial de Usuario	
Numero: 6	Nombre: Guardar Pruebas
Usuario: Especialista audiométrico	
Modificación de historia número:	Iteración asignada: 6
Prioridad en negocio: Alta (ALTA/Media Baja)	Puntos estimados: 10
Riesgo en desarrollo: Medio (Alto/Medio/Bajo)	Puntos reales:
Descripción: Yo como especialista requeriría, que en la aplicación se pueda guardar las pruebas realizadas anteriormente por el usuario, para mostrarlas al especialista si lo requiere.	
Observaciones: Estas pruebas se las grabará en la memoria del celular dentro del paquete de la aplicación para evitar su eliminación accidentalmente.	

Historial de Usuario	
Numero: 7	Nombre: Creación de Página Web
Usuario: Especialista audiométrico	
Modificación de Historia Numero:	Iteración asignada: 7
Prioridad en Negocio: Baja (ALTA/Media Baja)	Puntos estimados: 9

Riesgo en Desarrollo: Bajo (Alto/Medio/Bajo)	Puntos reales:
Descripción: Yo como especialista requeriría que la aplicación cuente con una página web de publicidad en donde se tenga la información necesaria y su descarga.	
Observaciones: Esta página se la podría realizar con un programa como Joomla para poder realizarla rápidamente y no cueste tanto su elaboración.	

Planificación de iteración:

Tabla 3. Plan de iteraciones

	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5	Iteración 6	Iteración 7	Total, desarrollo
Horas semanales	48	8	24	42	8	20	48	150
Semana	1	2	3	4	5	6	7	7
Historias de Usuario (Velocidad del Proyecto)	9	9	9	9	9	8	5	

Elaborado por: Autor

Complemento caso de uso:

Tabla 4. Identificación de los actores.

Actores de la aplicación de audiometría tonal móvil	
Usuario	Usa la aplicación, Realizar la prueba audiométrica.

Elaborado por: Autor

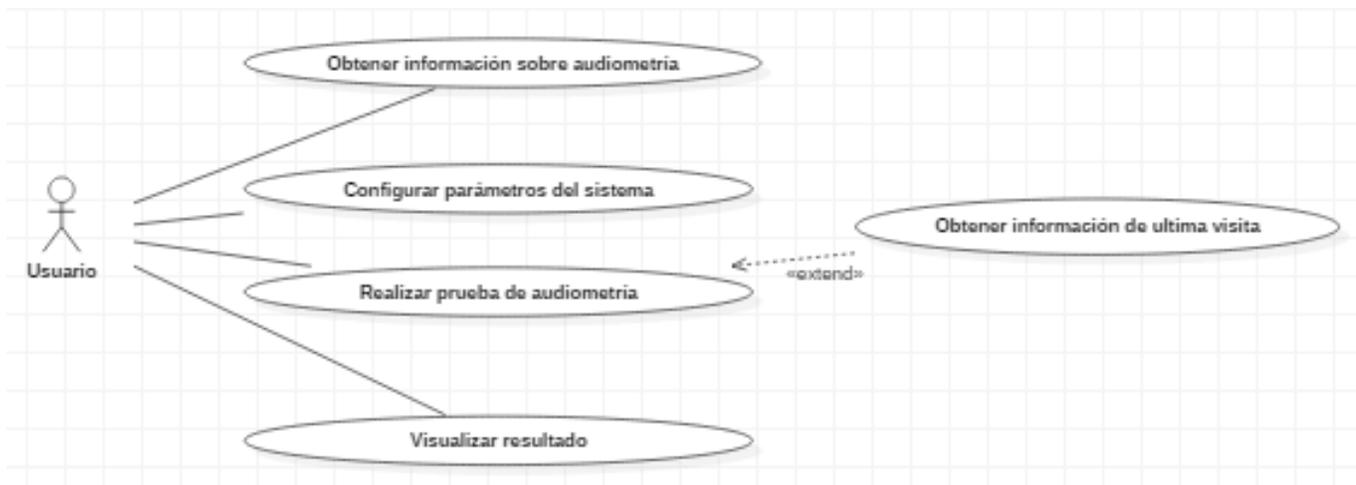


Diagrama 1. Caso de uso.
Elaborado por: Autor

Especificación de casos de uso:

Tabla 5. Especificación caso de uso N°1.

CASO DE USO NUMERO 1:		
Nombre	Obtener Información sobre audiometría	
Descripción	Se le muestra la información sobre audiometría de tonos y que es lo que hace esta aplicación y que resultados se espera.	
Actores	Usuario	
Precondición		
Flujo del sistema	Paso	Acción
	1	Este caso de uso comienza cuando el actor da clic en la aplicación
	2	El sistema despliega el menú de la aplicación.
	3	El usuario ingresa en la opción del menú "Información"
	4	El sistema despliega la ventana con información.
Post condición:		

Elaborado por: Autor

Tabla 6. Especificación caso de uso N°2

CASO DE USO NUMERO 2:		
Nombre	Configurar parámetros del Sistema	
Descripción	Colocar parámetros para que la aplicación funcione como requiera el usuario.	
Actores	Usuario	
Precondición	1.	Colocar los parámetros iniciales ya definidos o una configuración previa
Flujo del sistema	Paso	Acción
	1	Este caso de uso comienza cuando el actor da clic en la aplicación

	2	El sistema despliega el menú de la aplicación.
	3	El usuario ingresa en la opción del menú "Configuración"
	4	El sistema despliega la ventana de configuración y muestra opciones (cambiar frecuencias de audiometría, si requiere que se le pida datos al usuario, cambiar parámetros de calibración).
	5	El usuario realiza los cambios en la configuración
	6	El sistema valida los cambios.
Post condición:	0	El usuario regresar al menú del sistema e iniciar la prueba de audiometría para que se vean reflejados los cambios que se realizaron en configuración.

Elaborado por: Autor

Tabla 7. Especificación caso de uso N°3

CASO DE USO NUMERO 3:		
Nombre	Realizar prueba de audiometría	
Descripción	Identificar al usuario para el uso de la aplicación.	
Actores	Usuario	
Precondición	1.	Se verifico la presencia de auriculares
Flujo del sistema	Paso	Acción
	1	Este caso de uso comienza cuando el actor da clic en la aplicación
	2	El sistema despliega el menú de la aplicación.
	3	El usuario ingresa a "iniciar prueba de audición".
	4	El sistema despliega la ventana

	5	El usuario ingresa los datos pedidos y coloca siguiente
	6	El sistema despliega la ventana de recomendaciones.
	7	El usuario lee y acata las recomendaciones
	8	El sistema verifica que los auriculares están conectados para seguir con el despliegue de la ventana de la prueba audiométrica
	9	El usuario inicia el examen.
Post condición:	o	Si el usuario no se coloca los audífonos no se inicia la ventana de prueba de audiometría de tonos.
Elaborado por: Autor		

Tabla 8. Especificación caso de uso N°4.

CASO DE USO NUMERO 4:		
Nombre	Visualizar resultado	
Descripción	Se muestran el resultado previo haberse realizado la prueba audiométrica.	
Actores	Usuario	
Precondición	1.	Haberse realizado la prueba audición
Flujo del sistema	Paso	Acción
	1	El usuario termina la interacción en el caso de uso n°3
Post condición:	2	El sistema procesara la prueba realizada y mostrara el resultado
	3	El usuario visualizara su resultado de la prueba
	o	Realizarse la prueba audiometría como se lo explica la aplicación. Caso contrario el usuario no podrá visualizar los resultados esperados. Y existiría un error en el examen.

2.1.2. Usuario.

Para la obtención de los requerimientos de usuario, se usó la técnica de la encuesta con un cuestionario de preguntas de opción múltiple. Ver **Anexo B**.

Las preguntas fueron realizadas con la herramienta Google Forms a 44 personas y se obtuvieron los siguientes resultados:

Tabla 9. Resultados de la encuesta

Pregunta	Resultado
¿Cuál es su sexo?	El 38,6% son mujeres y el resto son hombres
¿Cuál es su edad?	Rango de edad desde 23 a 55 años
¿Cuenta usted con un teléfono celular inteligente "¿Smartphone" o una Tablet, o ambos?	El 72,7% de los encuestados tiene un smartphone, el 25% tiene un smartphone y tablet y el resto no tiene ninguno
¿Qué tipo sistema operativo tiene su Smartphone o Tablet?	El 84,4% de los encuestados tiene un sistema operativo Android y el resto tiene un sistema operativo ISO
El tipo de auriculares utiliza	El 45,5% de los usuarios usan auriculares intra-aurales, el 27,3% utilizan auriculares supra-aurales y el resto utiliza auriculares circum-aurales
¿Cuáles de las siguientes características considera usted que debe tener una aplicación de audiometría tonal móvil que pretende igualar la funcionalidad de un aparato audiométrico de tonos?	Los usuarios desean las siguientes características en orden de prioridad satisfaga el problema, que sea de fácil acceso, sea simple, no precise de internet, sea de fácil uso y tenga seguridad en general
¿A través de qué medios le gustaría encontrar información sobre este producto?	El 62,5% de los encuestados le gustaría recibir información de la aplicación mediante página web, el 25% mediante redes sociales y el resto mediante tienda de aplicaciones.
¿Cuál sería su criterio para comprar una aplicación que realice una prueba auditiva?	El 71,9% de los encuestados compraría la aplicación de acuerdo con el historial de éxitos y fracasos que ha tenido, el 18,8% de acuerdo con la validación que se le ha dado y el resto de acuerdo por quien esta validada la aplicación.
¿Preferiría que sea una aplicación de pago o gratuita?	El 71,9% de los encuestados prefiere que la aplicación sea de pago

¿En qué valor compraría esta aplicación?	El 44% de los encuestados desea que la aplicación cueste 1 dólar, el 28% \$1,50, el 20% \$ y el resto \$3.
¿De dónde acostumbra a descargar sus aplicaciones?	El 84% de los encuestados descargan sus aplicaciones desde Google Play y el resto desde App Store
Este producto es realizado como una tesis para tercer nivel ¿Eso lo hace más, o menos interesante para usted?	El 80% de los encuestados les parece interesante que la aplicación sea un trabajo de tesis de tercer nivel, el resto es indiferente.
¿De realizarse esta aplicación estaría interesado(a) en adquirirla?	El 88,9% de los encuestados están interesados en adquirir esta aplicación y el resto no

Elaborado por: Autor

De acuerdo con los resultados de la encuesta se concluye que la aplicación de audiometría:

- Debe ser compatible para Smartphone con sistema operativo Android
- Se debe utilizar auriculares tipo intra-aurales
- La aplicación debe satisfacer el problema, ser de fácil acceso y no precise de internet
- Debe tener una página web donde haya toda la información sobre la aplicación
- Mostrar los resultados sobre los éxitos y fracasos de la aplicación
- Debe tener un costo para brindar el servicio de soporte técnico
- La aplicación debe costar \$1
- Debe estar disponible en Google Play
- La mayoría de las personas están interesadas en adquirir la aplicación.

2.1.3. Dispositivo móvil (Smartphone o Tablet).

Luego de analizar los requerimientos de los usuarios y del especialista en audiometría, se obtienen los siguientes requerimientos mínimos que el dispositivo móvil debe tener para ejecutar la aplicación, a continuación, los detallo en la siguiente tabla:

Tabla 10. Requerimientos mínimos del dispositivo

REQUERIMIENTO	CARACTERÍSTICA
Versión de Android	4.4.x KitKat o superior
Procesador	1.2 GHz Dual Core
Memoria RAM	1 GB

Almacenamiento	50 MB disponibles
Pantalla	800 x 480 pixeles, 4 pulgadas
Auriculares	De tipo Intra-aurales de buena calidad de preferencia, los que vienen los dispositivos.

Elaborado por: Autor

Se ha considerado la versión de Android 4.4x KitKat como referencia, para que, desde esta versión, hasta las nuevas versiones del Sistema Operativo tengan compatibilidad y cuenten con un porcentaje de relevancia del 93.5% de dispositivos que podrán instalar la aplicación, desde esta versión, las bibliotecas externas que se usaron para dar soporte a la aplicación cuentan con las funciones requeridas para el correcto funcionamiento de la aplicación. Los datos porcentuales de las versiones del S.O Android, se obtuvieron de la paginas de desarrolladores de Android. En la **Tabla. 11** se muestran los porcentajes con respecto a uso de las versiones Android en dispositivos activos, estos datos fueron recopilados durante un período de 7 días hasta el 8 de enero del 2018.

Tabla 11. Versión de Android y su porcentaje de dispositivos activos

Versión	Code Name	API	Distribución
2.3.3 - 2.3.7	Gingerbread	10	0,40%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0,50%
4.1.x	Jelly Bean	16	1,90%
4.2.x		17	2,90%
4.3		18	0,80%
4.4	KitKat	19	12,80%
5.0	Lollipop	21	5,70%
5.1		22	19,40%
6.0	Marshmallow	23	28,60%
7.0	Nougat	24	21,10%
7.1		25	5,20%
8.0	Oreo	26	0,50%
8.1		27	0,20%

Elaborado por: autor

CAPITULO III.
DISEÑO

3.1. Arquitectura.

La aplicación no requiere de acceso a internet, ni a consultas de datos externos, sin embargo, maneja almacenamiento con archivos locales donde guarda información de los análisis desarrollados, su funcionalidad depende de las utilidades de los dispositivos móviles como: salida de audio para auriculares, almacenamiento y procesamiento de información.

La aplicación se desarrolló para el sistema operativo Android, debido a que Android es el sistema operativo más usado en el mundo teniendo 37,93% de usuarios frente a otras plataformas. Como se lo demuestra a continuación en la **Figura.22**.

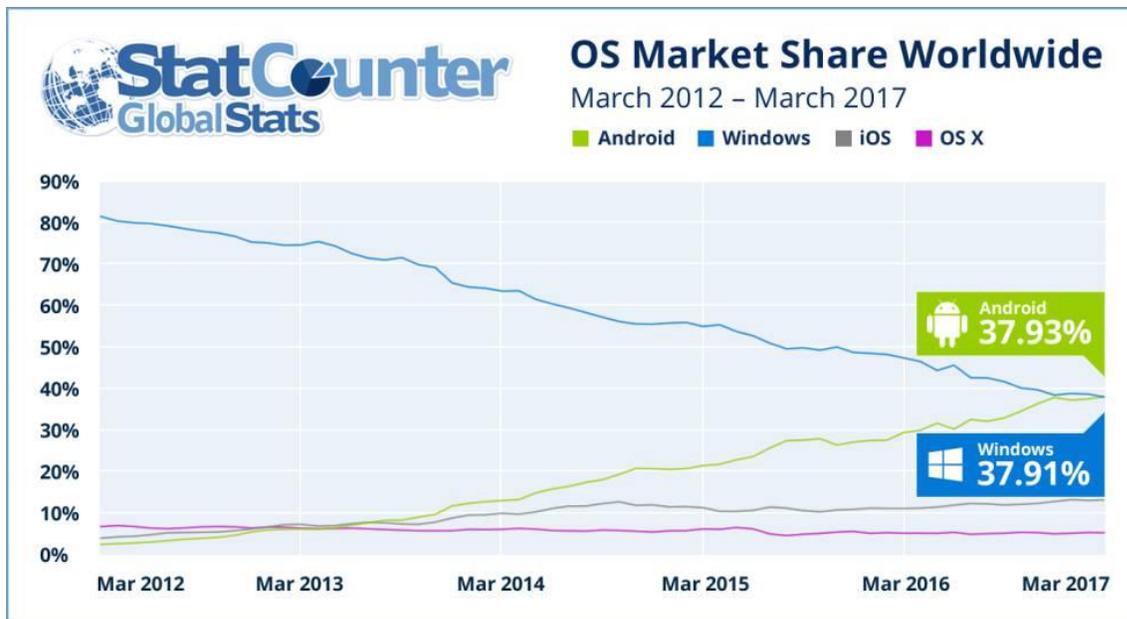


Figura 21. Sistemas operativos más usados en el mundo
Fuente: (José Zuriarrain, 2017)
Elaborado por: (José Zuriarrain, 2017)

La arquitectura de software que se uso es la de aplicaciones móviles autocontenidas, en el cual su contenido ya sea menú, imágenes, e información van a ser estáticas y su información puede ser almacenado en el mismo dispositivo, por este motivo y al ser una aplicación la cual su objetivo es realizar un examen de audiometría de tonos no es necesario usar más que los permisos de almacenamiento y uso de los auriculares, siendo estos requisitos necesarios para el almacenamiento de los resultados de las pruebas audiométricas y conexión de los auriculares externos.

A continuación, el esquema de aplicaciones autocontenidas.

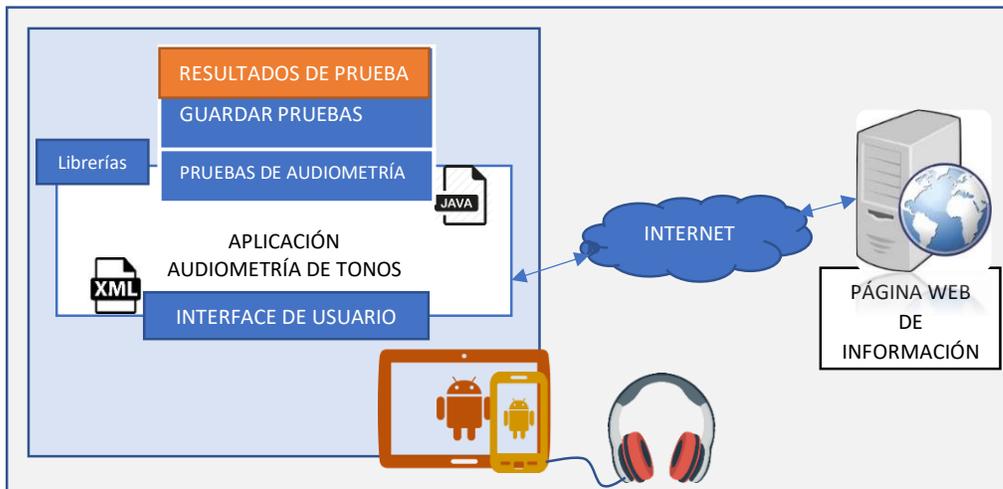


Figura 22. Esquema de aplicaciones autocontenidas
Elaborado por: Autor

Las ventajas de usar esta arquitectura son las siguientes:

- La aplicación puede funcionar en cualquier momento con la mayor velocidad y con todo el almacenamiento que cuente el dispositivo si fuese necesario.
- No requiere de esperas de servicio de conexión para usarla y cero gastos en lo que son servicios de hosting, bases de datos, entre otros.
- Por otra parte, para el modelado de la aplicación ya sean de la parte lógica, física, de procesos como de desarrollo se realizó con el Modelo “4+1” vistas de Kruchten y se realizaron las distintas vistas de la aplicación.

3.2. Datos.

Para el desarrollo de la aplicación se usó la metodología ágil XP (Programación Extrema), debido a las ventajas que tiene con respecto a las demás. A continuación, las ventajas y diferencias entre las distintas metodologías y cuál es la más adecuada para desarrollar la aplicación.

De acuerdo con el cuadro comparativo **Tabla.2**, la Metodología XP (Programación Extrema) es la más adecuada para este tipo de proyecto, ya que provee al programador un sinnúmero de ventajas.

Cabe mencionar que no existe una metodología universal para aplicarla con éxito a cualquier proyecto de desarrollo de software, toda metodología debe ser adaptada a todo el contexto

del proyecto como son: tiempo de desarrollo, tipo de sistema, costos, cantidad de desarrolladores, etc.

Sin embargo, las ventajas de la Programación Extrema son el ahorro de tiempo, recursos y se puede realizar un proyecto con 1 desarrollador.

- Presenta un código más simple y fácil de entender que permite la reducción del número de errores en su desarrollo.
- Capacidad de respuesta ante cambios permitiendo la adaptación a nuevos requisitos de la empresa u organización.

Para el diseño de los datos de uso la vista lógica de modelo 4+1 vistas de Kruchten, de los cuales se utilizó los distintos diagramas UML, para modelar las clases y datos necesarios para el desarrollo de las aplicaciones, para desarrollar estos diagramas se usó el programa de StarUML que cuenta con una gran variedad de diagramas y diagramas del modelo 4+1 vistas como se muestra en la **Figura 24**.

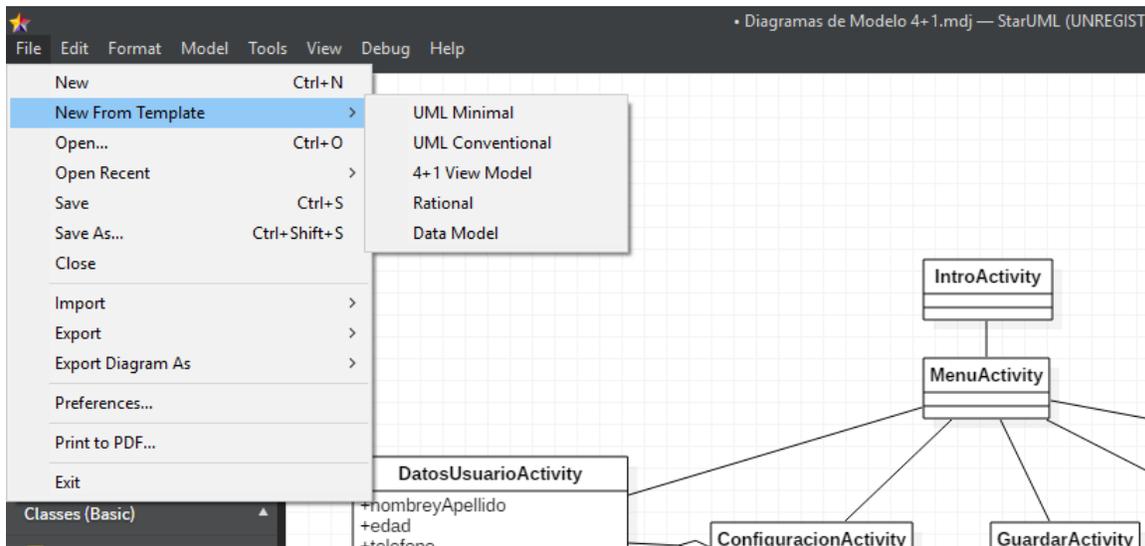


Figura 23. StarUML Modelo 4+1 vistas.
Elaborado por: Autor

Para el desarrollo de la codificación en Android Studio se usó como lenguaje de programación JAVA y XML.

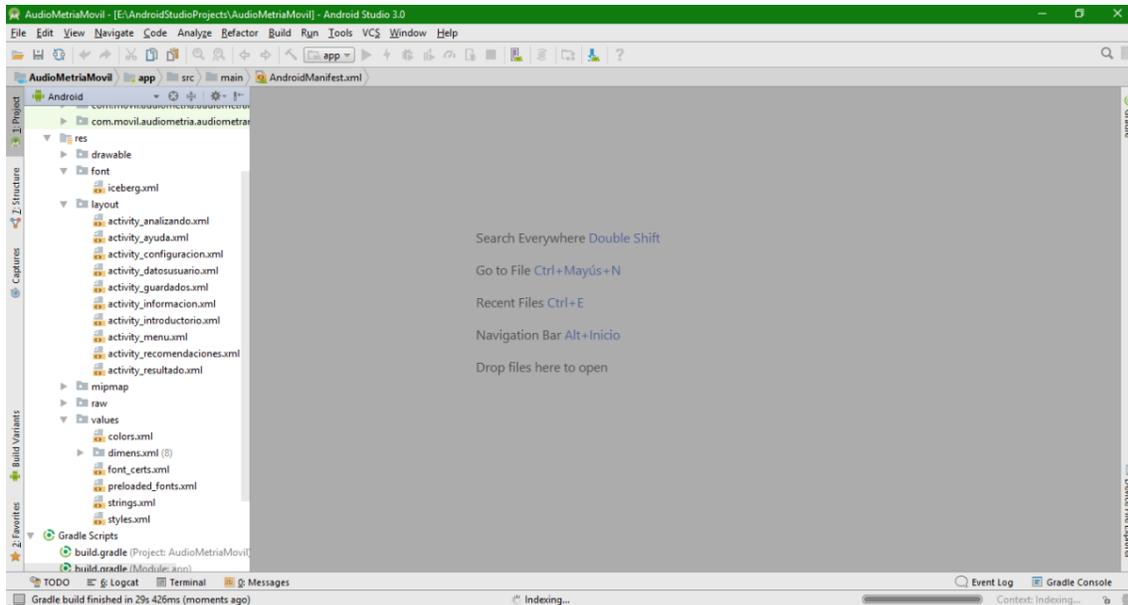


Figura 24. Plataforma de desarrollo.
Elaborado por: Autor

3.3. Interfaces.

El diseño conceptual de las interfaces se realizó mediante los diagramas UML, desarrollados por el modelo 4+1 vistas, para el desarrollo del código se usó la plataforma Android Studio, la cual permite realizar la codificación de las vistas de usuario en formato XML, esto es de gran utilidad ya que permite usar una misma interface para distintos dispositivos, además de diseñar los componentes con distintas dimensiones y distintos archivos Strings.xml, donde puede colocarse la información de cada componente, para luego enlazarlo con la vista y obtener al final un diseño responsivo de la aplicación. Esto ayudará para que en un futuro se pueda realizar una actualización de la aplicación, la cual cuente con múltiples lenguajes agregando más Strings.xml con la traducción de la información de los componentes y las dimensiones de nuevos dispositivos.

3.4. Implantación.

Para realizar la implantación de la aplicación se usó una Tablet Samsung virtualizada con AVD Manager que integra Android Studio en el computador. A continuación, se muestra el proceso de creación de una máquina virtual en las siguientes figuras:

1. Iniciamos el programa Android Studio y dentro del buscamos en la barra de menú el icono de  AVD Manager. Luego aparece una nueva ventana para iniciar con la creación de la máquina virtual, ya sea para una Tablet o smartphone,

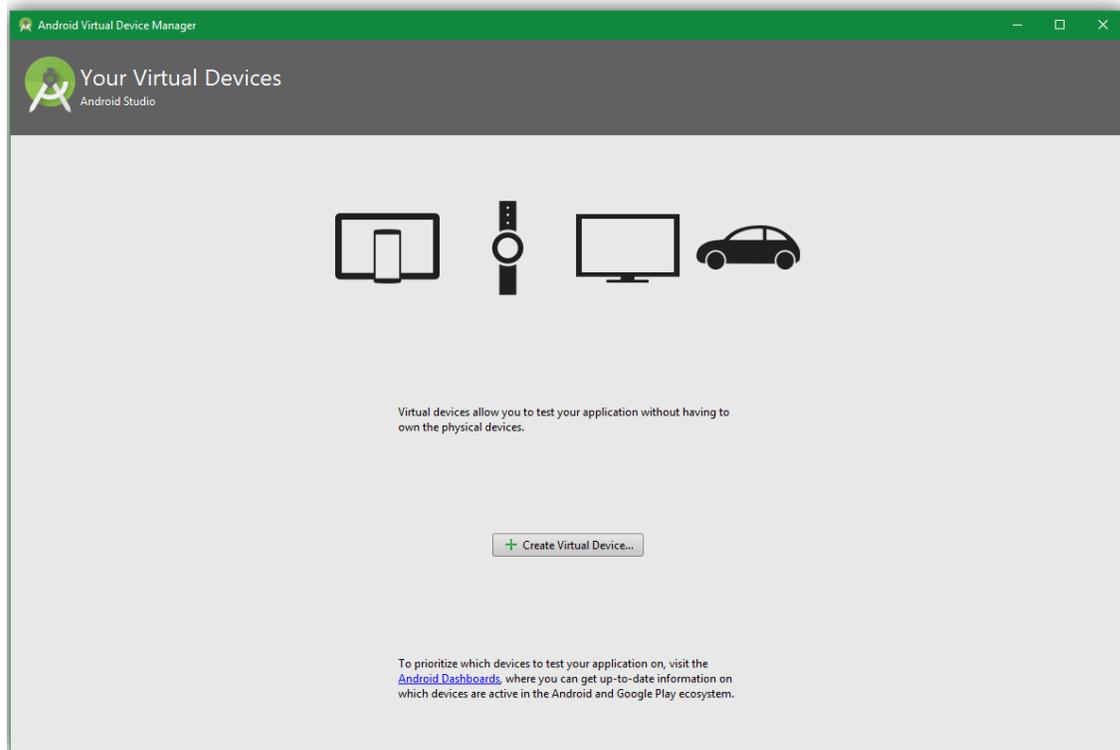


Figura 25. Ventana de creación de dispositivos virtuales.
Elaborado por: Autor

2. En esta nueva vista se puede especificar el dispositivo que se quiere virtualizar, Tablet o Smartphone y colocarle las características que desee como tamaño de pantalla, sistema operativo, memoria RAM, etc.

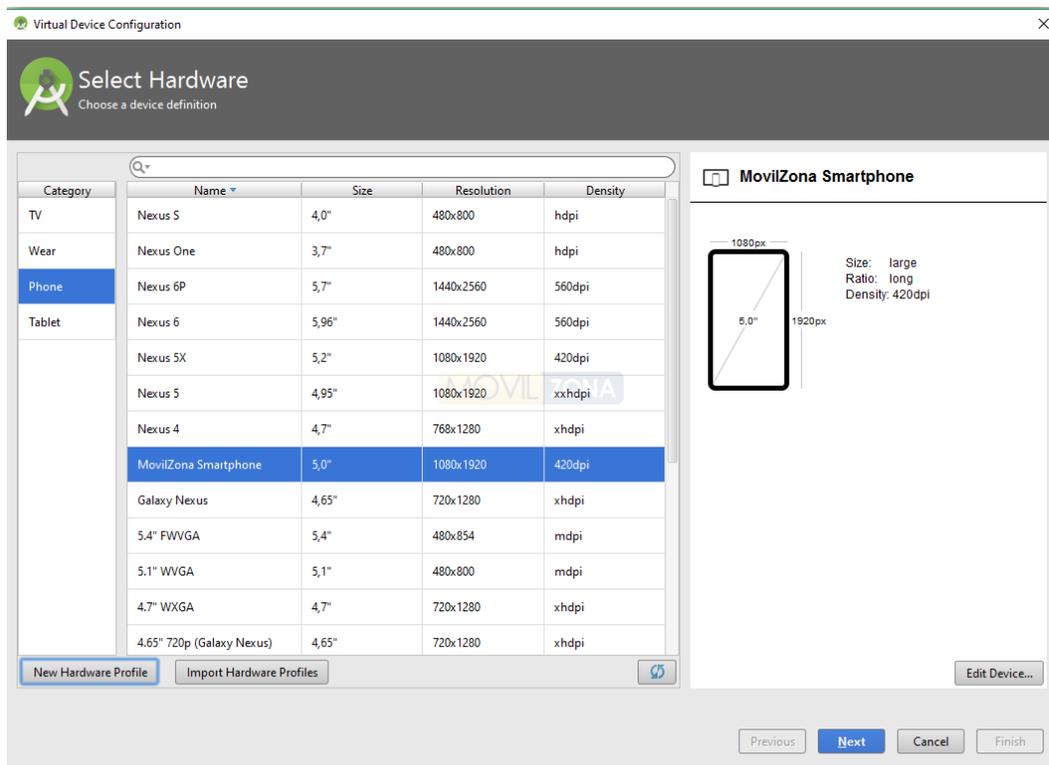


Figura 26. Ventana para escoger las especificaciones del dispositivo que se desea virtualizar.
Elaborado por: Autor

3. Por último, se finaliza y se ejecuta la máquina virtual.



Figura 27. Máquina virtual de una Tablet.
Elaborado por: Autor

Se realizó esta máquina virtual con el fin de tener acceso a muchos dispositivos móviles virtuales, es un método muy usado hoy en día por los desarrolladores, debido a que el mercado es tan variado en dispositivos, dimensiones, tamaños de pantalla como de píxeles por pulgada cuadrada. El costo de tener todos estos dispositivos físicos sería muy elevado para un desarrollador que quiere realizar las pruebas de funcionalidad de la aplicación, por esto la virtualización es lo más recomendable.

También se realizó las pruebas en 2 smartphone físicos de marcas Samsung y Huawei, la validación de su funcionalidad fue correctamente implantada, además, hay que mencionar que para la implantación será necesario recordar los requerimientos mínimos que tiene que tener el dispositivo móvil.

3.5. Fase de diseño.

Se elaboraron diseños breves que sirven de referencia para la implementación, otra práctica fundamental de la metodología de Programación Extrema (*XP*) es utilizar diseños tan simples como sea posible, el principio es utilizar el diseño más sencillo que consiga que todo funcione evitando diseñar características extra y que toman demasiado tiempo.

Para esta fase vamos a emplear el modelado UML ya que se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece varios diagramas de los cuales se pueden modelar sistemas, los desarrolladores lo eligen puesto que es un lenguaje que ayuda a discutir los problemas y soluciones implicadas en la construcción del sistema y principalmente porque es el lenguaje de modelado de software más conocido y utilizado.

Se optó por la realización de los siguientes diagramas:

- ✓ **Diagramas de casos de uso.**
- ✓ **Diagramas de clases**
- ✓ **Diagramas de secuencia.**
- ✓ **Diagrama de actividad**
- ✓ **Diagrama de componentes**

A parte de los modelados UML se complementa el diseño con modelos como el arquitectónico con el objetivo de ofrecer una visión simplificada del sistema, de forma que una persona pueda mirar el diagrama y entender lo que se quiere conseguir o desarrollar.

El modelo de navegación nos ayuda en la comprensión del orden de presentación de las pantallas de la aplicación con los contenidos y los vínculos que existen en cada una de ellas.

3.5.1. Vista lógica.

Diagrama de clases.

El diagrama que se muestra a continuación (*Ver Diagrama 3*) está directamente relacionado con las clases que se utilizaron en la aplicación móvil. Es importante recalcar que algunas clases son creadas de acuerdo con la necesidad del proyecto. El en siguiente diagrama vamos a ver todas las distintas clases con sus respectivos atributos y como están relacionados unas con otras.

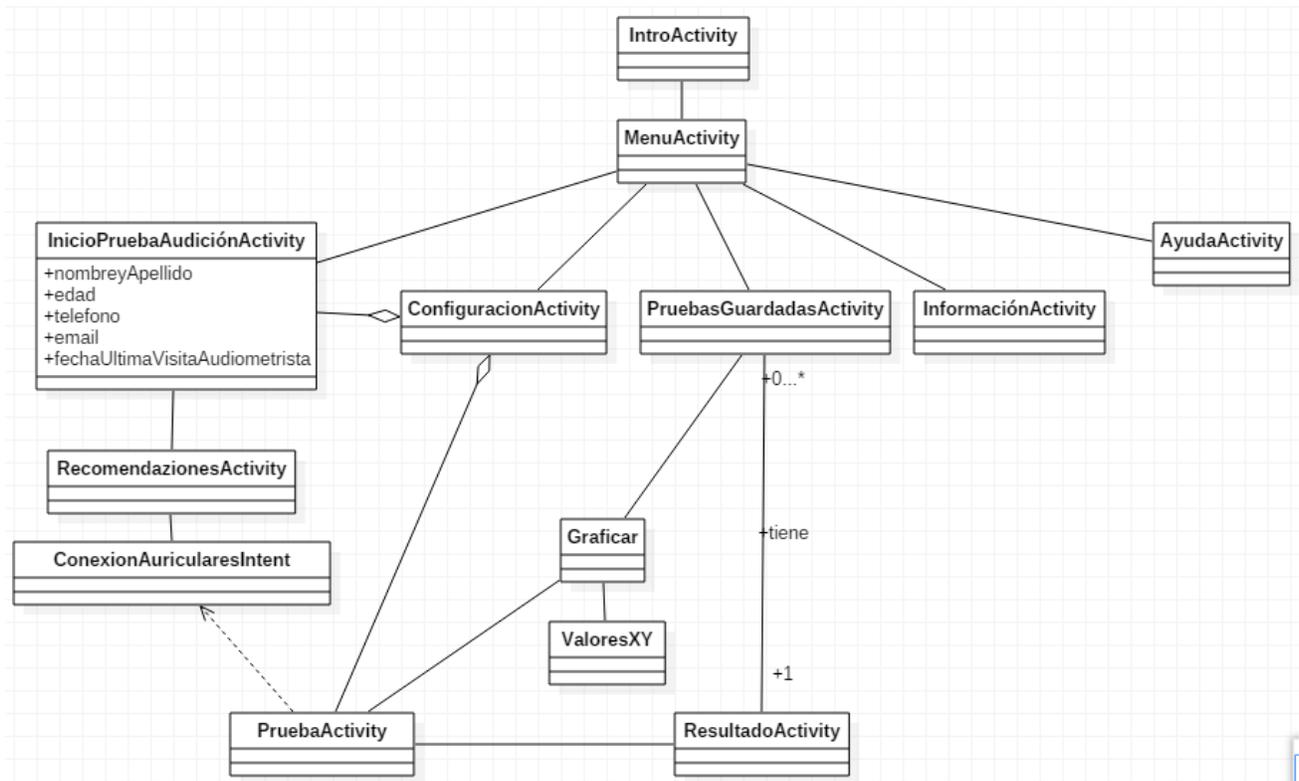


Diagrama 2. Diagrama de clases.
Elaborado por: Autor

3.5.2. Vista de desarrollo.

Diagrama de componentes.

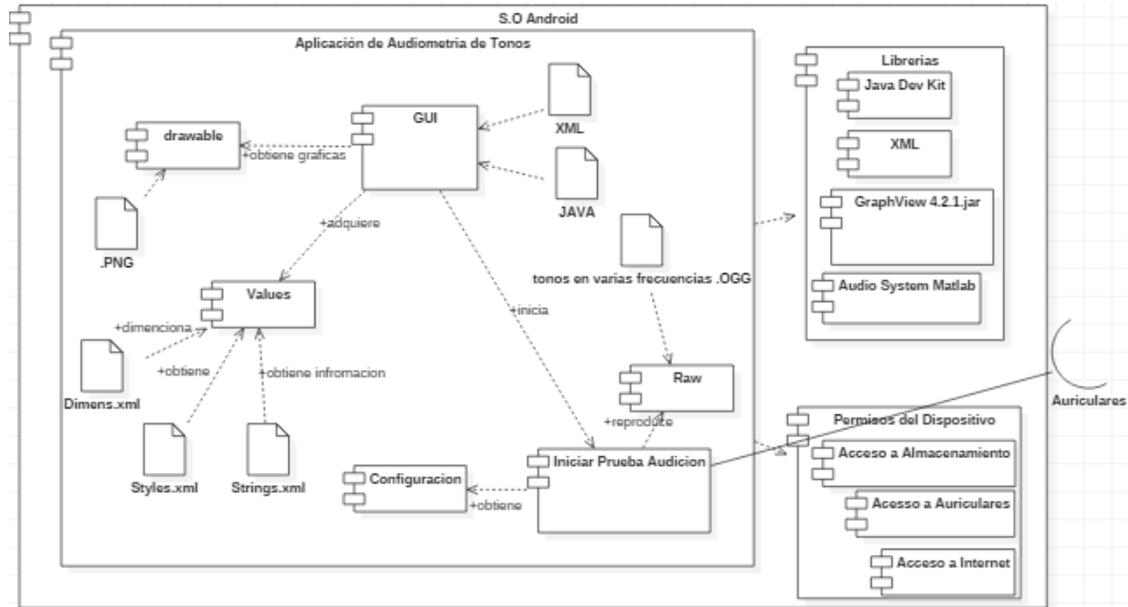


Diagrama 3. Diagrama de componentes, pruebas guardadas.

Elaborado por: Autor

3.5.3. Vista de procesos.

Diagrama de actividades

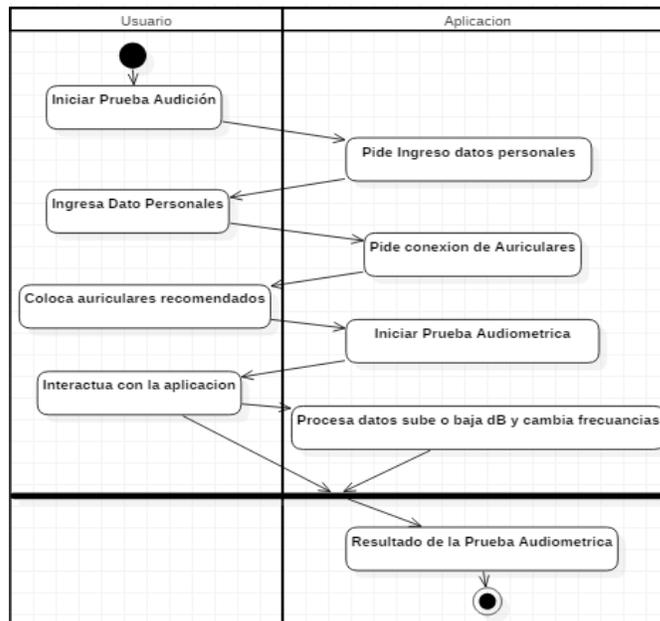


Diagrama 4. Diagrama de actividades, inicio prueba audiometría tonal.

Elaborado por: Autor

3.5.4. Diagrama de navegación.

El diagrama de navegación indica o muestra el orden de relación de las pantallas o componentes del software y sirve para la comprensión del orden de presentación de las pantallas o Activities con los contenidos de la aplicación.

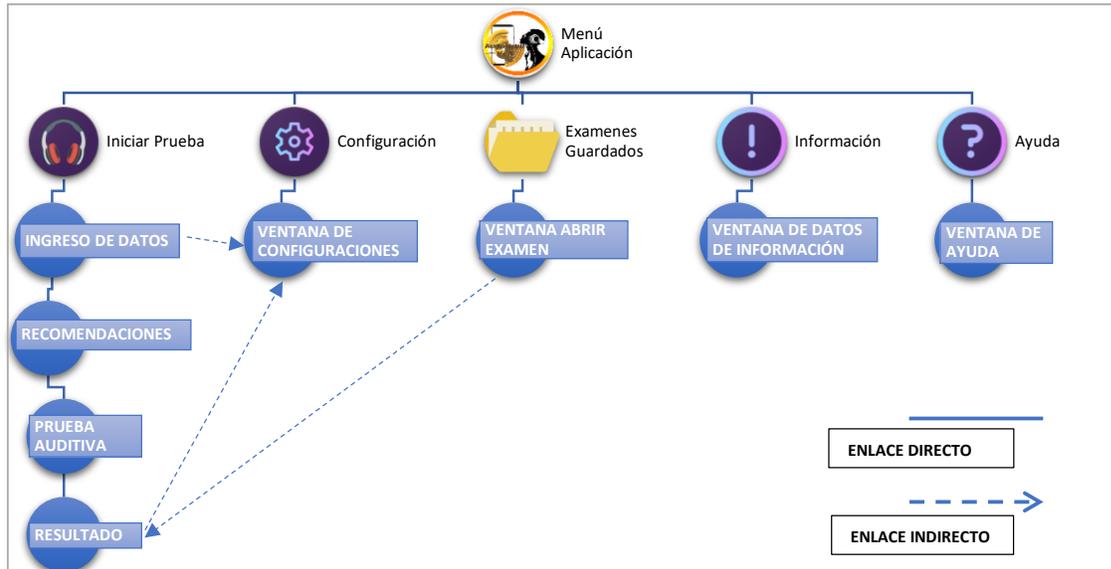


Diagrama 5. Diagrama de navegación.
Elaborado por: Autor

3.5.5. Diseño de los bocetos de las pantallas del sistema (prototipo).

Para la parte de diseño de prototipos del proyecto se realizó con la ayuda de una herramienta Web (Online) llamada *Ninja Mock*. La podemos utilizar ingresando a la página: <http://ninjamock.com/> Basta con registrarnos para empezar a utilizarla.

Al ser una herramienta de nivel intermedio y no muy avanzada sirve para la creación de bocetos para dispositivos móviles con sistema Operativo como: Android, iOS y Windows Phone. Los elementos que se agregan a la aplicación ofrecen un acabado de dibujo a mano alzada, su principal ventaja es que es gratuita. La interacción de sus prototipos no va más allá de enlaces entre pantallas, para ideas básicas, para un diseño rápido y para un primer boceto, resulta muy eficiente para lo que buscamos.

3.5.5.1. Boceto de la pantalla de IntroAPP.

Detalles técnicos:

- El video de introducción es formato MP4.

- El video es en resolución Full HD o 1920x1080.
- Cuenta con sonido
- Tiene logo de la aplicación.

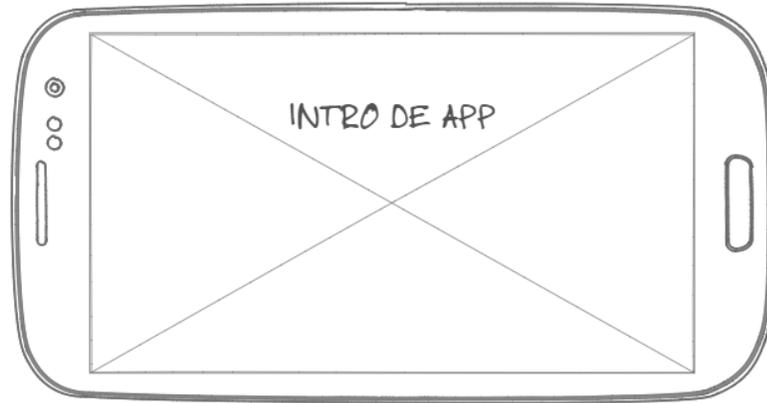


Figura 28. Prototipo pantalla de IntroAPP.
Elaborado por: Autor

3.5.5.2. Boceto de la pantalla del menú.

Detalles técnicos:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos.
- Los iconos tienen un formato .png para evitar fondos.
- Los iconos como las imágenes tienen distintos tamaños para cada resolución de pantallas.
- La letra está ajustada a cada tipo de pantalla para que no sea ni muy pequeña ni muy grande
- Tiene un fondo atractivo que ayude al usuario a confiar en la aplicación y sea de formato jpg.

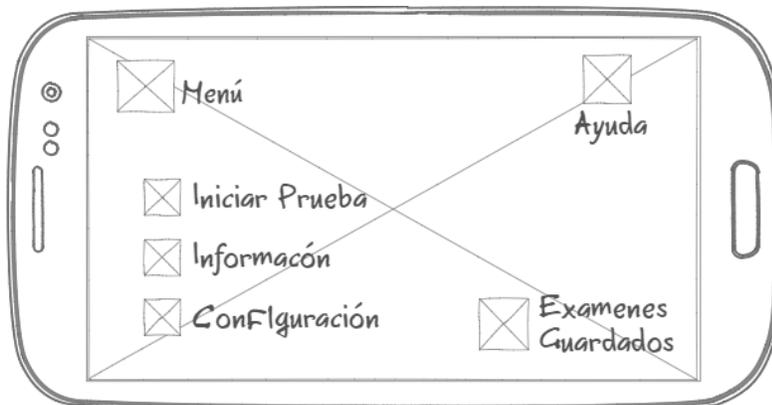


Figura 29. Prototipo pantalla menú.
Elaborado por: Autor

3.5.5.3. Boceto de la pantalla de inicio de prueba de audición.

Detalles técnicos:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos.
- Los iconos como las imágenes tienen distintos tamaños para cada resolución de pantallas.
- Cuenta con un Switch botton para poder activar.
- Cada TextView esta diseñado para pedir datos alfanuméricos, numéricos o alfabéticos dependiendo del caso.
- Cuenta con botón que seguirá a la siguiente actividad.
- Colocar un fondo de formato jpg.

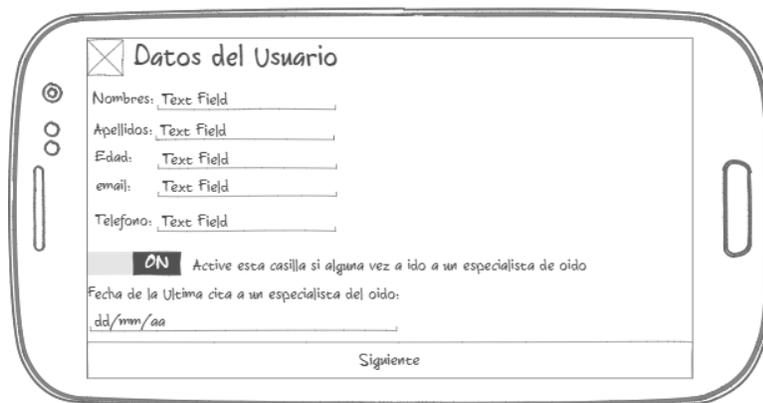


Figura 30. Prototipo pantalla datos de usuario.
Elaborado por: Autor

Detalles técnicos de la pantalla de recomendación:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos
- Los iconos como las imágenes tengan los distintos tamaños para cada resolución de pantallas.
- Colocar una condición que si no se conectan los audífonos no se puede seguir con la prueba audiometricas, ademas con un alert que me muestre cual es problema.
- Colocar imágenes y texto de recomendación.
- Colocar un fondo de formato jpg.

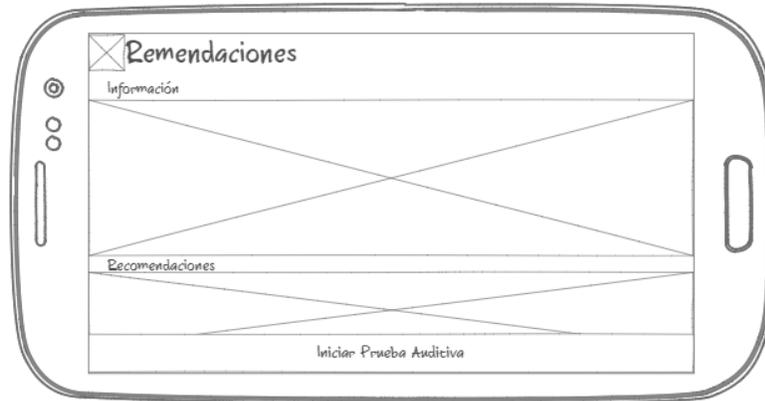


Figura 31. Prototipo pantalla de recomendación.
Elaborado por: Autor

Detalles técnicos de la pantalla de Graficar:

- La grafica de audición cuenta con el 85% de la pantalla.
- Los botones como los radioButtons usan el 15% de la pantalla restante.
- Los botones tienen un color representativo dependiendo de su acción
 - Rojo: no escucho
 - Verde: si escucho
 - Azul: apenas puedo oír
- Las líneas de la gráfica tienen color específico dependiendo del oído
 - Rojo oído derecho
 - Azul oído izquierdo
- Además cada oído en la gráfica debe estar representado por un símbolo gráfico
 - Un círculo: oído derecho
 - Una X: oído izquierdo
- Al llegar a un nivel muy alto de decibelios en cada frecuencia se grafique un símbolo distinto al terminar.
 - Un círculo con una flecha para abajo: ausencia de respuesta oído derecho
 - Una X con flecha: ausencia de respuesta oído izquierdo

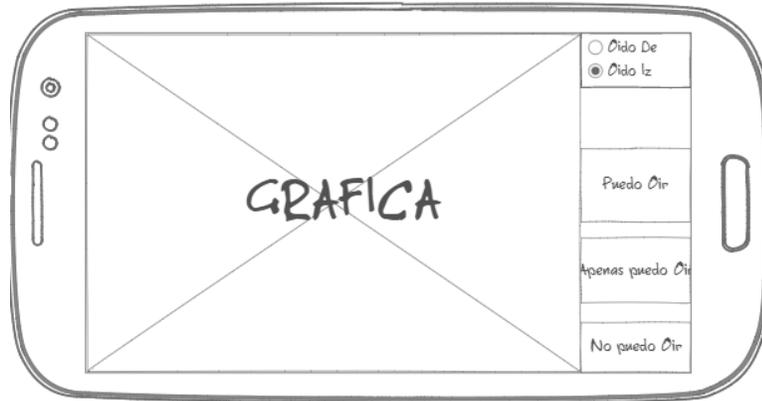


Figura 32. Prototipo pantalla prueba auditiva.
Elaborado por: Autor

Detalles técnicos:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos
- Los iconos como las imágenes tienen distintos tamaños para cada resolución de pantallas.
- Colocar la gráfica de los 2 oídos con sus niveles de pérdida.
- Colocar imágenes y texto de niveles de pérdida auditiva y que significa cada una.
- Colocar un fondo de formato jpg.



Figura 33. Prototipo pantalla respuesta.
Elaborado por: Autor

3.5.5.1. Boceto de la pantalla de configuración.

Detalles técnicos:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos
- Los iconos como las imágenes tienen los distintos tamaños para cada resolución de pantallas.
- Usar botones Switch para activar parámetros avanzados

- Usar opción buttons para escoger entre opciones.
- Usar alerts para avisar un cambio erróneo o correcto.
- Para las frecuencias usar checkbox para escoger entre frecuencias y botones para frecuencias predeterminadas y aceptar los cambios.
- Colocar un fondo de formato jpg.

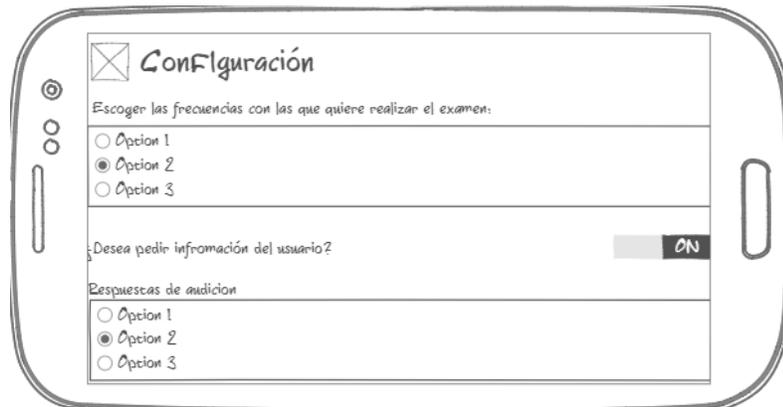


Figura 34. Prototipo pantalla configuración.
Elaborado por: Autor

3.5.5.2. Boceto de la pantalla de información.

Detalles técnicos de la pantalla:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos
- Los iconos como las imágenes tienen los distintos tamaños para cada resolución de pantallas.
- Colocar una condición que si no se conectan los audífonos no se puede seguir con la prueba audiométrica, además con un alert que me muestre cual es problema.
- Colocar imágenes y texto de información sobre audiometría y que hace la aplicación.
- Colocar un fondo de formato jpg.

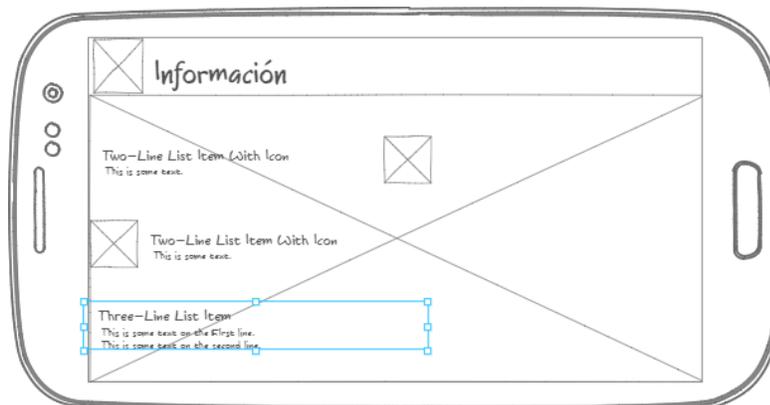


Figura 35. Prototipo pantalla información.
Elaborado por: Autor

3.5.5.3. *Boceto de la pantalla de ayuda.*

Detalles técnicos:

- Usar framelayout para evitar solapamiento entre opciones de menú e iconos
- Los iconos como las imágenes tienen los distintos tamaños para cada resolución de pantallas.
- Colocar una condición que si no se conectan los audífonos no se puede seguir con la prueba audiométrica, además con un alert que me muestre cual es problema.
- Colocar imágenes y texto de recomendación.
- Colocar el icono de la aplicación con formato .png
- Colocar el link web de la página de la aplicación usando un botón.
- Colocar información de su desarrollo en textview
- Colocar un fondo de formato jpg.

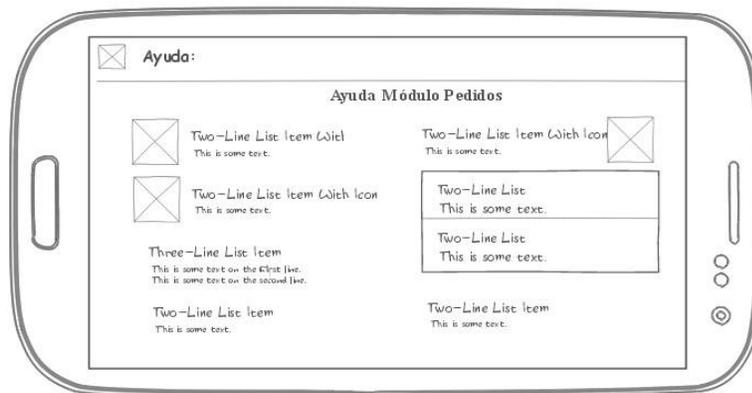


Figura 36. Prototipo pantalla ayuda.
Elaborado por: Autor

**CAPITULO IV.
CONSTRUCCIÓN**

4.1. Librerías.

Están normalmente hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez.

4.1.1. Uso de la librería Java y XML.

Para la realización de la aplicación se usó las distintas bibliotecas que vienen incorporadas en la herramienta de desarrollo de Android Studio ya sean para Java como para XML.

4.1.2. Uso de la librería GraphView.

Se usó la librería GraphView para crear programáticamente diagramas flexibles ya sean líneas, barras graficas de puntos o implementar cualquier tipo personalizo de gráfica.

Para poder usar esta librería lo primero que se debe realizar es la integración de una línea de código en la siguiente dirección del proyecto /Gladle en Android Studio dentro de ella en el archivo build. gradle bajo el directorio de la aplicación en el bloque de dependencias colocamos:

```
compile 'com.jjoe64:graphview:4.2.1'
```

Ahora para continuar con la creación del elemento GraphView es necesario en una vista (Layout XML) colocar el siguiente código:

```
<com.jjoe64.graphview.GraphView  
android:layout_width="match_parent"  
android:layout_height="200dip"  
android:id="@+id/graph" />
```

El cual nos permitirá declarar el elemento grafica GraphView e iniciar con la inicialización en el código Java.

Como ejemplo de una gráfica de serie de algunos datos sacado de <http://www.android-graphview.org/simple-graph/>:

```
GraphView graph = (GraphView) findViewById(R.id.graph);
```

```

LineGraphSeries<DataPoint> series = new LineGraphSeries<>(new DataPoint[] { new
DataPoint(0, 1),
new DataPoint(1, 5),
new DataPoint(2, 3),
new DataPoint(3, 2),
new DataPoint(4, 6)
});
graph.addSeries(series);

```

Y su grafica correspondiente, a continuación, en la Figura. 38:

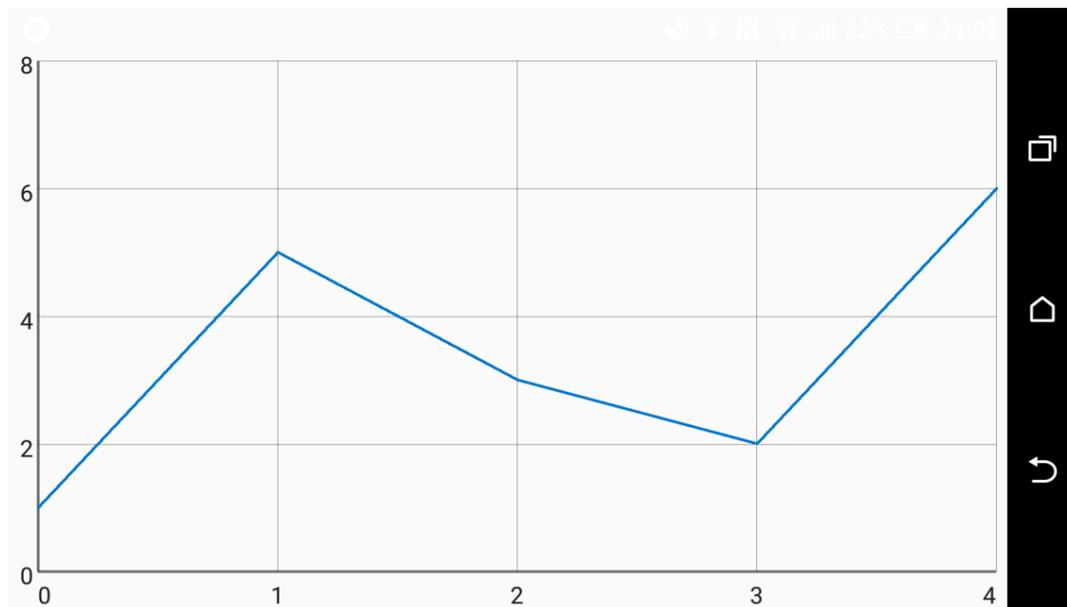


Figura 37. Grafica de ejemplo con GraphView.

Fuente: <http://www.android-graphview.org/simple-graph/>

Elaborado por: Autor

4.1.3. Uso de las librerías de la plataforma Matlab.

Según la información consultada Android Studio no cuenta con las herramientas necesarias para la creación de los tonos de audio necesarios para realizar la audiometría, por eso se buscó otra herramienta para poder integrar los tonos con la aplicación de audiometría, se usó Matlab ya que es una plataforma de desarrollo que cuenta con una herramienta para la creación de audio de alta calidad llamada **Audio System Toolbox**, que proporciona bibliotecas de algoritmos de procesamiento de audio como filtrado, ecualización, control de rango dinámico, fuentes (osciladores de audio y sintetizadores de tabla de ondas) con la

misma se puede crear las distintas frecuencias de tonos con una calidad excelente para la audiometría de tonos puros para dispositivos móviles.

4.2. Desarrollo.

En este subcapítulo se aplicó las fases para el desarrollo de una aplicación móvil según la metodología Programación Extrema (XP) como son: La fase de planificación en donde se analiza involucrados y requerimientos de software. La fase de diseño donde está toda la parte de diagramación, La fase de codificación que es concretamente la fase de programación y por último la fase de pruebas donde se realiza las pruebas respectivas con su debida documentación.

Algunas frecuencias pueden generar sonidos muy perturbadores en ciertos decibeles, esto ocasiona que, al cambiar de frecuencia en la prueba, el oído siga con el sonido de la anterior frecuencia y no se logre percibir si escucha o no en la actual frecuencia, lo que puede producir resultados erróneos. Esto se tomó en cuenta para el desarrollo de la aplicación móvil.

4.2.1. Fase de planificación.

Descripción de los interesados (Stakeholders).

Stakeholders: Involucrados o interesados, son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa. Tomado de (GrupoWikipedia,2015).

En la siguiente matriz se puede apreciar cuáles son los stakeholders o involucrados en el proyecto de desarrollo de la aplicación, para dispositivos móviles Android que permitirá realizar un test de audiometría de tonos puros, enfocado al consumidor.

Tabla 12. Matriz de Stakeholders del proyecto

NOMBRE	CARGO	INSTITUCIÓN	RELEVANCIA PROYECTO (1-5)
Usuario(s)	Consumidor	Cualquiera	5
Paúl Andrés Landázuri Peña	Desarrollador del Trabajo de Titulación	Universidad Técnica Particular de Loja	5

Dr. Fausto Coello Serrano	Especialista de Audiometría	Universidad Central del Ecuador	3
Ing. Marco Patricio Abad Espinoza	Director del trabajo	Universidad Técnica Particular de Loja	1

Elaborado por: Autor

Todos cumplen un papel fundamental dentro de la aplicación a desarrollar, pero unos tienen más relevancia que otros, Ver **Tabla 12**. Para un mejor entendimiento de las funciones se realizó otra tabla con las tareas o funciones de cada involucrado. Ver **Tabla 13**.

Tabla 13. Resumen de las tareas de los involucrados.

CARGO	DESCRIPCIÓN	RESPONSABILIDAD
Consumidor	Persona o personas que usaran la aplicación.	<ol style="list-style-type: none"> 1. Validar la aplicación 2. Dar uso de aplicación.
Desarrollador del trabajo de titulación	Mi persona, encargado de desarrollo del trabajo de titulación hasta su finalización.	<ol style="list-style-type: none"> 1. Desarrollar la Aplicación de Audiometría Tonal. 2. Desarrollar el documento del Trabajo de Titulación.
Especialista de audiometría	Persona que me entregó la información para realizar la aplicación.	<ol style="list-style-type: none"> 1. Explicación sobre cómo se realiza una prueba de audiometría. 2. Información sobre los requerimientos necesarios que la aplicación debe tener.
Tutor	Persona que dirige el proyecto.	<ol style="list-style-type: none"> 1. Verificar el correcto desarrollo del proyecto. 2. Revisar avances proyecto.

Elaborado por: Autor

4.2.2. Funcionalidades de la aplicación.

La aplicación está compuesta de 10 partes o módulos en las cuales se detalla cada una de las opciones con las que cuenta:

1. IntroAPP

- Se visualiza el video introductorio de la aplicación de audiometría

2. Menú

- Visualiza las distintas opciones que tiene la aplicación
 - Visualizar inicio
 - Visualizar configuración
 - Visualizar exámenes guardados
 - Visualizar información
 - Visualizar ayuda

3. Iniciar prueba de audición

- Añadir Datos
 - Nombres
 - Apellidos
 - Edad
 - Email
 - Teléfono
- Añadir la última visita al especialista en audiometría
 - Ingresar la fecha del último examen audiométrico

4. Configuración

- Visualizar opciones de configuración
 - Activar o desactivar pedido de información del usuario
 - Selección entre resultados
 - Banana auditiva
 - Niveles de pérdida
 - Añadir o quitar frecuencias de análisis
 - Poder regresar a las frecuencias por defecto

5. Exámenes guardados

- Visualizar un elemento Spinner
 - Seleccionar el nombre del usuario
 - Visualizar el diagrama audiométrico de resultados

6. Información

- Visualizar la información sobre audiometría de tonos.

7. Ayuda

- Visualizar información de ayuda
- Visualizar información sobre la aplicación
- Su creador
- Versión
- Página web de referencia

Para un mejor entendimiento podemos revisar más adelante la interacción de los usuarios con los diferentes módulos en los casos de uso.

4.2.3. Atributos de calidad.

Rendimiento de la aplicación

- La aplicación procesa los resultados (datos) al usuario en tiempo real.
- El tiempo de respuesta promedio de la aplicación dependiendo del celular no debe superar los 6 segundos.

Seguridad

- Los datos del usuario como los exámenes guardados van a estar almacenados en la memoria, podrán ser leídos solo por la aplicación y sin posibilidad de acceso por otra aplicación externa.

Disponibilidad

- La aplicación estará disponible el 100% del tiempo, ya que se trata de una aplicación nativa que se instalará en el dispositivo móvil.
- La aplicación será cargada al Google Play Store de Android, cuando sea completamente validada.
- La aplicación contará con una página web con información sobre su desarrollo, información de audiometría y con direccionamiento a la tienda de Google Play Store de Android para descargar la aplicación.

Portabilidad

- Compatibilidad con plataformas: El sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.4.0. ya sea en una Tablet o un Smartphone.

Operatividad

- La aplicación podrá ser operada por cualquier usuario ya que incluye información indispensable, para que su manipulación sea lo más sencilla posible.

4.2.4. Restricciones.

El dispositivo móvil en donde se ejecute la aplicación debe tener los siguientes requisitos mínimos, debido a que necesitamos velocidad de procesamiento para reproducir las distintas frecuencias tonales y además necesitaremos almacenamiento interno del dispositivo.

Las restricciones descritas y analizadas en la **Tabla. 10**, son solo una sugerencia para el óptimo funcionamiento de la aplicación, que sea rápida, tenga una buena visualización, tener un mínimo de almacenamiento libre en el dispositivo y no quitar o restringir que la aplicación pueda funcionar en dispositivos de menor gama con procesamientos más lentos.

4.3. Fase de codificación.

En esta fase se necesita entender muy bien los requerimientos, como los diagramas UML, la información adicional obtenida del especialista en audiometría y de los encuestados, para posteriormente poderlos plasmar en la aplicación de audiometría móvil.

Se inicia el proceso de codificación en Android Studio y se explica el desarrollo en una breve descripción del código, de las clases más relevantes dentro de la aplicación como son:

- La Clase Java Menú (clase principal, llamada diferentes objetos)
- La Clase Java Prueba Auditiva (clase donde se va a realizar el análisis audio métrico de tonos)
- La Clase Java Configuración (clase donde se va a realizar la codificación de las distintas configuraciones que se van a poder realizar en la aplicación)
- El Archivo XML Activity_Menu (crear interfaz para el usuario)

4.3.1. Requerimientos de desarrollo.

En el **Anexo C** se describe sobre Android Studio y su instalación, además las siguientes son las herramientas, servicios y aplicaciones que se necesita para empezar a desarrollar un proyecto en Android.

- Entorno de desarrollo para aplicación.
 - JDK Java.
 - Android Studio (IDE) o Eclipse (IDE) con ADT (Plugins)
 - Android SDK (librerías y paquetes)
 - Librería de GraphView
 - IDE Matlab con librerías para la creación de los tonos en diferentes frecuencias.

4.3.2. Desarrollo del módulo de prueba audición.

Se explica con breve detalle el código de algunas llamadas o líneas donde es importante acotar una explicación, de las clases más relevantes del proyecto:

4.3.2.1. Descripción la clase Java graficar.

Ver **Anexo D.**

4.3.2.2. Descripción de la clase Java prueba auditiva.

Ver **Anexo E.**

4.3.2.3. Descripción de la clase Java configuración.

Ver **Anexo F.**

4.3.2.4. Archivo XML Activity_Menu.xml (menú principal).

Ver **Anexo G.**

4.3.2.5. Descripción del código para la creación de tono en la plataforma Matlab

Se utilizó la plataforma Matlab la cual cuenta con librerías para la creación, modificación, edición y mezclas de audio en distintas calidades. A continuación, describo el código que se usó.

```

clear; %limpiamos listado de variables usadas
clc; %limpiamos pantalla

fs=44100; %definimos variable frecuencia de sampleo
f=18000; %definimos variable frecuencia de nuestro sonido

duracion_segundos=20; %definimos la duracion de nuestro sonido

b=f*2*pi; %definimos variable b a utilizar en nuestra funcion a*sin(bt+c)
a=1; % a debe ser 1 para que la funcion no se distorcion al reproducirse
c=0;

t= [0:1/44100: duracion_segundos];

x=[sin(b*t)];

audioplayer (x, fs); %reproducir el sonido

% para hacer un grafico de la onda de sonido
sub_t= (1:100); % tomamos los 100 primeros puntos del vector t

sub_x=x (1:100); % tomamos los 100 primeros puntos del vector x

stem (sub_t, sub_x); %graficamos
%sound (x, fs);
filename='a18000hz.wav'
audiowrite (filename, x, fs); %guardamos el sonido en .wav

```

4.4. Fase de pruebas.

Una vez finalizada la fase de codificación y la comprobación del correcto funcionamiento de la aplicación en cada uno de sus interfaces, se realizaron las pruebas respectivas tomando en cuenta los aspectos siguientes:

1. Para la primera prueba funcional se realizó un análisis del código y posterior compilación para obtener el formato de archivo .apk que es el ejecutable que reconoce el sistema

operativo de Android, dado el caso de que no se pudiese compilar el archivo se obtendrá una serie de errores con la respectiva información del error y posterior proceder a corregirlo y finalmente obtener al archivo correctamente compilado.

2. Se aprobó la aplicación en diferentes dispositivos virtuales con versiones distintas del sistema operativo Android, como también de distintas dimensiones de pantalla, densidad de pixeles y todos estos dispositivos van a cumplir con los requisitos mínimos de hardware de nuestra aplicación.
3. Realizar las respectivas pruebas funcionales de las ventanas, componentes, imágenes, la graficación del diagrama, funcionalidad de los algoritmos y reproducción de tonos en las diferentes frecuencias de audio, como también las distintas configuraciones de la aplicación y el almacenamiento del diagrama audiométrico y datos del usuario.

4.4.1. Elaboración y ejecución del plan de pruebas.

Para la primera prueba funcional, se realizó la compilación de la aplicación en Android Studio como se muestra a continuación en la **Figura.39**.

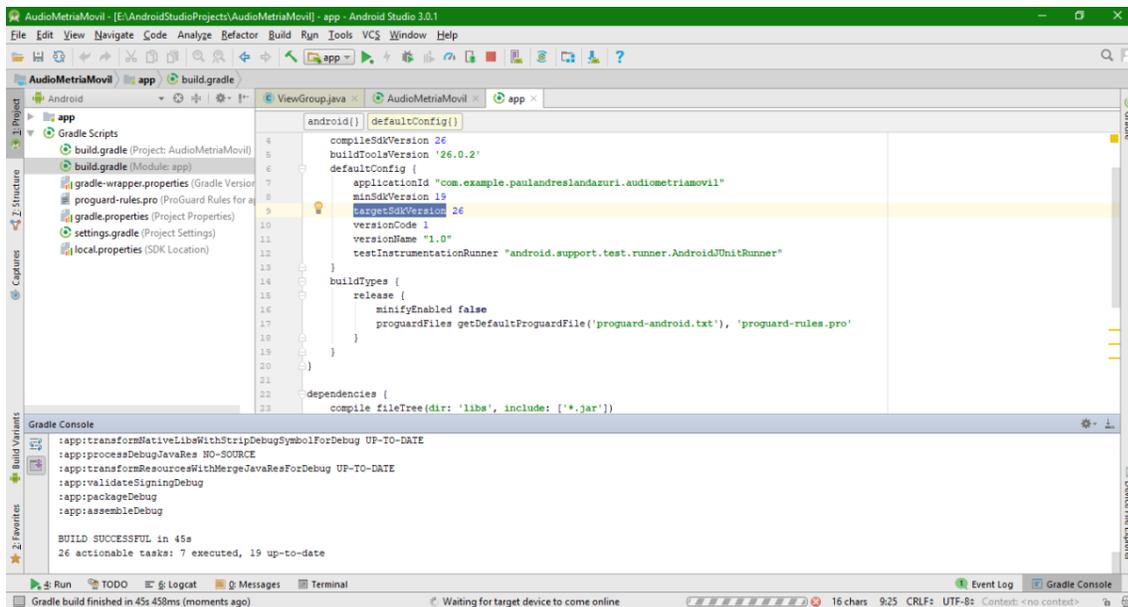


Figura 38. Compilación de la Aplicación
Elaborado por: Autor

La segunda prueba funcional, se realizó en una máquina virtual con una Tablet (Ver **Figura 40.**) de las siguientes características:

1. Samsung Galaxy Tab 3 T211
2. Dual Core 1.2 GHz de 1GB de RAM
3. Sistema Operativo Android 4.1.2 *Jelly Bean*

Mediante la herramienta de uso de memoria se verificó como la aplicación interactúa con el dispositivo y que modulo consume más memoria, luego de esto se realizó una optimización del código, para que funcione más rápido.



Figura 39. Pruebas realizadas en un Samsung Galaxy Tab 3 T211
Elaborado por: Autor

La tercera prueba se realizó una prueba funcional relacionada a la interacción en la interface de la “PRUEBA DE AUDICIÓN”, con el objetivo de observar la diagramación y el correcto funcionamiento de la reproducción de los tonos.

Tabla 14. Interacción con la interface de “prueba de audición”.

Historia de usuario: Autenticación correcta de usuarios.	
Descripción:	Una vez que se ha ejecutado la aplicación previamente instalada se debe ingresar en el menú de la aplicación, dar clic en “iniciar prueba de audición” y luego de llenar los datos de usuario seguir con las recomendaciones para usar la aplicación e iniciar una nueva ventana donde se indica como interactuar con la aplicación y se continua con la escucha de

	los tonos y mediante los botones se da la respuesta, la cual veremos en la diagramación del grafico de audiometría del oído izquierdo como derecho.
Condiciones de ejecución	Tiene que estar conectado los auriculares de tipo intra-aural y colocados correctamente en el oído correcto mediante la observación de la letra en el auricular si dice I o D o en inglés L o R que significa oído izquierdo o derecho respectivamente.
Entrada	<ol style="list-style-type: none"> 1. Presionar el botón “Puedo Oír” si puede escucharlo muy bien. 2. Presionar el botón “Apenas Puedo Oír” para confirmar cuando escuche el tono o cuando deje de escucharlo. 3. Presionar el botón “No Puedo Oír” si, no puede escucharlo.
Resultado esperado	<ol style="list-style-type: none"> 1. Al presionar el botón “Puedo Oír”, bajara los dB hasta llegar al menor valor de 0dB con la condición de que si estuvo el tono es menor igual a 1000Hz seguir reduciendo la frecuencia hasta llegar al valor de la frecuencia menor y luego de esto subir al valor mayor a 1000Hz y de igual manera hasta llegar a la frecuencia mayor y cambiar automáticamente el examen del oído al otro oído y si ya se lo realizó terminar el proceso y visualizar la respuesta en la nueva ventana que se abrirá. 2. Al presionar el botón “No Puedo Oír”, subirá los dB hasta el valor mayor que se puede realizar con esa frecuencia y seguidamente iniciara con la condición siguiente de que sí estuvo el tono es menor igual a 1000Hz seguir reduciendo la frecuencia hasta llegar al valor de la frecuencia menor y luego de esto subir al valor mayor a 1000Hz y de igual manera hasta llegar a la frecuencia mayor y cambiar automáticamente el examen al otro oído y si ya se lo realizo terminar el proceso y visualizar la respuesta en la nueva ventana que se abrirá.

	<p>3. Presionar el botón “Apenas Puedo Oír”, se cambia de tono de frecuencia con la condición de que sí estuvo el tono es menor igual a 1000Hz seguir reduciendo la frecuencia hasta llegar al valor de la frecuencia menor y luego de esto subir al valor mayor a 1000Hz y de igual manera hasta llegar a la frecuencia mayor y cambiar automáticamente el examen al otro oído y si ya se lo realizo terminar el proceso y visualizar la respuesta en la nueva ventana que se abrirá.</p> <p>Y cualquiera de los botones podrá terminar la prueba cuando termine su proceso.</p>
Evaluación de la prueba	Prueba satisfactoria

Elaborado por: Autor

4.4.2. Análisis de resultados.

Tabla 15. Análisis de resultados de la primera prueba funcional.

N °	Actividad	Responsable	Observaciones
1.	IntroAPP	Desarrollador (Yo)	El intro funciona correctamente, la aplicación se demora el tiempo en el que se termina el video.
2.	Menú	Desarrollador (Yo)	El menú funciona correctamente en el dispositivo, pero las dimensiones hay que corregirlas.
3.	Inicio prueba de audición	Desarrollador (Yo)	Se observa que al estar desconectado los auriculares de igual forma inicia la prueba. Esto hay que corregir
4.	Configuración	Desarrollador (Yo)	En la ventana de configuración falta aún implementar los niveles de audición hasta conseguir la validación con un aparato audiométrico y por lo demás está correctamente funcionando y se puede quitar los datos de usuario y

			aumentar o disminuir las frecuencias para la prueba audiometrica.
5.	Información	Desarrollador (Yo)	Para la información es igual solo lectura y se seguirá agregando nuevos datos con información.
6.	Ayuda	Desarrollador (Yo)	La ayuda es de lectura por lo que no hubo mayores inconvenientes. Se sugieren agregar algunas funcionalidades.

Elaborado por: Autor

Las pruebas por lo general indican cuando el software está terminado, si las pruebas funcionan la codificación ha terminado.

4.5. Documentación.

Manual de la aplicación de audiometría tonal móvil.

Ver **Anexo. H**

**CAPITULO V.
VALIDACIÓN**

5.1. Pruebas realizadas.

Se realizaron pruebas para comparar el funcionamiento del aparato audiométrico tonal móvil vs el aparato convencional de audiometría.

La prueba inicia con la inspección de los oídos de cada paciente, si es necesario se realiza una limpieza.

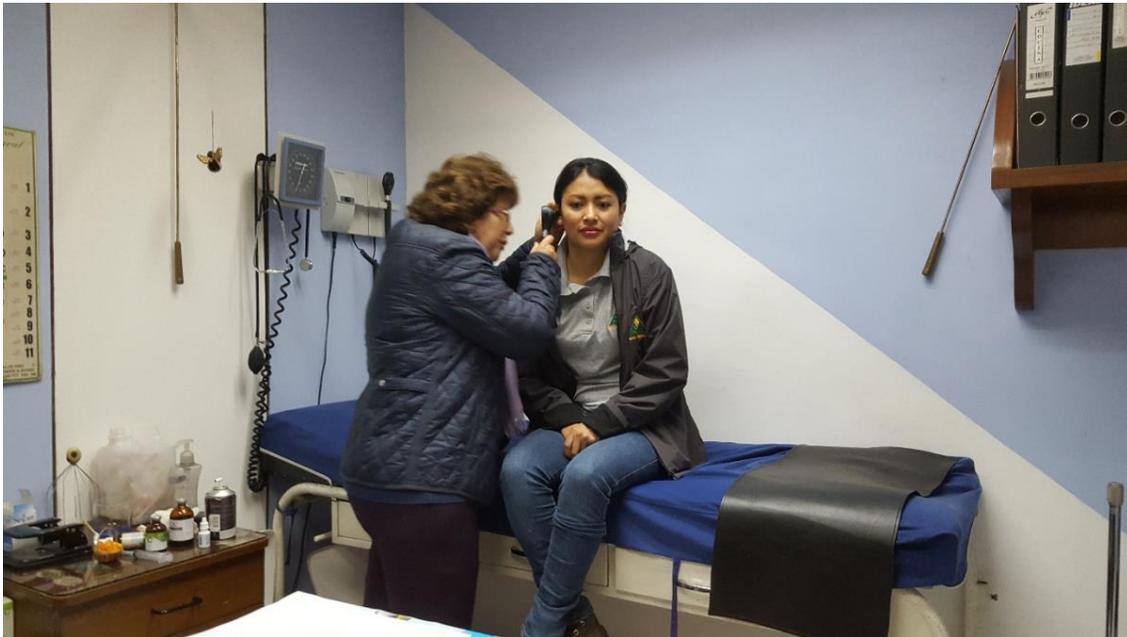


Figura 40. Limpieza auditiva
Elaborado por: Autor

En una cabina insonorizada se realiza la prueba de audiometría en el móvil y luego con el aparato de audiometría convencional.

Para la validación de la aplicación se realizaron pruebas de audiometría en siete pacientes, con la aplicación móvil vs el audiómetro convencional, obteniendo los diagramas de audiometría de tonos de ambos oídos y la identificación de los niveles de pérdida de audición. Ver **Anexo I**.

Los resultados globales se muestran en la **Tabla 16**.

Tabla 16. Resultados prueba de audiometría

Equipo	Paciente	1		2		3		4		5		6		7	
		Oído	Derecho	Izquierdo	Derecho	Izquierdo	Derecho	Izquierdo	Derecho	Izquierdo	Derecho	Izquierdo	Derecho	Izquierdo	Derecho
Dispositivo Convencional	Frecuencia, Hz	dB	dB	dB	dB	dB	dB	dB	dB	dB	dB	dB	dB	dB	dB
	125	15	15	20	20	10	20	15	10	20	15	40	25	15	10
	500	10	10	45	40	10	15	10	5	20	15	20	10	10	10
	1000	5	10	55	75	10	10	5	5	15	10	10	10	5	5
	2000	15	10	85	75	5	5	5	0	15	10	10	10	15	5
	4000	15	15	75	65	5	5	5	0	10	10	15	75	10	10
	6000	15	20	80	65	5	5	5	5	5	5	20	75	10	10
	8000	15	20	70	45	0	0	0	0	20	5	15	100	10	10
	Nivel de audición	Sin pérdida	Sin pérdida	Perdida Severa	Perdida Severa	Sin pérdida	Sin pérdida	Sin pérdida	Sin pérdida	Sin pérdida	Sin pérdida	Sin pérdida	Perdida Profunda	Sin pérdida	Sin pérdida
Aplicación Móvil	125	15	15	10	15	5	5	15	20	5	5	20	35	5	5
	500	10	10	40	30	5	5	10	10	10	5	10	20	5	5
	1000	5	5	75	75	5	5	10	10	5	5	5	15	5	5
	2000	5	5	80	80	5	5	5	5	5	5	15	25	5	5
	4000	10	10	80	80	5	5	5	10	5	5	30	30	5	5
	6000	20	15	75	75	5	5	15	10	5	5	25	75	5	10
	8000	10	15	75	75	5	5	15	10	20	10	30	75	15	10
		Nivel de audición	Sin pérdida	Sin pérdida	Perdida Severa	Perdida Severa	Sin pérdida	Perdida Severa	Sin pérdida						

Elaborado por: Autor

En las siguientes gráficas se representan los diagramas de audiometría de la aplicación móvil vs el dispositivo de audiometría convencional de cada paciente:

Paciente No. 1

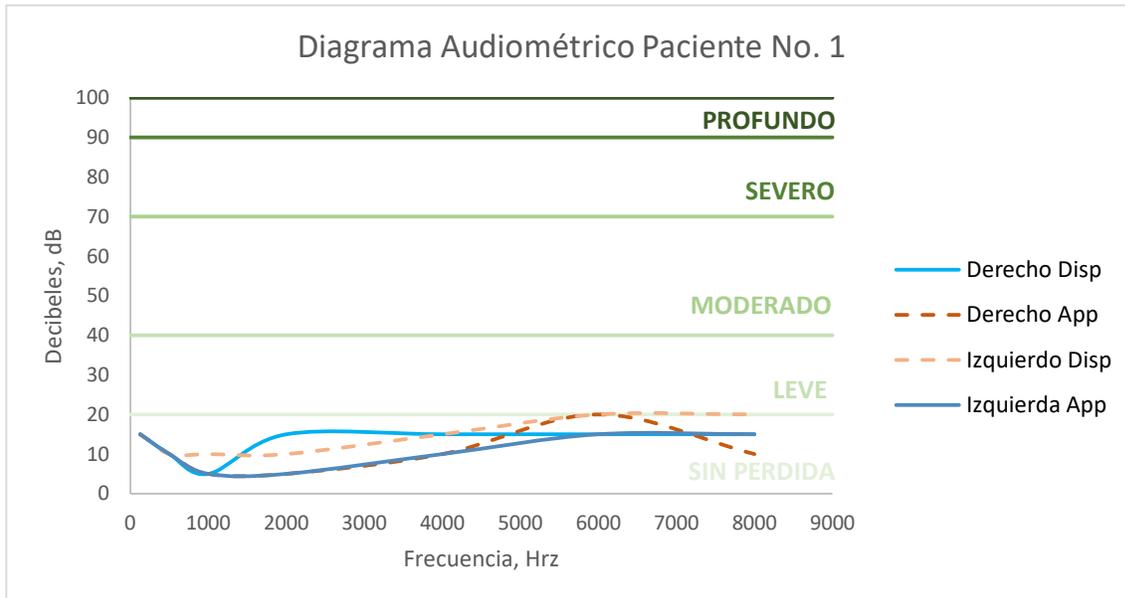


Figura 41. Resultados del paciente 1
Elaborado por: Autor

Según la **Figura. 42**, se observa que el paciente No. 1 no tiene pérdida de audición en ningún oído, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias agudas de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

Paciente No. 2

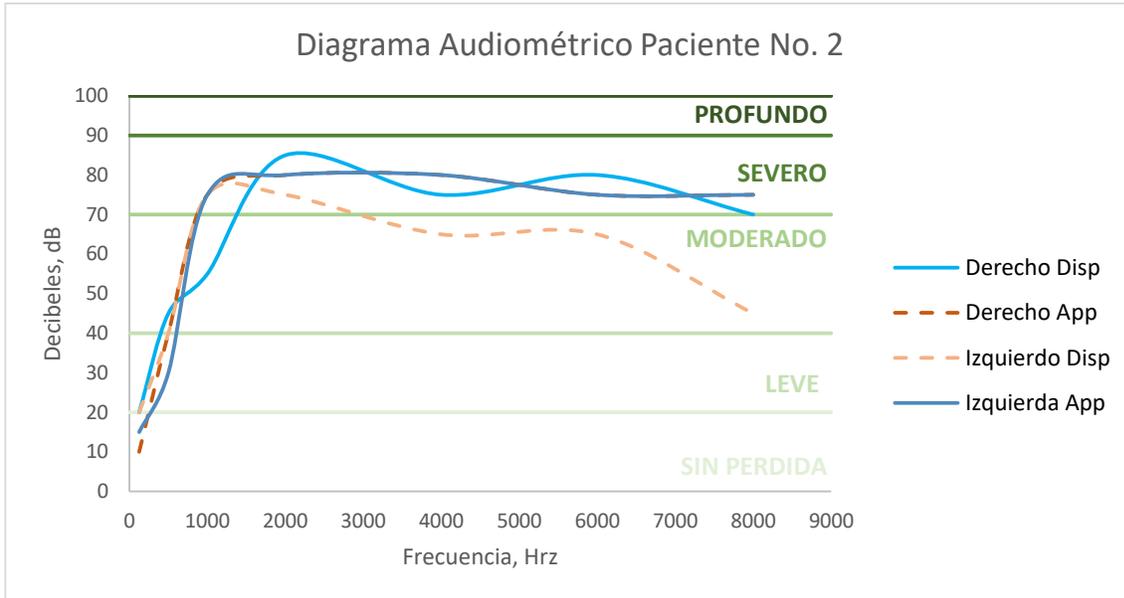


Figura 42. Resultados del paciente 2
Elaborado por: Autor

Según la **Figura. 43**, se observa que el paciente No. 2 tiene una pérdida de audición severa en ambos oídos para las frecuencias agudas, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias agudas y graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

Paciente No. 3

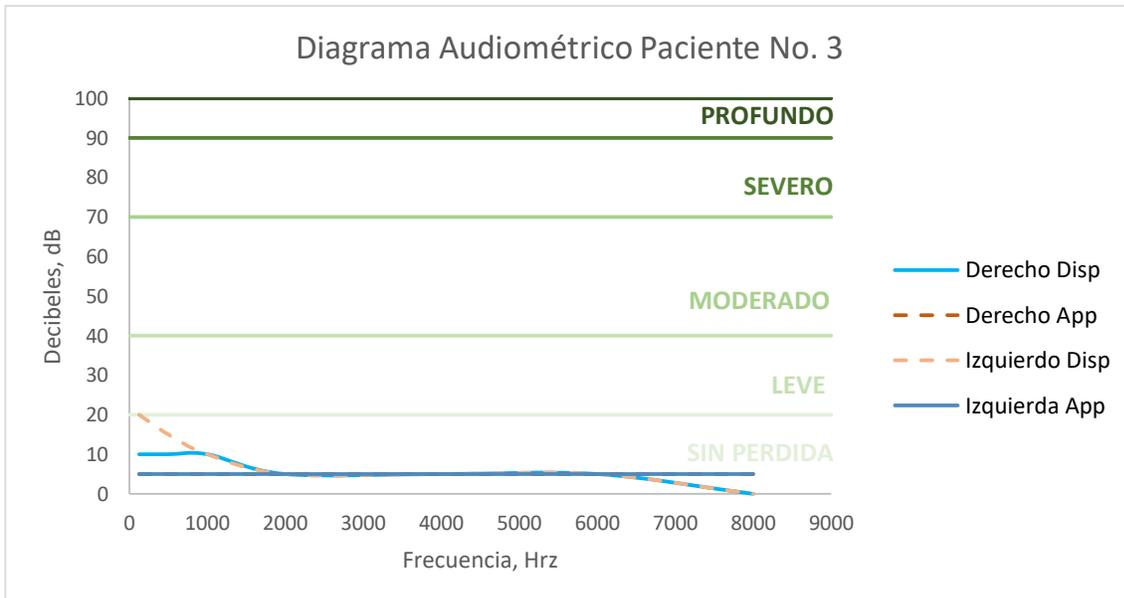


Figura 43. Resultados del paciente 3
Elaborado por: Autor

Según la **Figura. 44**, se observa que el paciente No. 3 no tiene una pérdida de audición en ambos oídos, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias agudas y graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

Paciente No. 4

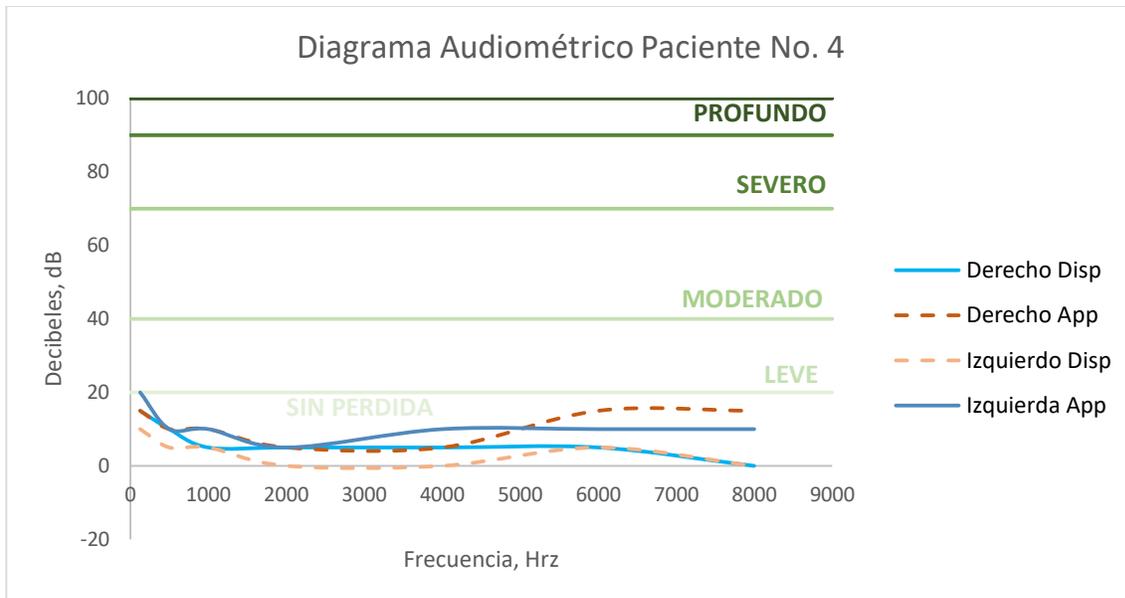


Figura 44. Resultados del paciente 4
Elaborado por: Autor

Según la **Figura. 45**, se observa que el paciente No. 4 no tiene una pérdida de audición en ambos oídos, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

Paciente No. 5

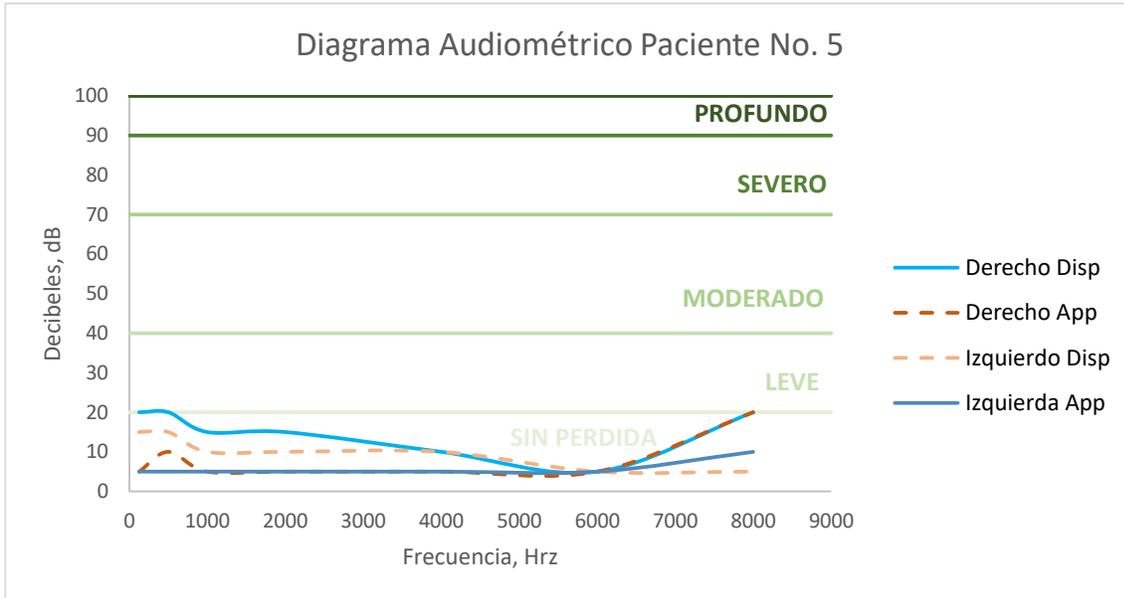


Figura 45. Resultados del paciente 5
Elaborado por: Autor

Según la **Figura. 46**, se observa que el paciente No. 5 no tiene una pérdida de audición en ambos oídos, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

Paciente No. 6

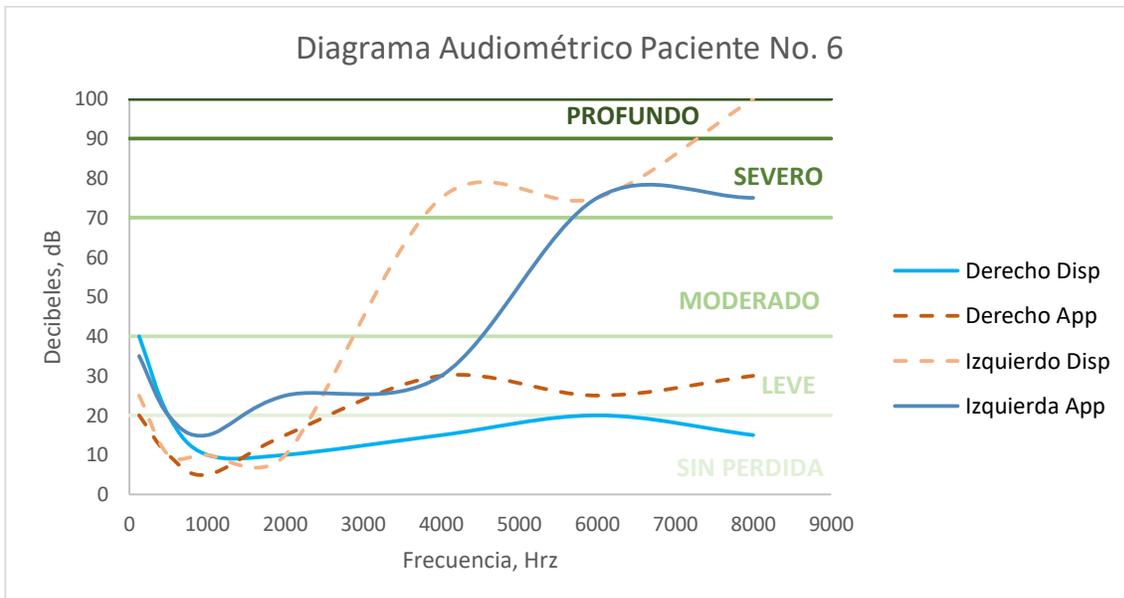


Figura 46. Resultados del paciente 6
Elaborado por: Autor

Según la **Figura. 47**, se observa que el paciente No. 6 tiene una pérdida de audición severa en el oído izquierdo y el oído derecho no tiene pérdida de audición, esto se refleja en la aplicación móvil, en la gráfica del audiómetro convencional se tiene una pérdida profunda para el oído izquierdo y para el oído derecho no tiene pérdida de audición, en las frecuencias graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, estas variaciones si afectan al resultado final.

Paciente No. 7

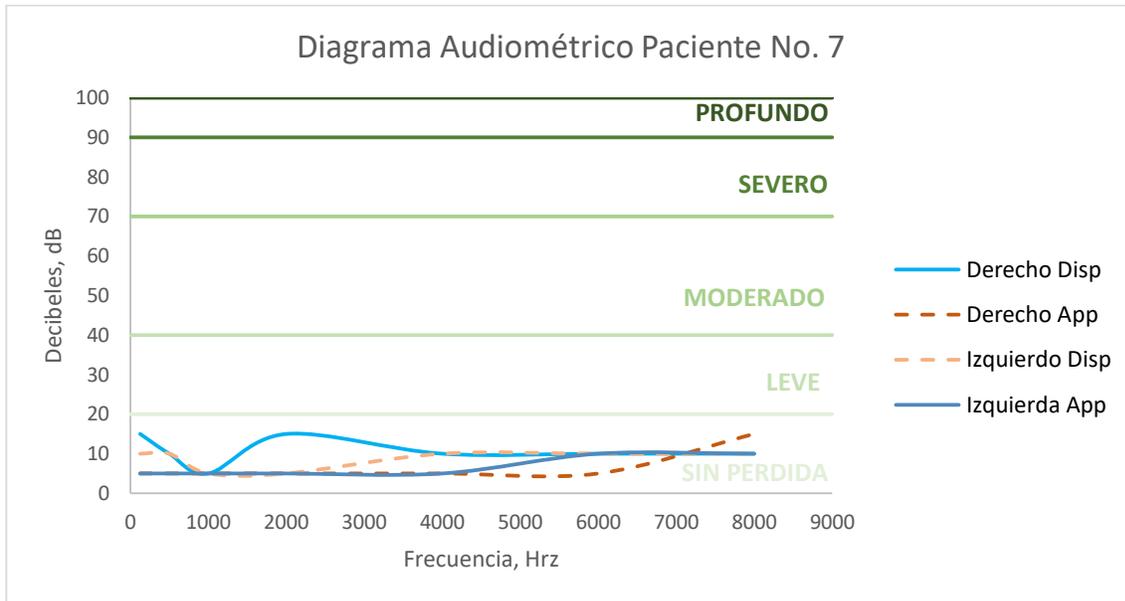


Figura 47. Resultados del paciente 7
Elaborado por: Autor

Según la **Figura. 48** se observa que el paciente No. 7 no tiene una pérdida de audición en ambos oídos, esto se refleja tanto en la gráfica de la aplicación móvil como en la del audiómetro convencional, en las frecuencias graves de la aplicación móvil se puede notar ligeras variaciones de los decibeles en cuanto al audiómetro convencional, pero estas variaciones no afectan al resultado final.

CONCLUSIONES

A continuación, expondré los resultados que se consiguieron al desarrollar el trabajo de titulación.

- Se pueden identificar 4 distintos problemas de audición, dependiendo el nivel en que se encuentren los resultados y con la realización de un promedio de estos, se determina el nivel de pérdida de audición en el que se encuentra el usuario.
- El principio del equipo de audiometría es generar tonos puros en base a distintas frecuencias, las cuales se aumentan o disminuyen en decibeles, para poder determinar la pérdida de audición del paciente.
- Con el uso de la metodología de desarrollo XP, el uso de patrones de diseño, el modelado 4 + 1 vistas y el uso de diagramas UML se obtuvo una representación gráfica y estructurada de como será la aplicación. Posteriormente, fue desarrollada mediante la aplicación Android Studio. Por otro lado, con el uso correcto de librerías y utilidades se obtuvo una aplicación para realizar una prueba de audiometría de tonos, obteniendo como resultado final un diagrama audiométrico que identifica el nivel de pérdida de audición de cada usuario.
- La aplicación de audiometría posee igual funcionamiento que un aparato de audiometría tradicional ya que fue desarrollado de la misma manera como estos funcionan, pero con las ventajas que, al ser una aplicación móvil, se cuenta con datos del paciente y permite guardar información de las pruebas realizadas.
- La implementación en los distintos dispositivos virtuales de la aplicación de audiometría permitió obtener datos de funcionamiento para una revisión y corrección de problemas de funcionamiento; mismos que fueron resueltos y permiten su adecuada funcionalidad en los dispositivos implantados.
- Según las pruebas realizadas el comportamiento de la aplicación fue satisfactorio, ya que se obtuvo diagramas de audiometría con no más del 26% de variación en más o en menos con respecto al aparato audiométrico convencional.

RECOMENDACIONES

- Es necesario realizar una validación con un número mayor de personas a partir de una muestra representativa determinada estadísticamente, en donde se realice la prueba de audiometría a personas con niveles normales de audición, con la aplicación móvil y un audiómetro convencional, con el objeto de determinar el nivel de error que exista entre los resultados obtenidos, de existir error.
- Se podría realizar la determinación del porcentaje de error de los decibeles entre la aplicación móvil vs el audiómetro convencional, para posteriormente calibrar a la aplicación. La calibración consiste en la colocación de los rangos idóneos de pérdida de audición en la aplicación, esto se realiza mediante la modificación de los parámetros que se encuentran en la opción configuración de la aplicación.
- Por el costo de las pruebas de audiometría para la validación, se debería realizar este proyecto como una investigación.
- Para poder usar la aplicación comercialmente y permitir el acceso de la mayor cantidad de personas posibles, sería necesario desarrollarla para otros sistemas operativos móviles.

BIBLIOGRAFIA

- Avison, D. E., & Fitzgerald, G. (Guy). (2006). *Information systems development : methodologies, techniques & tools*. McGraw-Hill. Retrieved from https://books.google.com.ec/books?id=xvFZAQAACAAJ&source=gbs_book_other_versions
- Ayala, J. M. (2014). *Arquitectura de aplicaciones Web-Consideraciones de Diseño*. Retrieved from <http://jmaw.blogspot.com/2013/01/arquitectura-de-aplicaciones-web.html>
- Bruce Blaus. (n.d.). Blausen 0329 EarAnatomy InternalEar-es - Oído interno - Wikipedia, la enciclopedia libre. Retrieved November 26, 2017, from https://es.wikipedia.org/wiki/Oído_interno#/media/File:Blausen_0329_EarAnatomy_InternalEar-es.png
- Cortés Aguilera, A. J., Enciso Higuera, J., & Reyes González, C. M. (2012). La audiometría de tonos puros por conducción aérea en la consulta de enfermería del trabajo. *Medicina Y Seguridad Del Trabajo*, 58(227), 136–147. <https://doi.org/10.4321/S0465-546X2012000200007>
- David Ramirez Torres. (2017). Comparativa entre XP y SCRUM | Métodos Ágiles de Programación y Soporte de Software. Retrieved April 6, 2018, from <http://davidrmetodosagiles.blogspot.com/2017/02/comparativa-entre-xp-y-scrum.html>
- Estudiantes: Isidro Pulido y David Rivas, & Asesores: Servín Rivas y Rabadán Malda. (2010). Instituto Politecnico Estudio de la perdida de audición por el uso de reproductores portatiles de audio. Retrieved from <http://tesis.ipn.mx/jspui/bitstream/123456789/11683/1/2.pdf>
- Extreme Programming: A Gentle Introduction. (2013). Retrieved January 4, 2018, from <http://www.extremeprogramming.org/>
- Gerardo Armando Esparza Mendez. (2013). Ensayos de Medicina, otras cosas y la vida: Pruebas de Función Auditiva. Retrieved March 14, 2018, from <http://aprendicesdemedicina.blogspot.com/2013/05/pruebas-de-funcion-auditiva.html>
- GILBER ARANGUREN. (n.d.). Comparación de los patrones de arquitectura MVC, MV-VM, MVP | Tópicos generales de Ingeniería de Software. Retrieved January 16, 2018, from

- <https://ingsoftwarei2014.wordpress.com/category/comparacion-de-los-patrones-de-arquitectura-mvc-mv-vm-mvp/>
- Henry Vandyke Carter. (1853). Archivo: Gray907.png - Wikimedia Commons. Retrieved November 26, 2017, from <https://commons.wikimedia.org/wiki/File:Gray907.png>
- Itectu, R. C. H. (n.d.). *MICROSOFT A P P L I C A T I O N A R C H I T E C T U R E*.
- Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. *IEEE Software*, 12(6), 42–50. <https://doi.org/10.1145/216591.216611>
- Masalski, M., Kipiński, L., Grysiński, T., & Kręcicki, T. (2016). Hearing Tests on Mobile Devices: Evaluation of the Reference Sound Level by Means of Biological Calibration. *Journal of Medical Internet Research*, 18(5), e130. <https://doi.org/10.2196/jmir.4987>
- OMS | Sordera y pérdida de la audición. (2017). *WHO*. Retrieved from <http://www.who.int/mediacentre/factsheets/fs300/es/>
- Patrón MVP - Blog de Imaginanet. Artículos especializados en soluciones y desarrollo web. (2012). Retrieved January 16, 2018, from <https://www.imaginanet.com/blog/patron-mvp.html>
- rebaza, G. gastañadui. (n.d.). Metodologías-Agiles. Retrieved from <https://www.academia.edu/16347741/Metodologias-Agiles>
- Ronald Furet H. (2013). Las Curvas Isofónicas de Fletcher y Munson. Retrieved November 26, 2017, from <https://www.bunker-audio.com/bunker-audio-portal-sonido-documentos.php?id=8>
- Scrum, M., Manuel, A., Gallego, T., Ana, C., & Domingo, C. (n.d.). Metodología Scrum . □.
- Sebastián, G. de. (1997). *Audiología práctica*. Médica Panamericana. Retrieved from https://books.google.cl/books?id=xchNAAAACAAJ&hl=es&source=gbs_navlinks_s
- Seto, W. W., & Sarmiento Rodríguez, G. (1973). *Teoría y problemas de acústica*. México: Mcgraw-hill. Retrieved from <http://www.worldcat.org/title/teoria-y-problemas-de-acustica/oclc/644328676>
- Sillitti, A. (2011). *Agile processes in software engineering and extreme programming : 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011, Proceedings*. Springer. Retrieved from https://books.google.com.ec/books?id=d7WqCAAQBAJ&dq=K.+B.+w.+C.+Andres,+Extreme+Programming+Explained,+2004.&hl=es&source=gbs_navlinks_s
- Taladro, M. (2012). Audiograma de sonidos familiares ¿ Qué puede escuchar mi hijo ?, (2002), 1–2.
- Tomy Rhymond. (2011). MVC, MVP and MVVM – Tomy’s Blog. Retrieved January 16, 2018, from <https://tomyrhymond.wordpress.com/2011/09/16/mvc-mvp-and-mvvm/>
- Vique, R. R. (2010). Métodos para el desarrollo de aplicaciones móviles. *Universidad Oberta*

- de Catalunya, 66.
- Yoandy Campos Diaz. (2008). Estilos Arquitectónicos. Retrieved April 5, 2018, from <http://estilosarquitectonicos.blogspot.com/>
- Avison, D. E., & Fitzgerald, G. (Guy). (2006). *Information systems development : methodologies, techniques & tools*. McGraw-Hill. Retrieved from https://books.google.com.ec/books?id=xvFZAQAACAAJ&source=gbs_book_other_versions
- Ayala, J. M. (2014). *Arquitectura de aplicaciones Web-Consideraciones de Diseño*. Retrieved from <http://jmaw.blogspot.com/2013/01/arquitectura-de-aplicaciones-web.html>
- Bruce Blaus. (n.d.). Blausen 0329 EarAnatomy InternalEar-es - Oído interno - Wikipedia, la enciclopedia libre. Retrieved November 26, 2017, from https://es.wikipedia.org/wiki/Oído_interno#/media/File:Blausen_0329_EarAnatomy_InternalEar-es.png
- Cortés Aguilera, A. J., Enciso Higuera, J., & Reyes González, C. M. (2012). La audiometría de tonos puros por conducción aérea en la consulta de enfermería del trabajo. *Medicina Y Seguridad Del Trabajo*, 58(227), 136–147. <https://doi.org/10.4321/S0465-546X2012000200007>
- David Ramirez Torres. (2017). Comparativa entre XP y SCRUM | Métodos Ágiles de Programación y Soporte de Software. Retrieved April 6, 2018, from <http://davidrmetodosagiles.blogspot.com/2017/02/comparativa-entre-xp-y-scrum.html>
- Estudiantes: Isidro Pulido y David Rivas, & Asesores: Servín Rivas y Rabadán Malda. (2010). Instituto Politecnico Estudio de la perdida de audición por el uso de reproductores portatiles de audio. Retrieved from <http://tesis.ipn.mx/jspui/bitstream/123456789/11683/1/2.pdf>
- Extreme Programming: A Gentle Introduction. (2013). Retrieved January 4, 2018, from <http://www.extremeprogramming.org/>
- Gerardo Armando Esparza Mendez. (2013). Ensayos de Medicina, otras cosas y la vida: Pruebas de Función Auditiva. Retrieved March 14, 2018, from <http://aprendicesdemedicina.blogspot.com/2013/05/pruebas-de-funcion-auditiva.html>
- GILBER ARANGUREN. (n.d.). Comparación de los patrones de arquitectura MVC, MV-VM, MVP | Tópicos generales de Ingeniería de Software. Retrieved January 16, 2018, from <https://ingsoftwarei2014.wordpress.com/category/comparacion-de-los-patrones-de-arquitectura-mvc-mv-vm-mvp/>
- Henry Vandyke Carter. (1853). Archivo: Gray907.png - Wikimedia Commons. Retrieved November 26, 2017, from <https://commons.wikimedia.org/wiki/File:Gray907.png>
- Ilectu, R. C. H. (n.d.). *MICROSOFT A P P L I C A T I O N A R C H I T E C T U R E*.

- Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. *IEEE Software*, 12(6), 42–50. <https://doi.org/10.1145/216591.216611>
- Masalski, M., Kipiński, L., Grysiński, T., & Kręcicki, T. (2016). Hearing Tests on Mobile Devices: Evaluation of the Reference Sound Level by Means of Biological Calibration. *Journal of Medical Internet Research*, 18(5), e130. <https://doi.org/10.2196/jmir.4987>
- OMS | Sordera y pérdida de la audición. (2017). WHO. Retrieved from <http://www.who.int/mediacentre/factsheets/fs300/es/>
- Patrón MVP - Blog de Imaginanet. Artículos especializados en soluciones y desarrollo web. (2012). Retrieved January 16, 2018, from <https://www.imaginanet.com/blog/patron-mvp.html>
- rebaza, G. gastañadui. (n.d.). Metodologías-Agiles. Retrieved from <https://www.academia.edu/16347741/Metodologias-Agiles>
- Ronald Furet H. (2013). Las Curvas Isofónicas de Fletcher y Munson. Retrieved November 26, 2017, from <https://www.bunker-audio.com/bunker-audio-portal-sonido-documentos.php?id=8>
- Scrum, M., Manuel, A., Gallego, T., Ana, C., & Domingo, C. (n.d.). Metodología Scrum . □.
- Sebastián, G. de. (1997). *Audiología práctica*. Médica Panamericana. Retrieved from https://books.google.cl/books?id=xchNAAAACAAJ&hl=es&source=gbs_navlinks_s
- Seto, W. W., & Sarmiento Rodríguez, G. (1973). *Teoría y problemas de acústica*. México: Mcgraw-hill. Retrieved from <http://www.worldcat.org/title/teoria-y-problemas-de-acustica/oclc/644328676>
- Sillitti, A. (2011). *Agile processes in software engineering and extreme programming : 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011, Proceedings*. Springer. Retrieved from https://books.google.com.ec/books?id=d7WqCAAQBAJ&dq=K.+B.+w.+C.+Andres,+Extreme+Programming+Explained,+2004.&hl=es&source=gbs_navlinks_s
- Taladro, M. (2012). Audiograma de sonidos familiares ¿ Qué puede escuchar mi hijo ?, (2002), 1–2.
- Tomy Rhymond. (2011). MVC, MVP and MVVM – Tomy's Blog. Retrieved January 16, 2018, from <https://tomyrhymond.wordpress.com/2011/09/16/mvc-mvp-and-mvvm/>
- Vique, R. R. (2010). Métodos para el desarrollo de aplicaciones móviles. *Universidad Oberta de Catalunya*, 66.
- Yoandy Campos Diaz. (2008). Estilos Arquitectónicos. Retrieved April 5, 2018, from <http://estilosarquitectonicos.blogspot.com/>

ANEXOS

Anexo. A

ENCUESTA DIRIGIDA PARA UN ESPECIALISTA EN AUDIOMETRÍA DE TONOS

Objetivo:

Determinar el procedimiento necesario para realizar una audiometría de tonos, además de obtener del especialista los requerimientos mínimos y esenciales con el fin de realizar una aplicación de audiometría de tonos para dispositivos móviles.

Datos del Encuestado.

Nombre y Apellido: Dr. Fausto Cuello Serrano

Cargo: Decano de la Facultad de Ciencias de la Discapacidad de la Universidad Central del Ecuador.

Preguntas

1. ¿Qué se le pregunta al paciente, antes de que se realice la audiometría de tonos?

- ¿Cuál es el motivo de su visita?
- ¿Ha tenido alguna molestia en sus oídos ya sea hormigueo o ha tenido problemas para escuchar algunos sonidos, siente que su oído se encuentra tapado?
- ¿Alguna vez ha asistido a un audiólogo?
Si su respuesta es sí.
 - ¿Cuándo fue la última vez que asistió a un audiólogo?
 - ¿Cuál fue su motivo?

2. ¿Cuáles son los pasos iniciales para realizar la audiometría de tonos?

- Primeramente, se observarán con el otoscopio el interior del conducto auditivo y el tímpano.
- Segundo se le pide al paciente que se ubique en una cabina insonorizada para la prueba audiométrica.
- Tercero se le pedirá al paciente que se coloque un tipo de auriculares.
- Cuarto se inicia con el test de audiometría.

3. ¿Qué tipos de auriculares conoce, que se usan para la audiometría de tonos?

Existen varios tipos de auriculares y su uso depende del lugar donde se valla a realizar el examen y el tipo de problema con el que cuenta el paciente.

Los tipos son los siguientes:

Circumaurales

También llamados *over-ear*, es al tipo de auriculares al que pertenecen los modelos que ofrecen una mayor calidad de sonido, ideal para disfrutar el audio a pleno en el hogar y además cuentan con almohadillas que pueden aislar parte de ruido.

- **Ventajas:** gran calidad de sonido, aislamiento de ruidos externos, cómodos.
- **Desventajas:** demasiado grandes, es la categoría más costosa de auriculares.

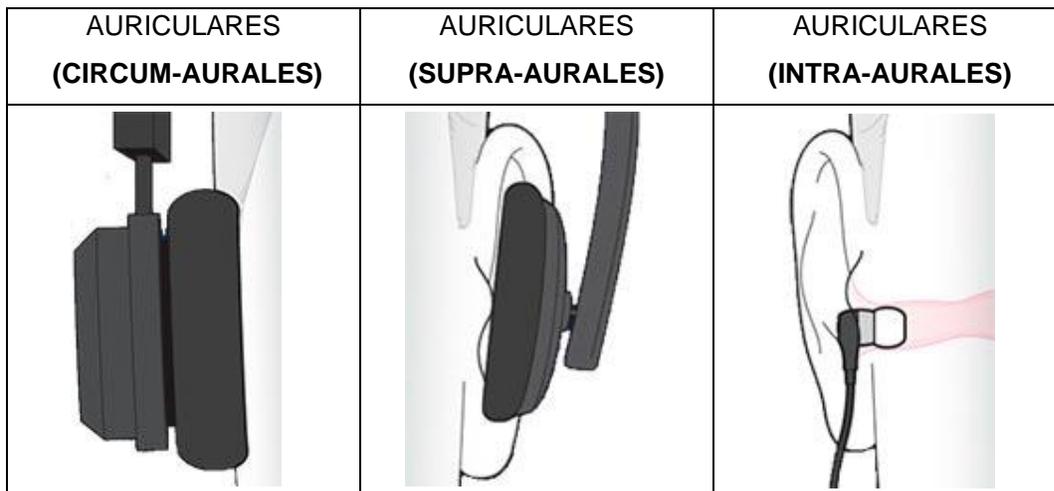
Supra aurales

Los supra aurales suelen ser la opción básica para escuchar música en el hogar. El sonido depende de su calidad, pero no son una gran opción para usar en audiometría.

Intraauriculares

También conocidos como intraauriculares, son los que cuentan con una almohadilla que es introducida dentro de la oreja, son una buena opción para escuchar mientras se viaja de un lugar a otro, debido a su calidad aceptable y buen aislamiento de ruidos gracias a un estudio realizado se sabe que pueden bajar entre 20 a 30 dB.

- **Ventajas:** pequeños, livianos, buena calidad de sonido y aislamiento de ruidos externos.
- **Desventajas:** suelen ser incómodos y su calidad de sonido está limitada por el tamaño.



4. ¿En qué lugar y horas del día es recomendable realizarse el análisis de audiometría tonal?

Le recomendaría que se lo hiciera en un lugar lo más insonorizado posible y no se tome en cuenta una hora ya que dependería la ubicación del paciente, puede estar en la ciudad o en el campo y a distintas el ruido puede ser diferente en estos lugares.

5. ¿El paciente tiene que venir preparado, para realizarse el examen de audiometría?

Para el examen, se le pregunta al paciente si en las últimas 4 horas ah estado escuchando ruidos muy agudos, por ejemplo, que se haya encontrado en un concierto o cerca de parlantes, ya que esto influiría en el resultado del examen.

6. ¿Qué frecuencias de tonos generalmente usa, para analizar al paciente y de que depende el uso de estos?

A la mayoría de los pacientes se les realiza el examen con frecuencias que van desde:

(125,250,500,750,1000,1500,2000,3000,4000,6000,8000,10000,12000,14000,16000) Hz.

Siendo habitual el uso de las siguientes frecuencias:

(250,500,1000,2000,4000,6000,8000) Hz.

7. ¿Con que frecuencia y a qué nivel de decibelios se debe iniciar el examen y por qué?

Se inicia con frecuencias graves menores e iguales a 1000Hz y valor 40 dB como inicio de todas las frecuencias y aumentando o disminuyendo en el proceso del examen, con respecto a cómo escuché el paciente, si escucha se disminuyen en 5 dB o si no escucha se aumenta en 5 dB y luego de terminar con frecuencias menores e iguales a 1000Hz se continua con las frecuencias mayores a 1000 Hz y con el mismo procedimiento de decibelios como anteriormente expliqué.

8. ¿Importa con que oído se inicie el examen de audiometría tonal?

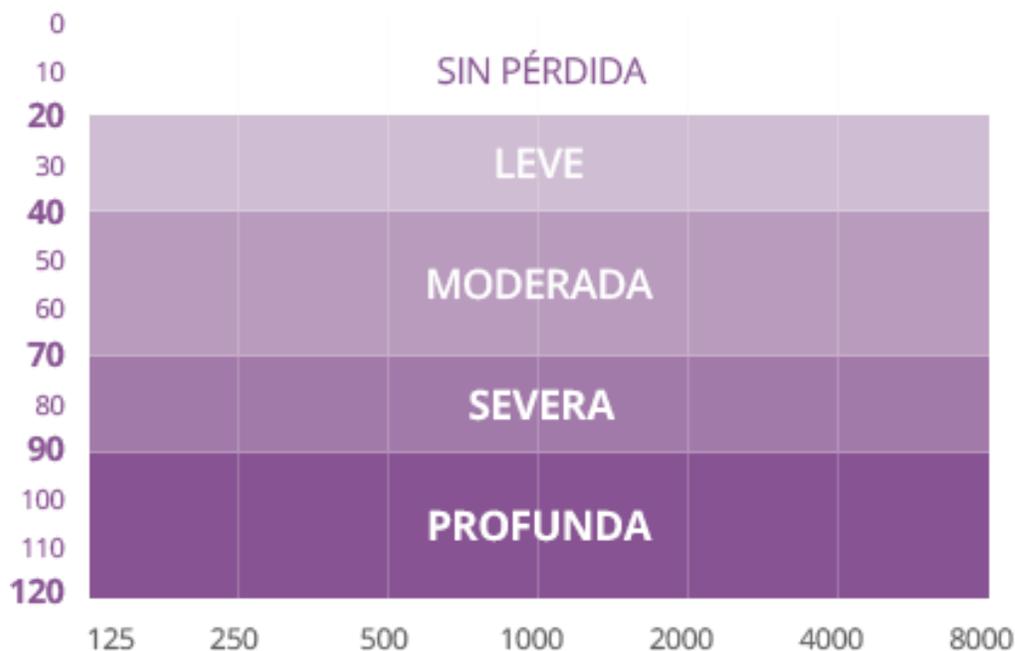
No importa, con cual se inicie el examen.

9. ¿Qué se realiza en el aparato de audiometría cuando el paciente escucha el tono y cuando no lo escucha?

Se aumenta o disminuye los decibelios con respecto a si el paciente escucha o no escucha el tono, disminuyendo en 5dB o aumentando en 5dB respectivamente o cambiando de frecuencia.

10. ¿Cómo se interpreta el diagrama audiométrico?

Se realiza la interpretación con respecto al diagrama de referencia de pérdida auditiva de adultos o ya sea de niños además el nivel con el que se haya calibrado el audiómetro.



11. ¿Los resultados están basados en tabla de referencia o algún documento?

Dependiendo de las normas de audiometría que se esté usando para su evaluación.

12. ¿Realiza alguna calibración, al equipo audiométrico?

Si se realiza una calibración debido a que son aparatos analógicos y se los calibra con respecto a los parámetros de carácter técnico vigentes internacionalmente, estos son

cumplimiento de los estándares de las normas ANSI S3.6, EN 60645-1, EN60645-2, ISO 389-1; ISO389-3, ISO389,4; ISO389-7.

¿Con que intervalo de tiempo realiza la calibración?

Cada 6 meses.

REQUERIMIENTOS PARA UNA APLICACIÓN MÓVIL DE AUDIOMETRÍA TONAL

1. ¿Alguna vez ha usado o ha escuchado sobre una aplicación de audiometría para dispositivos móviles?

No esta es la primera vez, por eso creo es un gran tema el que propone.

2. ¿Cuáles cree que serían los requisitos mínimos que debería tener una aplicación móvil, para realizar un examen audiométrico de tonos?

Una aplicación que se encuentre bien diseñada y desarrollada para que el usuario pueda manejarla de una forma fácil y sin complicaciones y cuente con toda la información necesaria de cómo realizarse el examen además si lo requiriera el usuario cuente con información de ayuda, además que se le pida como un requisito necesario antes de usar la aplicación que el dispositivo cuente con auriculares intra-aural o como mínimo de tipo circum-aurales y consejos para la utilización de unos u otros y que sean los que viene con el dispositivo del fabricante del dispositivo, ya que son de mejor calidad y como resultado final del examen sea lo más real posible al del aparato de audiometría convencional.

3. ¿Qué tipo de auricular es el más recomendable para usar?

El que recomiendo para la aplicación móvil de audiometría serían los de tipo Intra-aural, ya que un estudio realizado en la Universidad Central se demuestra que este tipo de auriculares pueden reducir el ruido externo entre 20 y 30 decibelios y serian adecuados para una aplicación móvil que se va a realizar fuera de una cabina insonorizada además de buscar otros parámetros de uso además se tendría que buscar dispositivos que cuenten de fábrica con este auricular.

Que cuenten con las frecuencias habituales mencionadas en la pregunta anterior.

Su proceso sea como el mencionado anteriormente para la evaluación del paciente con respecto a los dB.

4. ¿En su consideración como cree que debería estar diseñada la aplicación para atraer a las personas a usarla?

Primero que nada, es necesario que exista una validación de la aplicación con el dispositivo convencional de audiometría y como mínimo realizar un análisis estadístico con 250 personas que cuenten con audición normal para ir verificando los niveles en que escuchan con el aparato audiométrico y el dispositivo móvil para de esta manera obtener los niveles de audición y con estos realizar el diagrama de pérdida auditiva ya sea para niños o adultos y

con esta información ver como se promueve la aplicación y para el diseño realizarlo lo más posible a como se realiza una audiometría convencional como explicaba anteriormente.

5. ¿En su consideración para que dispositivo móvil fuese mejor realizar la aplicación, para Tablet o Smartphone o ambos?

Para mi consideración, me gustaría que funcionase en ambos dispositivos.

6. ¿Las frecuencias de audio tienen algo en especial o pueden ser generadas con cualquier programa?

Para obtener las frecuencias de audio se las puede usar mediante un algoritmo y se lo puede realizar digitalmente.

7. ¿Sería factible realizar un eliminador de enmascaramiento sonoro (eliminación de ruido externo)?

En mi consideración se lo puede realizar, pero va a ser necesario realizar pruebas muy minuciosas, que se lo podría plan tantear como un proyecto.

Anexo. B

Encuesta "Realización de una aplicación de audiometría tonal móvil"

La aplicación de audiometría móvil es una aplicación que pretende igualar la funcionalidad de un aparato audiométrico de tonos, la cual permitirá realizarse un examen de audiometría tonal y poder identificar cual es el nivel de pérdida auditiva con el que cuenta el paciente, además de proveer la información recomendada por un especialista sobre el cuidado auditivo para el cuidado de su salud. Todo esto se realizará en un dispositivo móvil que podría ser una Tablet o Smartphone y utilizando un tipo específico de auriculares.

El motivo de realizar esta encuesta es obtener información de usuarios potenciales, para desarrollar una aplicación de acuerdo con sus requerimientos. La encuesta se realizó mediante  Google Forms

ENCUESTA

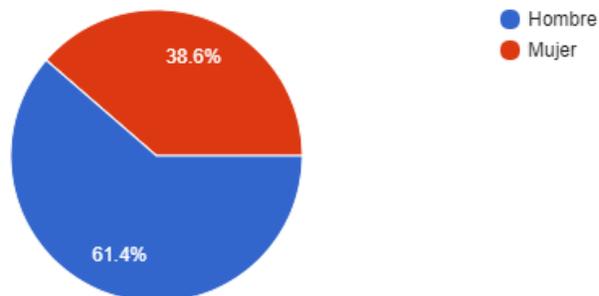
1. ¿Cuál es su sexo? Marca solo un óvalo.

- Hombre
 Mujer

Resultados

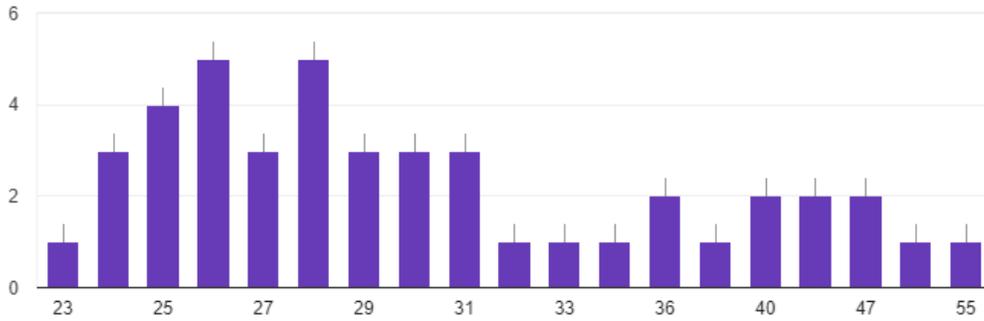
¿Cuál es su sexo?

44 respuestas



2. ¿Cuál es su edad?

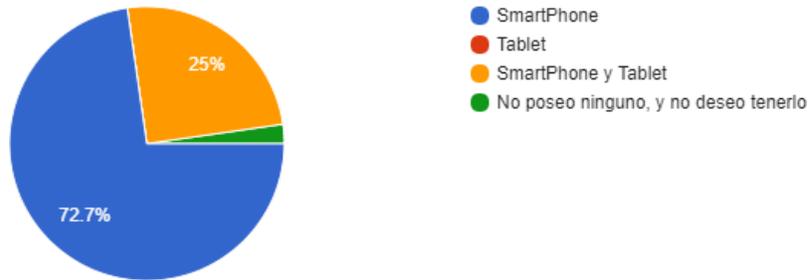
Resultados



3. ¿Cuenta usted con un teléfono celular inteligente "¿Smartphone" o una Tablet, o ambos? Marca solo un óvalo.

- Smartphone
- Tablet
- Smartphone y Tablet
- No poseo ninguno, y no deseo tenerlo

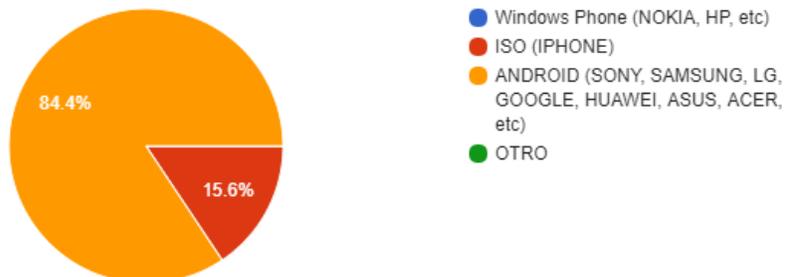
Resultados



4. ¿Qué tipo sistema operativo tiene su Smartphone o Tablet? Marca solo un óvalo.

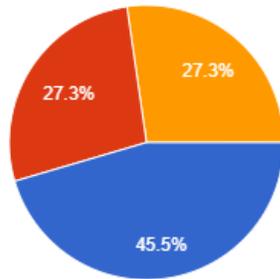
- Windows Phone (NOKIA, HP, etc.)
- ISO (IPHONE)
- ANDROID (SONY, SAMSUNG, LG, GOOGLE, HUAWEI, ASUS, ACER, etc.)
- OTRO

Resultados



5 ¿Qué tipo de auriculares utiliza? * Marca solo un óvalo.

- Intra-Aurales (Los que cuentan con una goma de silicona y se introducen el oído)
- Supra-Aurales (los que cuentan con almohadilla y no cubre toda la oreja)
- Circum-Aurales (Los que cubren toda la oreja más conocidos como cascos)
- Otro
- Resultados**

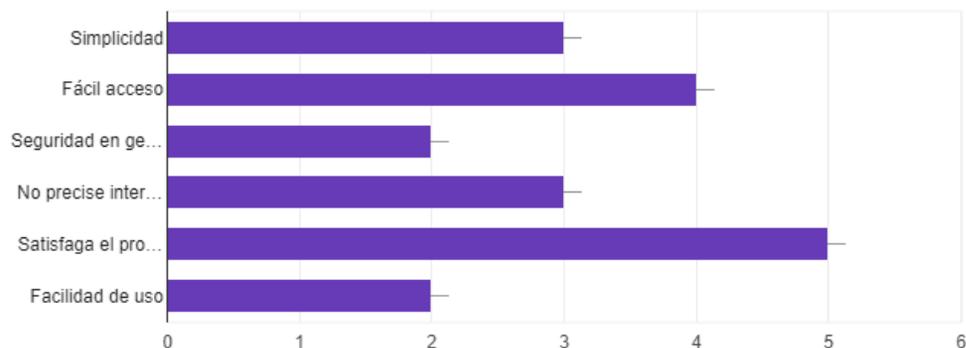


- Intra-Aurales (Los que cuentan con una goma de silicona y se introducen el oído)
- Supra-Aurales (los que cuentan con almohadilla y no cubre toda la oreja)
- Circum-Aurales (Los que cubren toda la oreja más conocidos como cascos)
- Otro

6. ¿Cuáles de las siguientes características considera usted que debe tener una aplicación de audiometría tonal móvil que pretende igualar la funcionalidad de un aparato audiométrico de tonos? Selecciona todas las opciones que correspondan.

- Simplicidad
- Fácil acceso
- Seguridad en general
- No precise internet para funcionar
- Satisfaga el problema
- Facilidad de uso

Resultados

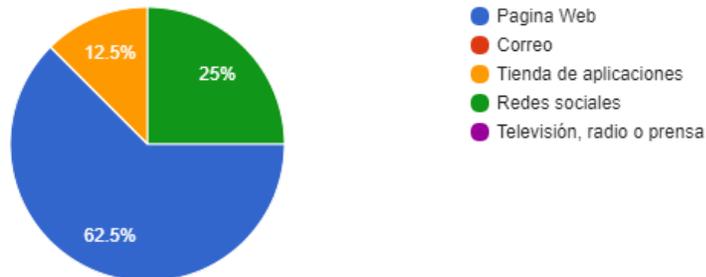


7. ¿A través de qué medios le gustaría encontrar información sobre este producto?

Marca solo un óvalo.

- Página Web
- Correo
- Tienda de aplicaciones
- Redes sociales
- Televisión, radio o prensa

Resultados



8. ¿Cuál sería su criterio para comprar una aplicación que realice una prueba auditiva?

Marca solo un óvalo.

- El historial de éxitos y fracasos que ha tenido
- La validación que se le ha dado a la aplicación.
- Qué organización o empresa la creo
- Por quién está avalada la aplicación

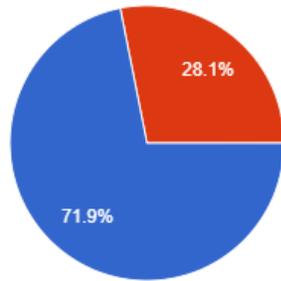
Resultados



9. ¿Preferiría que sea una aplicación de pago o gratuita? Marca solo un óvalo.

- Pago (Se da soporte a la aplicación continuamente)
- Gratuita (No se da soporte a la aplicación)

Resultados

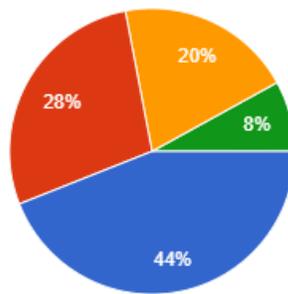


- Pago (Se da soporte a la aplicación continuamente)
- Gratuita (No se da soporte a la aplicación)

10. ¿En qué valor compraría esta aplicación? Marca solo un óvalo.

- 1 dólar
- 1.5 dólares
- 2 dólares
- 3 dólares

Resultados

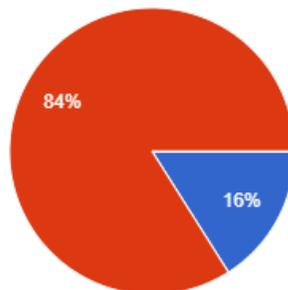


- 1 dólar
- 1.50 dólares
- 2 dólares
- 3 dólares

11. ¿De dónde acostumbra a descargar sus aplicaciones? Marca solo un óvalo.

- App Store (Apple)
- Google Play
- Windows Phone Store

Resultados

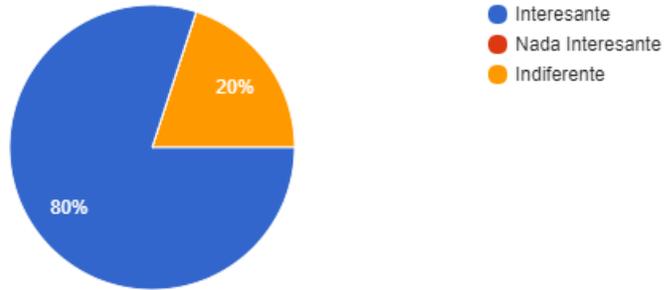


- App Store (Apple)
- Google Play
- Windows Phone Store

12. Este producto es realizado como una tesis para tercer nivel ¿Eso lo hace más, o menos interesante para usted? Marca solo un óvalo.

- Interesante
- Nada Interesante
- Indiferente

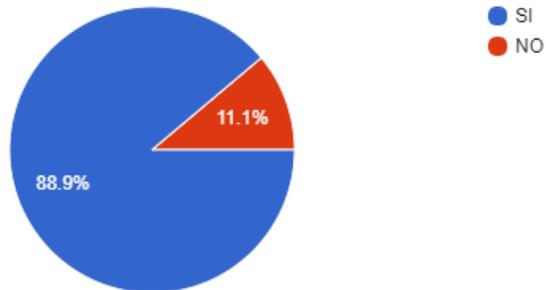
Resultados



13. ¿De realizarse esta aplicación estaría interesado(a) en adquirirla? Marca solo un óvalo.

- SI
- NO

Resultados



Anexo. C

Para el desarrollo usare el IDE (Entorno de Desarrollo Integrado) de Android Studio, el cual cuenta con las herramientas suficientes para poder realizar la codificación, el diseño de las interfaces de usuario y además cuenta a la vez con una serie de máquinas virtuales por defecto y con la posibilidad de descargar máquinas virtuales de dispositivo de distintos fabricantes. Esto me ayudara en la virtualización de la aplicación en distintos dispositivos, para verificar el diseño responsivo de las interfaces de usuario como también, para realizar las distintas pruebas ya sean funcionales o de rendimiento, además cuenta con una herramienta de monitoreo, que permite visualizar en tiempo real el uso de la memoria que tiene la aplicación en el dispositivo ya sea este virtual o real y de este modo poder optimizar la aplicación y tener un mejor rendimiento.

Instalación de la IDE de Android y el JDK (Java Development Kit)

1.) Verificar los requerimientos mínimos que necesita Android Studio para correr en tu computador, a continuación, los detallo:

Tabla 17. Requerimientos mínimos de Android Studio.

Características	Windows	Mac	Linux
Versión S. O	Windows 7/8/10 (32 o 64 bits).	Mac OS X 10.8.5 o superior.	GNOME o KDE Desktop. Ubuntu... 64 bits / 32 bits. GNU C (glibc) 2.1 o superior.
Memoria RAM	2 GB de RAM (8 GB de RAM recomendado).	2 GB de RAM (8 GB de RAM recomendado).	2 GB de RAM (8 GB de RAM recomendado).
Memoria	2 GB de espacio libre mínimo (4 GB recomendado).	2 GB de espacio libre mínimo (4 GB recomendado).	2 GB de espacio libre mínimo (4 GB recomendado).
Versión de Java JDK	JDK 8.	Java 6.	Java 8.
Procesador	64 bits procesador AMD o Intel y procesador Intel (emulador).	64 bits y procesador Intel (emulador).	64 bits y procesador Intel (emulador).

Elaborado por: Mi Autoría

2.) Instalar la versión más reciente del JDK (Java Development Kit) para poder correr Android Studio, a continuación, link para descargar:

<http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>

3.) Descargar la versión más reciente de Android Studio de su sitio Web:

<https://developer.android.com/studio/index.html>

4.) Ya descargado los componentes es solo cuestión de seguir los pasos de instalación de uno de los instaladores. Recordar, primeramente, se debe instalar el JDK (ya que, si este no es instalado primero, el instalador de Android Studio no podrá finalizar).

5.) Listo, ya solo es cuestión de iniciar el programa y empezar la programar.

Anexo. D

Descripción la clase Java Graficar.

Fragmento de código clase Graficar.java

```
public class Graficar {
    private static final String TAG = "AnalizandoActivity";

    //Agregar PointsGraphSeries al tipo DataPoint
    private PointsGraphSeries<DataPoint>
    xySeriesD,booleanxySeriesD,xySeriesI,booleanxySeriesI;
    private LineGraphSeries<DataPoint> xySeriesLineD,xySeriesLineI;

    //Haciendo de xyValueArray una variable global
    private ArrayList<XYValue> xyValueArrayD,xyValueArrayI;

    //private XYPlot plot;
    private GraphView mScatterPlot;

    public void iniciarGrafica(GraphView mScatterPlot,String [] valoresX)
    {
        //Declarando la Grafica de Dispersión
        this.mScatterPlot=mScatterPlot;
        //set Scrollable and Scaleable
        mScatterPlot.getViewPort().setScalable(true);
        mScatterPlot.getViewPort().setScalableY(true);
        mScatterPlot.getViewPort().setScrollable(true);
        mScatterPlot.getViewPort().setScrollableY(true);

        //Establecer limites de valores x manuales y en el tamaño no se
        considera el 0
        mScatterPlot.getViewPort().setXAxisBoundsManual(true);
        mScatterPlot.getViewPort().setMinX(0);
        mScatterPlot.getViewPort().setMaxX(valoresX.length-1);

        //Establecer limites de valores y manuales y en el tamaño no se
        considera el 0
        mScatterPlot.getViewPort().setYAxisBoundsManual(true);
        mScatterPlot.getViewPort().setMinY(-20);
        mScatterPlot.getViewPort().setMaxY(0);

        //Colocar Titulos en la Grafica
        mScatterPlot.setTitle("Audiometría de Tonos");
        mScatterPlot.setTitleTextSize(35);
        mScatterPlot.setTitleColor(Color.BLACK);

        mScatterPlot.getGridLabelRenderer().setHorizontalAxisTitle("Frecuencia en
        [Hz]");
        mScatterPlot.getGridLabelRenderer().setLabelHorizontalHeight(15);
        mScatterPlot.getGridLabelRenderer().setHorizontalLabelsAngle(30);

        mScatterPlot.getGridLabelRenderer().setHorizontalAxisTitleColor(Color.BLA
        CK);
        mScatterPlot.getGridLabelRenderer().setVerticalAxisTitle("Nivel
        de Audición en /[dB]");

        mScatterPlot.getGridLabelRenderer().setVerticalAxisTitleColor(Color.BLACK
        );
        mScatterPlot.getGridLabelRenderer().setGridColor(Color.BLACK);
    }
}
```

```

        mScatterPlot.getGridLabelRenderer().setHighlightZeroLines(true);
        StaticLabelsFormatter staticLabelsFormatter = new
StaticLabelsFormatter(mScatterPlot);
        staticLabelsFormatter.setHorizontalLabels(valoresX);
        staticLabelsFormatter.setVerticalLabels(new String[] {"100dB",
"90dB", "80dB", "70dB", "60dB", "50dB", "40dB", "30dB", "20dB", "10dB",
"0dB"});

mScatterPlot.getGridLabelRenderer().setLabelFormatter(staticLabelsFormatt
er);

        xyValueArrayD = new ArrayList<>();
        xyValueArrayI = new ArrayList<>();
    }

    public void removerGraficaD(int punto){
        xyValueArrayD.remove(punto);
        this.mScatterPlot.removeAllSeries();
    }

    public void initD(int xi,int yj,boolean banderanielVolumenFinal) {
        //Declarando el Objeto xySeries
        xySeriesD = new PointsGraphSeries<>();
        booleanxySeriesD= new PointsGraphSeries<>();
        xySeriesLineD = new LineGraphSeries<>();
        Log.d(TAG, "onClick: Agregando nuevos puntos. (x,y): (" + xi +
",," + yj + ","+banderanielVolumenFinal+""));
        xyValueArrayD.add(new XYValue(xi, yj,banderanielVolumenFinal));
        //Ordenar la matriz de Valores XYvalueArray
        xyValueArrayD = sortArray(xyValueArrayD);
        crearGraficaD();
    }

    private void crearGraficaD(){
        //Agregar los datos a la serie
        for(int i = 0;i <xyValueArrayD.size(); i++){
            try{
                int x = (int) xyValueArrayD.get(i).getX();
                int y = (int) xyValueArrayD.get(i).getY();

if(xyValueArrayD.get(i).getBanderanielVolumenFinal()==true){
                    booleanxySeriesD.appendData(new DataPoint(x,y),true,
20);
                }
                xySeriesD.appendData(new DataPoint(x,y),true, 20);
                xySeriesLineD.appendData(new DataPoint(x,y),true, 20);
            }catch (IllegalArgumentException e){
                Log.e(TAG, "createScatterPlot: Arguento de Excepcion
Iligal: " + e.getMessage() );
            }
        }
        //Establecer algunas propiedades
        xySeriesD.setColor(Color.RED);
        xySeriesD.setCustomShape(new PointsGraphSeries.CustomShape()
{
            @Override
            public void draw(Canvas, Paint, float x, float y,
DataPointInterface dataPoint) {
                paint.setStrokeWidth(4);
                RectF oval = new RectF();
                oval.set(x - 7, y - 7, x + 7, y + 7);
                paint.setStyle(Paint.Style.STROKE);

```

```

        canvas.drawArc(oval, 0, 360, false, paint);
    }
    });
    booleanxySeriesD.setColor(Color.RED);
    booleanxySeriesD.setCustomShape(new
PointsGraphSeries.CustomShape() {
    @Override
    public void draw(Canvas, Paint, float x, float y,
DataPointInterface dataPoint) {
        paint.setStrokeWidth(4);
        //dibujar linea
        canvas.drawLine(x - 7, y + 7, x - 14, y + 14 ,paint);
        //dibujar la punta
        canvas.drawLine(x - 14,y + 14,x - 14,y + 10,paint);
        canvas.drawLine(x - 14,y + 14,x - 10,y + 14,paint);
    }
    });
    xySeriesLineD.setColor(Color.RED);
    graficarD();
}

public void graficarD(){
    mScatterPlot.addSeries(xySeriesLineD);
    mScatterPlot.addSeries(booleanxySeriesD);
    mScatterPlot.addSeries(xySeriesD);
}

public ArrayList<XYValue> guardarAnalisisD(){
    return xyValueArrayD;
}

public void removerGraficaI(int punto){
    xyValueArrayI.remove(punto);
    this.mScatterPlot.removeAllSeries();
}

public void initI(int xi,int yj, boolean banderanielVolumenFinal) {
    //Declarando el Objeto xySeries
    xySeriesI = new PointsGraphSeries<>();
    booleanxySeriesI= new PointsGraphSeries<>();
    xySeriesLineI = new LineGraphSeries<>();
    Log.d(TAG, "onClick: Agregando nuevos puntos. (x,y): (" + xi +
    "," + yj + ")");
    xyValueArrayI.add(new XYValue(xi, yj, banderanielVolumenFinal));
    //Ordenar la matriz de Valores XYvalueArray
    xyValueArrayI = sortArray(xyValueArrayI);
    crearGraficaI();
}

private void crearGraficaI(){
    //Agregar los datos a la serie
    for(int i = 0;i <xyValueArrayI.size(); i++){
        try{
            int x = (int) xyValueArrayI.get(i).getX();
            int y = (int) xyValueArrayI.get(i).getY();

if(xyValueArrayI.get(i).getBanderanielVolumenFinal()==true) {
                booleanxySeriesI.appendData(new DataPoint(x,y), true,
20);
            }
        }
    }
}

```

```

        xySeriesI.appendData(new DataPoint(x,y), true, 20);
        xySeriesLineI.appendData(new DataPoint(x,y), true, 20);
    } catch (IllegalArgumentException e){
        Log.e(TAG, "createScatterPlot: Argumento de Excepcion
Illegal: " + e.getMessage() );
    }
}
//Establecer algunas propiedades
xySeriesI.setColor(Color.BLUE);
xySeriesI.setCustomShape(new PointsGraphSeries.CustomShape() {
    @Override
    public void draw(Canvas canvas, Paint paint, float x,
float y, DataPointInterface dataPoint) {
        paint.setStrokeWidth(4);
        canvas.drawLine(x - 7, y - 7, x + 7, y + 7, paint);
        canvas.drawLine(x + 7, y - 7, x - 7, y + 7, paint);
    }
});
booleanxySeriesI.setColor(Color.BLUE);
booleanxySeriesI.setCustomShape(new
PointsGraphSeries.CustomShape() {
    @Override
    public void draw(Canvas canvas, Paint paint, float x,
float y, DataPointInterface dataPoint) {
        paint.setStrokeWidth(4);
        canvas.drawLine(x - 7, y - 7, x + 14, y + 14, paint);
        //dibujar la punta
        canvas.drawLine(x+14,y+14,x+14,y+10,paint);
        canvas.drawLine(x+14,y+14,x+10,y+14,paint);
    }
});
xySeriesLineI.setColor(Color.BLUE);
graficarI();
}

public void graficarI(){
    mScatterPlot.addSeries(xySeriesLineI);
    mScatterPlot.addSeries(booleanxySeriesI);
    mScatterPlot.addSeries(xySeriesI);
}

public void graficarResultado(ArrayList<XYValue>
xyValueArrayI,ArrayList<XYValue> xyValueArrayD){
    this.mScatterPlot.removeAllSeries();
    //Graficar linea de oido Derecho
    xySeriesD = new PointsGraphSeries<>();
    booleanxySeriesD= new PointsGraphSeries<>();
    xySeriesLineD = new LineGraphSeries<>();
    this.xyValueArrayD=xyValueArrayD;
    crearGraficaD();

    //Graficar linea de oido Izquierdo
    xySeriesI = new PointsGraphSeries<>();
    booleanxySeriesI= new PointsGraphSeries<>();
    xySeriesLineI = new LineGraphSeries<>();
    this.xyValueArrayI=xyValueArrayI;
    crearGraficaI();
}

public ArrayList<XYValue> guardarAnalisisI(){

```

```

        return xyValueArrayI;
    }

    //Ordenar el ArrayList <XYValue> con respecto a los valores de x.
    private ArrayList<XYValue> sortArray(ArrayList<XYValue> array){
        /*
        //Ordena los xyValues en orden ascendente para prepararlos para
        el PointsGraphSeries <DataSet>
        */
        int factor =
Integer.parseInt(String.valueOf(Math.round(Math.pow(array.size(),2))));
        int m = array.size() - 1;
        int count = 0;
        Log.d(TAG, "Clasificacion del Array: Clasificación del
XYArray.");
        while (true) {
            m--;
            if (m <= 0) {
                m = array.size() - 1;
            }
            Log.d(TAG, "Clasificacion del Array: m = " + m);
            try {
                //print out the y entrys so we know what the order looks
like
                //Log.d(TAG, "sortArray: Order:");
                //for(int n = 0;n < array.size();n++){
                //Log.d(TAG, "sortArray: " + array.get(n).getY());
                //}
                int tempY = (int) array.get(m - 1).getY();
                int tempX = (int) array.get(m - 1).getX();
                boolean tempBanderanivelVolumenFinal=array.get(m -
1).getBanderanivelVolumenFinal();
                if (tempX > array.get(m).getX()) {
                    array.get(m - 1).setY(array.get(m).getY());
                    array.get(m).setY(tempY);
                    array.get(m - 1).setX(array.get(m).getX());
                    array.get(m).setX(tempX);
                    array.get(m -
1).setBanderanivelVolumenFinal(array.get(m).getBanderanivelVolumenFinal()
);
array.get(m).setBanderanivelVolumenFinal(tempBanderanivelVolumenFinal);
                } else if (tempX == array.get(m).getX()) {
                    count++;
                    Log.d(TAG, "Clasificacion del Array: count = " +
count);
                } else if (array.get(m).getX() > array.get(m - 1).getX())
{
                    count++;
                    Log.d(TAG, "Clasificacion del Array: count = " +
count);
                }
                //break when factorial is done
                if (count == factor) {
                    break;
                }
            } catch (ArrayIndexOutOfBoundsException e) {
                Log.e(TAG, "Clasificacion del Array:
ArrayIndexOutOfBoundsException. Necesita más de 1 punto de datos para
crear Trazado." +
                    e.getMessage());
            }
        }
    }

```

```
        break;
    }
    return array;
}
```

Descripción: Esta clase se encarga:

- Realizar la gráfica con los puntos enviados a esta clase.
- Ordenas las coordenadas para poder graficarlas.
- Crear la gráfica para el oído izquierdo o derecho
- Graficar los distintos gráficos de terminación para oído derecho o izquierdo
- Eliminar la gráfica del oído izquierdo derecho.

Anexo. E

Descripción de la clase Java Prueba Auditiva.

Fragmento de código clase PruebaAuditiva.java

```
public class AnalizandoActivity extends AppCompatActivity implements
View.OnClickListener {

    private Toast toast;
    private GraphView mScatterPlot;
    private Button btnNoPuedoOir, btnPuedoOir, btnApenasPuedoOir;
    private RadioGroup radiogrupoizde;
    private RadioButton radioButtonDerecho, radioButtonIzquierdo;

    //Declarando la clase Graficar
    private Graficar graficar;
    private int contNivelVolumen
=1000, contID, contActivacionRadioButtonI=0,
        contActivacionRadioButtonD=0, valueX, valueY;
    private boolean
banderaOidoD=false, banderaOidoI=false, banderaID=false,
        banderaniivelVolumenFinal=false, banderaPasoOido=false,
banderabntNoPuedoOir=false;

    //Deteccion de Auriculares y Desconexion con pausa de audio
    private ConexionAuricularIntentReceiver conexionauricular;
    private IntentFilter intentFilter = new
IntentFilter(AudioManager.ACTION_AUDIO_BECOMING_NOISY);

    //Reproductor de tonos
    private SoundPool sp;

    //Declaracion de variables para inicializacion en onResume
    private int tamañoFrecuenciasMenores1000Analizando,
enviovectorFrecuenciasActivasGrafica,
        envioparametromaximovolumenAnalizando,
tamañoFrecuenciasMenores1000Analizando2,
        tamañoFrecuenciasMenores1000Analizando3,
numeroPaciente =0;
    private ArrayList<Integer>
arrayparametrosmaximoVolumen, arrayAudiosCargados;
    private String[] frecuenciasEnviadas,
vectorarrayparametrosmaximoVolumen;
    private String frecuenciaActiva;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_analizando);
        //Declaracion de variables
        mScatterPlot = (GraphView) findViewById(R.id.graph);
        btnApenasPuedoOir = (Button)
findViewById(R.id.btnApenasPuedoOir);
        btnPuedoOir = (Button) findViewById(R.id.btnPuedoOir);
        btnNoPuedoOir = (Button) findViewById(R.id.btnNoPuedoOir);
        radiogrupoizde = (RadioGroup) findViewById(R.id.radgroup);
        radioButtonDerecho= (RadioButton) findViewById(R.id.radbDerecha);
        radioButtonIzquierdo=
(RadioButton) findViewById(R.id.radbIzquierdo);
```

```

        arrayAudiosCargados=new ArrayList<Integer>();
        //Verificar Conexion de Auriculares y pausa de musica si se los
retira
        conexionauricular = new ConexionAuricularIntentReceiver();
        /* MediaSessionCompat.Callback callback = new
        MediaSessionCompat.Callback() {
            @Override
            public void onPlay() {
                registerReceiver(conexionauricular,
intentFilter);
            }

            @Override
            public void onStop() {
                unregisterReceiver(conexionauricular);
            }
        };*/
        //Obteniendo valores de ConfiguracionActivity
        CargarConfiguracion();
        //Cargando el Array de los parametros de maximo volumen
        arrayparametrosmaximoVolumen =new ArrayList<Integer>();
        for(int i = 0; i< vectorarrayparametrosmaximoVolumen.length;
i++){
        arrayparametrosmaximoVolumen.add(Integer.parseInt(vectorarrayparametrosma
ximoVolumen[i]));
        }
        //Iniciando Grafica Layout
        graficar = new Graficar();
        graficar.iniciarGrafica(mScatterPlot,frecuenciasEnviadas);

        //Iniciando Audio
        inicializarSoundPool();

        //Mensaje de Explicativo
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Iniciando Aplicación de Audio mas
informacion.....").setTitle("Atención!!")
        .setCancelable(false)
        .setNeutralButton("Aceptar",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
int id) {
                    dialog.cancel();
                }
            });
        AlertDialog alert = builder.create();
        alert.show();

        //Escuchando interacion en pantalla por el usuario de los
RadioGroup para analizar oido derecho o izquierdo
        radiogrupoidzde.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, @IdRes
int m) {
                if(m==R.id.radbDerecha){
                    mensaje("¡INICIANDO AUDIOMETRIA TONAL\nOIDO
Derecho!");

                    //Colocar las banderas de oidoD(activa) y oidoI

```

```

(desactivada para que la próxima vez que se escoja ya no se ejecuten las
condiciones de esta bandera).
    banderaOidoD=true; banderaOidoI=false;
banderanivelVolumenFinal=false;

//Iniciar contador para el nivel del volumen iniciado
en 40dB
    contNivelVolumen =40;

//Condicion que verifica que existan frecuencias
menores a 1000 para graficar si no existieran se inicia con la primera
frecuencias
    if(tamañoFrecuenciasMenores1000Analizando==0){
        valueX=1;
    }else {
        valueX = tamañoFrecuenciasMenores1000Analizando;
    }
    valueY=-8;

//Se envia los niveles del volumen
nivelVolumen(0,0.4f);
if(banderaPasoOido==true){
    graficar.graficarI();
}

//Se Grafican los Puntos
graficar.initD(valueX,valueY,banderanivelVolumenFinal);

//Para Frecuencias menores a 1000
contID=0;
banderaID=true;

tamañoFrecuenciasMenores1000Analizando2=tamañoFrecuenciasMenores1000Anali
zando-1;

tamañoFrecuenciasMenores1000Analizando3=tamañoFrecuenciasMenores1000Anali
zando;
} else if (m==R.id.radbIzquierdo) {
    mensaje("¡INICIANDO AUDIOMETRIA TONAL\NOIDO
IZQUIERDO!");

//Colocar las banderas de oidoI(activa) y oidoD
(desactivada para que la próxima vez que se escoja ya no se ejecuten las
condiciones de esta bandera).
    banderaOidoD=false; banderaOidoI=true;
banderanivelVolumenFinal=false;

//Iniciar contador para el nivel del volumen iniciado
en 40dB
    contNivelVolumen =40;

//Condicion que verifica que existan frecuencias
menores a 1000 para graficar si no existieran se inicia con la primera
frecuencias
    if(tamañoFrecuenciasMenores1000Analizando==0){
        valueX=1;
    }else {
        valueX = tamañoFrecuenciasMenores1000Analizando;
    }
    valueY=-8;

```

```

        //Se envia los niveles del volumen
        nivelVolumen(0.4f,0);
        if(banderaPasoOido==true){
            graficar.graficarD();
        }

        //Se Grafican los Puntos

graficar.initI(valueX,valueY,banderanivelVolumenFinal);

        //Para Frecuencias menores a 1000
        contID=0;
        banderaID=true;

tamañoFrecuenciasMenores1000Analizando2=tamañoFrecuenciasMenores1000Anali
zando-1;

tamañoFrecuenciasMenores1000Analizando3=tamañoFrecuenciasMenores1000Anali
zando;
    }
    });

    //Escuchando interacion en pantalla por el usuario de los botones
    btnApenasPuedoOir.setOnClickListener(this);
    btnPuedoOir.setOnClickListener(this);
    btnNoPuedoOir.setOnClickListener(this);
}

@Override
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.btnPuedoOir:
            //Si se preciona los botones de respuesta sin antes haber
            escogido el oido derecho o izquierdo
            if(contNivelVolumen ==1000){
                mensaje(";TIENE QUE ESCOGER CON QUE OÍDO INICIARA EL
ANÁLISIS!");
            }else {
                //Disminuyen los dB
                valueY++;
                contNivelVolumen -= 5;
                //Se activa cuando las Frecuencias sean mayores a 1000
                if(valueX>tamañoFrecuenciasMenores1000Analizando){
                    contID=valueX-1;
                }
                //Cuando llegue F[Hz] a 0Hz, subir el indicador al valor
                mayor que 1000Hz y restablecer valores a Default
                if (valueX == 1&& valueY==1) {
                    valueX = tamañoFrecuenciasMenores1000Analizando + 1;
                    valueY = -8;
                    contNivelVolumen = 40;
                    banderaID=false;
                    banderabntNoPuedoOir=true;
                    //Cuando no existan valores mayores a 1000Hz
                    if (tamañoFrecuenciasMenores1000Analizando ==
envioparametromaximovolumenAnalizando && banderaOidoD == true &&
contActivacionRadioButtonD == 0) {
                        contActivacionRadioButtonD = 1;
                        contActivacionRadioButtonI = 1;
                    }
                }
            }
        }
    }
}

```

```

        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonIzquierdo.setChecked(true);
        break;
    } else if (tamañoFrecuenciasMenores1000Analizando ==
envioparametromaximovolumenAnalizando && banderaOidoI == true &&
contActivacionRadioButtonI == 0) {
        contActivacionRadioButtonI = 1;
        contActivacionRadioButtonD = 1;
        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonDerecho.setChecked(true);
        break;
    } else if (tamañoFrecuenciasMenores1000Analizando ==
envioparametromaximovolumenAnalizando && contActivacionRadioButtonI == 1
&& contActivacionRadioButtonD == 1) {
        //Guardar Analisis y cargar ResultadosActivity
        GuardarAnalisisYMas();
    }
}
//Cuando llegue dB a 0dB y F[Hz] sean mayores al
tamañoFrecuenciasMenores1000Analizando, ir subiendo la F[Hz] y
restablecer valores a Default
    if (contNivelVolumen < -1 && valueX >
tamañoFrecuenciasMenores1000Analizando) {
        valueX++;
        valueY = -8;
        contNivelVolumen = 40;
        banderaID=false;
        banderabntNoPuedoOir=true;
        //Cuando termina el analisis de uno de los oidos
        if (valueX > envioparametromaximovolumenAnalizando &&
banderaOidoD == true && contActivacionRadioButtonD == 0) {
            contActivacionRadioButtonD = 1;
            contActivacionRadioButtonI = 1;
            banderaPasoOido=true;
            banderabntNoPuedoOir=false;
            radioButtonIzquierdo.setChecked(true);
            break;
        } else if (valueX >
envioparametromaximovolumenAnalizando && banderaOidoI == true &&
contActivacionRadioButtonI == 0) {
            contActivacionRadioButtonI = 1;
            contActivacionRadioButtonD = 1;
            banderaPasoOido=true;
            banderabntNoPuedoOir=false;
            radioButtonDerecho.setChecked(true);
            break;
        } else if (valueX ==
enviovectorFrecuenciasActivasGrafica -1 && contActivacionRadioButtonI ==
1 && contActivacionRadioButtonD == 1) {
            //Guardar Analisis y cargar ResultadosActivity
            GuardarAnalisisYMas();
        }
    }
//Cuando llegue dB a 0 y F[Hz] sean menores al
tamañoFrecuenciasMenores1000Analizando, ir bajando la F[HZ] y restablecer
valores a Default
    if (contNivelVolumen <= -1 && valueX <=
tamañoFrecuenciasMenores1000Analizando) {
        contNivelVolumen = 40;

```

```

        valueX--;
        valueY = -8;
        banderaID=false;
    }
    //Se van graficando los valores menores
    if ( 0<=contNivelVolumen && contNivelVolumen <= 100 &&
valueX != 0) {
        //Volumen para oido Derecho
        if (banderaOidoD == true) {
            nivelVolumen(0, (contNivelVolumen / 100f));
            if(banderaID==true){
                graficar.removeGraficaD(contID);
            }
            if(banderaPasoOido==true){
                graficar.graficarI();
            }
            graficar.initD(valueX, valueY, false);
            banderaID=true;
        }//Volumen para oido Izquierdo
        else if (banderaOidoI == true) {
            nivelVolumen((contNivelVolumen / 100f), 0);
            if(banderaID==true){
                graficar.removeGraficaI(contID);
            }
            if(banderaPasoOido==true){
                graficar.graficarD();
            }
            graficar.initI(valueX, valueY, false);
            banderaID=true;
        }
    }
}
break;
case R.id.btnApenasPuedoOir:
    //Si se preciona los botones de respuesta sin antes haber
    escogido el oido derecho o izquierdo
    if(contNivelVolumen ==1000){
        mensaje(";TIENE QUE ESCOGER CON QUE OÍDO INICIARA EL
ANÁLISIS!");
    }else{
        //Aumentando los dB
        valueY--;
        contNivelVolumen +=5;
        //Se activa cuando las Frecuencias sean mayores a 1000
        if(valueX>tamañoFrecuenciasMenores1000Analizando){
            contID=valueX-1;
        }
        //Se activa cuando llegue los dB a los nivel mas altos de
        cada frecuencia

        if(arrayparametrosmaximoVolumen.get(tamañoFrecuenciasMenores1000Analizando
o2)==valueY*-1&&valueX<=tamañoFrecuenciasMenores1000Analizando){
            banderanelVolumenFinal=true;
        }else
        if(arrayparametrosmaximoVolumen.get(tamañoFrecuenciasMenores1000Analizando
o3)==valueY*-1&&valueX>tamañoFrecuenciasMenores1000Analizando){
            banderanelVolumenFinal=true;
        }
        //Cuando llegue F[Hz] a 0Hz, subir el indicador al valor
        mayor que 1000Hz y restablecer valores a Default
        if(valueX==1&&

```

```

arrayparametrosmaximoVolumen.get (tamañoFrecuenciasMenores1000Analizando2)
==(valueY+1)*-1) {
    valueX=tamañoFrecuenciasMenores1000Analizando+1;
    valueY=-8;
    contNivelVolumen =40;
    banderaID=false;
    banderabntNoPuedoOir=true;
    //Cuando no existan valores mayores a 1000Hz
    if (tamañoFrecuenciasMenores1000Analizando==
envioparametromaximovolumenAnalizando
&&banderaOidoD==true&&contActivacionRadioButtonD==0) {
        contActivacionRadioButtonD=1;
        contActivacionRadioButtonI=1;
        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonIzquierdo.setChecked (true);
        break;
    }else if (tamañoFrecuenciasMenores1000Analizando==
envioparametromaximovolumenAnalizando
&&banderaOidoI==true&&contActivacionRadioButtonI==0) {
        contActivacionRadioButtonI=1;
        contActivacionRadioButtonD=1;
        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonDerecho.setChecked (true);
        break;
    }else if (tamañoFrecuenciasMenores1000Analizando==
envioparametromaximovolumenAnalizando
&&contActivacionRadioButtonI==1&&contActivacionRadioButtonD==1) {
        //Guardar Analisis y cargar ResultadosActivity
        GuardarAnalisisYMas();
    }
}
//Cuando llegue dB al envioparametromaximovolumenAnalizando y
F[Hz] sean mayores al tamañoFrecuenciasMenores1000Analizando, ir subiendo
la F[Hz] y restablecer valores a Default

if (arrayparametrosmaximoVolumen.get (tamañoFrecuenciasMenores1000Analizand
o3) == (valueY+1)*-1&&valueX>tamañoFrecuenciasMenores1000Analizando) {
    valueX++;
    valueY=-8;
    contNivelVolumen =40;
    banderaID=false;
    tamañoFrecuenciasMenores1000Analizando3++;
    banderabntNoPuedoOir=true;
    if (valueX> envioparametromaximovolumenAnalizando
&&banderaOidoD==true&&contActivacionRadioButtonD==0) {
        contActivacionRadioButtonD=1;
        contActivacionRadioButtonI=1;
        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonIzquierdo.setChecked (true);
        break;
    }else if (valueX> envioparametromaximovolumenAnalizando
&&banderaOidoI==true&&contActivacionRadioButtonI==0) {
        contActivacionRadioButtonI=1;
        contActivacionRadioButtonD=1;
        banderaPasoOido=true;
        banderabntNoPuedoOir=false;
        radioButtonDerecho.setChecked (true);
        break;
    }
}

```

```

    }else if(valueX== enviovectorFrecuenciasActivasGrafica -
1&&contActivacionRadioButtonI==1&&contActivacionRadioButtonD==1) {
        //Guardar Analisis y cargar ResultadosActivity
        GuardarAnalisisYMas();
    }
}
//Cuando llegue los dB al
envioparametromaximovolumenAnalizando y Hz menores a los
valorinicialAnalizando, ir bajando la F[HZ] y restablecer valores a
Default
    if
(arrayparametrosmaximoVolumen.get(tamañoFrecuenciasMenores1000Analizando2
)==(valueY+1)*-1 &&valueX<=tamañoFrecuenciasMenores1000Analizando) {
        valueX--1;
        valueY--8;
        contNivelVolumen =40;
        banderaID=false;
        tamañoFrecuenciasMenores1000Analizando2--;
    }
//Se van graficando los valores mayores
if(0<=contNivelVolumen && contNivelVolumen <= 100){
    //Volumen para oido Derecho
    if(banderaOidoD==true) {
        nivelVolumen(0, (contNivelVolumen /100f));
        if(banderaID==true) {
            graficar.removeGraficaD(contID);
        }
        if(banderaPasoOido==true) {
            graficar.graficarI();
        }
        graficar.initD(valueX,
valueY,banderanivelVolumenFinal);
        banderaID=true;
        banderanivelVolumenFinal=false;
    }//Volumen para oido Izquierdo
    else if(banderaOidoI==true) {
        nivelVolumen((contNivelVolumen /100f),0);
        if(banderaID==true) {
            graficar.removeGraficaI(contID);
        }
        if(banderaPasoOido==true) {
            graficar.graficarD();
        }
    }

graficar.initI(valueX,valueY,banderanivelVolumenFinal);
    banderaID=true;
    banderanivelVolumenFinal=false;
}
}
}
break;
case R.id.btnNoPuedoOir:
    //Si se preciona los botones de respuesta sin antes haber
escogido el oido derecho o izquierdo
    if(contNivelVolumen==1000) {
        mensaje(";TIENE QUE ESCOGER CON QUE OÍDO INICIARA EL
ANÁLISIS!");
    }else{
        if (banderabntNoPuedoOir==false) {
            valueX--1;
            valueY--8;

```

```

        contNivelVolumen = 40;
    }
    //Ir subiendo las F[Hz] mayorer a 1000Hz y restableciendo
    los valores a Default
    if(valueX>tamañoFrecuenciasMenores1000Analizando){
        valueX++;
        valueY=-8;
        contNivelVolumen =40;
        banderabntNoPuedoOir=true;
        tamañoFrecuenciasMenores1000Analizando3++;
        if(valueX> envioparametromaximovolumenAnalizando
&&banderaOidoD==true&&contActivacionRadioButtonD==0){
            contActivacionRadioButtonD=1;
            contActivacionRadioButtonI=1;
            banderabntNoPuedoOir=false;
            banderaPasoOido=true;
            radioButtonIzquierdo.setChecked(true);
            break;
        }else if(valueX>
envioparametromaximovolumenAnalizando
&&banderaOidoI==true&&contActivacionRadioButtonI==0){
            contActivacionRadioButtonI=1;
            contActivacionRadioButtonD=1;
            banderabntNoPuedoOir=false;
            banderaPasoOido=true;
            radioButtonDerecho.setChecked(true);
            break;
        }else if(valueX==
enviovectorFrecuenciasActivasGrafica -
1&&contActivacionRadioButtonI==1&&contActivacionRadioButtonD==1){
            //Guardar Analisis y cargar ResultadosActivity
            GuardarAnalisisYMas();
        }
    }
    //Cuando llegue F[Hz] a 0Hz, subir el indicador al valor
    mayor que 1000Hz y restablecer valores a Default
    if(valueX==0){
        valueX=tamañoFrecuenciasMenores1000Analizando+1;
        valueY=-8;
        contNivelVolumen =40;
        banderabntNoPuedoOir=true;
        //Cuando no existan valores mayores a 1000Hz
        if(tamañoFrecuenciasMenores1000Analizando==
envioparametromaximovolumenAnalizando
&&banderaOidoD==true&&contActivacionRadioButtonD==0){
            contActivacionRadioButtonD=1;
            contActivacionRadioButtonI=1;
            banderabntNoPuedoOir=false;
            banderaPasoOido=true;
            radioButtonIzquierdo.setChecked(true);
            break;
        }else if(tamañoFrecuenciasMenores1000Analizando==
envioparametromaximovolumenAnalizando
&&banderaOidoI==true&&contActivacionRadioButtonI==0){
            contActivacionRadioButtonI=1;
            contActivacionRadioButtonD=1;
            banderabntNoPuedoOir=false;
            banderaPasoOido=true;
            radioButtonDerecho.setChecked(true);
            break;
        }else if(tamañoFrecuenciasMenores1000Analizando==

```

```

envioparametromaximovolumenAnalizando
&&contActivacionRadioButtonI==1&&contActivacionRadioButtonD==1){
    //Guardar Analisis y cargar ResultadosActivity
    GuardarAnalisisYMas();
}
}
//Se van graficando los valores
if (0 <= contNivelVolumen && contNivelVolumen <= 100 &&
valueX != 0) {
    //Volumen para oido Derecho
    if (banderaOidoD == true) {
        nivelVolumen(0, 0.4f);
        if(banderaPasoOido==true){
            graficar.graficarI();
        }
        graficar.initD(valueX, valueY, false);
        banderaID = true;
    }//Volumen para oido Izquierdo
    else if (banderaOidoI == true) {
        nivelVolumen(0.4f, 0);
        if(banderaPasoOido==true){
            graficar.graficarD();
        }
        graficar.initI(valueX, valueY, false);
        banderaID = true;
    }
}
}
break;
default:
break;
}
}

private void inicializarSoundPool() {
    if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.LOLLIPOP){
        AudioAttributes audioAttributes=new AudioAttributes.Builder()
            .setContentTypes(AudioAttributes.CONTENT_TYPE_MUSIC)
            .setUsage(AudioAttributes.USAGE_GAME)
            .setFlags(AudioAttributes.FLAG_AUDIBILITY_ENFORCED)
            .build();
        sp = new SoundPool.Builder()
            .setMaxStreams(1)
            .setAudioAttributes(audioAttributes)
            .build();
        this.setVolumeControlStream(STREAM_MUSIC);
        //Cargando Audios
        for(int i=1;i<=(enviovectorFrecuenciasActivasGrafica-2);i++){
            frecuenciaActiva=frecuenciasEnviadas[i];
            if(frecuenciaActiva.equalsIgnoreCase("125")){
                arrayAudiosCargados.add(sp.load(this, R.raw.a125hz,
0));
            }
            else if(frecuenciaActiva.equalsIgnoreCase("250")){
                arrayAudiosCargados.add(sp.load(this, R.raw.a250hz,
0));
            }
            else if(frecuenciaActiva.equalsIgnoreCase("500")){
                arrayAudiosCargados.add(sp.load(this, R.raw.a500hz,
0));
            }
            else if(frecuenciaActiva.equalsIgnoreCase("750")){
                arrayAudiosCargados.add(sp.load(this, R.raw.a750hz,
0));
            }
        }
    }
}

```

```

    }else if(frecuenciaActiva.equalsIgnoreCase("1000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a1000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("1500")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a1500hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("2000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a2000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("3000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a3000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("4000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a4000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("6000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a6000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("8000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a8000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("10000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a10000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("12000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a12000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("14000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a14000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("16000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a16000hz,
0));
    }else if(frecuenciaActiva.equalsIgnoreCase("18000")){
        arrayAudiosCargados.add(sp.load(this, R.raw.a18000hz,
0));
    }
}
}else{
    sp = new SoundPool(1,AudioManager.STREAM_MUSIC,1);
    this.setVolumeControlStream(STREAM_MUSIC);
    //Cargando Audios
    for(int i=1;i<=(enviovectorFrecuenciasActivasGrafica-2);i++){
        frecuenciaActiva=frecuenciasEnviadas[i];
        if(frecuenciaActiva.equalsIgnoreCase("125")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a125hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("250")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a250hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("500")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a500hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("750")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a750hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("1000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a1000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("1500")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a1500hz,
0));
        }
    }
}
}

```

```

0));
        }else if(frecuenciaActiva.equalsIgnoreCase("2000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a2000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("3000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a3000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("4000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a4000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("6000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a6000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("8000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a8000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("10000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a10000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("12000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a12000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("14000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a14000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("16000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a16000hz,
0));
        }else if(frecuenciaActiva.equalsIgnoreCase("18000")){
            arrayAudiosCargados.add(sp.load(this, R.raw.a18000hz,
0));
        }
    }
}

private void nivelVolumen(float volI,float volD){
    //Iniciando Audio
    sp.play(arrayAudiosCargados.get(valueX-1),volI,volD,1,-1,1);
}

private void mensaje(String mensajes){
    Toast toast = Toast.makeText(AnalizandoActivity.this, mensajes,
Toast.LENGTH_LONG);
    toast.show();
}

private void CargarConfiguracion(){
    //Cargando valores de ConfiguracionActivity
    SharedPreferences configuraciones=
getSharedPreferences("CONFIGURACIONES", Context.MODE_PRIVATE);
    tamañoFrecuenciasMenores1000Analizando=
configuraciones.getInt("tamañoFrecuenciasMenores1000Analizando",3);

tamañoFrecuenciasMenores1000Analizando2=configuraciones.getInt("tamañoFre
cuenciasMenores1000Analizando",3)-1;

tamañoFrecuenciasMenores1000Analizando3=configuraciones.getInt("tamañoFre
cuenciasMenores1000Analizando",3);
    enviovectorFrecuenciasActivasGrafica =
configuraciones.getInt("tamañoFrecuenciasGraficadas",9);

```

```

envioparametromaximovolumenAnalizando=configuraciones.getInt("parametromaximovolumenAnalizando",7);
    vectorarrayparametrosmaximoVolumen
=configuraciones.getString("arrayparametrosmaximoVolumen","14,15,15,16,16,15,15,")
.split(",");

frecuenciasEnviadas=configuraciones.getString("vectorFrecuenciasActivas",
" ,250,500,1000,2000,4000,6000,8000, ,").split(",");
    }

    private void GuardarAnalisisYMas(){
        //Parar Audio
        sp.release();

        //Guardando el analisis del Paciente
        numeroPaciente();
        SharedPreferences configuraciones=
        getSharedPreferences("CONFIGURACIONES", Context.MODE_PRIVATE);
        SharedPreferences datospersona=
        getSharedPreferences("DATOSPERSONA"+numeroPaciente,Context.MODE_PRIVATE);
        SharedPreferences.Editor editor=datospersona.edit();

        //Guardar Array de parametro Analisis oido Derecho
        StringBuilder stringbuilderArrayOidoD=new StringBuilder();
        for(int i=0;i<graficar.guardarAnalisisD().size();i++){
            int x = (int) graficar.guardarAnalisisD().get(i).getX();
            int y = (int) graficar.guardarAnalisisD().get(i).getY();
            boolean
xySeries=graficar.guardarAnalisisD().get(i).getBanderanivelVolumenFinal()
;
            stringbuilderArrayOidoD.append(x+","+y+","+xySeries);
            stringbuilderArrayOidoD.append(",");
        }
        editor.putString("ArrayOidoD",stringbuilderArrayOidoD.toString());

        //Guardar Array de parametro Analisis oido Derecho
        StringBuilder stringbuilderArrayOidoI=new StringBuilder();
        for(int i=0;i<graficar.guardarAnalisisI().size();i++){
            int x = (int) graficar.guardarAnalisisI().get(i).getX();
            int y = (int) graficar.guardarAnalisisI().get(i).getY();
            boolean
xySeries=graficar.guardarAnalisisI().get(i).getBanderanivelVolumenFinal()
;
            stringbuilderArrayOidoI.append(x+","+y+","+xySeries);
            stringbuilderArrayOidoI.append(",");
        }
        editor.putString("ArrayOidoI",stringbuilderArrayOidoI.toString());

        editor.putString("vectorFrecuenciasActivas",configuraciones.getString("vectorFrecuenciasActivas",
" ,250,500,1000,2000,4000,6000,8000, ,"));
        editor.commit();
        //Envio a ResultadosActivity
        Intent intent=new Intent(this, ResultadosActivity.class);
        startActivity(intent);
        contNivelVolumen=-2;

    }

    private void numeroPaciente(){
        SharedPreferences numeroPacienteSharedPreferences=

```

```

getSharedPreferences("numeroPaciente", Context.MODE_PRIVATE);

numeroPaciente=numeroPacienteSharedPreferences.getInt("numPaciente",0);
}

@Override
public void onResume() {
    IntentFilter filter = new
IntentFilter(Intent.ACTION_HEADSET_PLUG);
    registerReceiver(conexionauricular, filter);
    super.onResume();
}

@Override
public void onPause() {
    unregisterReceiver(conexionauricular);
    super.onPause();
}
}

```

Descripción: Esta clase se encarga:

- Inicializa los diferentes componentes.
- Inicializa los diferentes tonos de frecuencia.
- Iniciar los tonos dependiendo de las versiones de Android.
- cargar la configuración de la actividad cargar configuración.
- Métodos para iniciar las pruebas de tonos para el oído izquierdo y derecho enviar las coordenadas de cada interacción del usuario con la frecuencia y los dB para ir graficando y disminuyendo o aumentando los niveles de dB además de subir y bajar las frecuencias con sus validaciones para los distintos botones.

Anexo. F

Descripción de la clase Java Configuración.

Fragmento de código clase Java Configuración.java

```
public class ConfiguracionActivity extends AppCompatActivity implements
View.OnClickListener {
    private CheckBox checkBoxconfiguracion_125Hz,
checkBoxconfiguracion_250Hz, checkBoxconfiguracion_500Hz,
checkBoxconfiguracion_750Hz, checkBoxconfiguracion_1000Hz,
checkBoxconfiguracion_1500Hz,
        checkBoxconfiguracion_2000Hz, checkBoxconfiguracion_3000Hz,
checkBoxconfiguracion_4000Hz, checkBoxconfiguracion_6000Hz,
checkBoxconfiguracion_8000Hz, checkBoxconfiguracion_10000Hz,
        checkBoxconfiguracion_12000Hz, checkBoxconfiguracion_14000Hz,
checkBoxconfiguracion_16000Hz, checkBoxconfiguracion_18000Hz;
    private Switch
swconfiguracionSelecciónCoefCalibracion, swconfiguracionNombrePaciente,

swconfiguracionEliminacionEnmascaramientoSonora, swconfiguracionMonitoreo
oRuido, swconfiguracionClasificacionOMS,

swconfiguracionGradosPerdidaAudicion, swconfiguracionBananaHabla, swconfigu
racionNormalEdad;
    private Button botonconfiguracionbuttonPredeterminados,
botonconfiguracionbuttonSeleccionarTodas,
botonconfiguracionbuttonAceptar;

    //Declaración de listas para guardado de información requerida por
las actividades
    private ArrayList<Integer> arrayparametrosmaximoVolumen =new
ArrayList<Integer>() {{
        add(14);add(15);add(15);add(16);add(16);add(15);add(15);}};
    private ArrayList<String> vectorFrecuenciasActivas =new
ArrayList<String>() {{
        add("
");add("250");add("500");add("1000");add("2000");add("4000");add("6000");
add("8000");add(" ");}};
    private int cont=0;
    public int tamañoFrecenciasGraficadas =
vectorFrecuenciasActivas.size(), tamañoFrecuenciasMenores1000Analizando=3;
    //Declaración de banderas
    public static boolean
banderacheckBoxconfiguracion_125Hz=false, banderacheckBoxconfiguracion_250
Hz=true,

banderacheckBoxconfiguracion_500Hz=true, banderacheckBoxconfiguracion_750H
z=false,

banderacheckBoxconfiguracion_1000Hz=true, banderacheckBoxconfiguracion_150
0Hz=false,

banderacheckBoxconfiguracion_2000Hz=true, banderacheckBoxconfiguracion_300
0Hz=false,

banderacheckBoxconfiguracion_4000Hz=true, banderacheckBoxconfiguracion_600
0Hz=true,

banderacheckBoxconfiguracion_8000Hz=true, banderacheckBoxconfiguracion_100
```

```

00Hz=false,

banderacheckBoxconfiguracion_12000Hz=false,banderacheckBoxconfiguracion_1
4000Hz=false,

banderacheckBoxconfiguracion_16000Hz=false,banderacheckBoxconfiguracion_1
8000Hz=false,

banderaswconfiguracionSelecciónCoefCalibracion=false,banderaswconfiguraci
onNombrePaciente=true,

banderaswconfiguracionEliminacionEnmascaramientoSonora=false,banderaswc
onfiguracionMonitoreoRuido=false,

banderaswconfiguracionClasificacionOMS=true,banderaswconfiguracionGradosP
erdidaAudicion=true,

banderaswconfiguracionBananaHabla=false,banderaswconfiguracionNormalEdad=
false;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_configuracion);
    //Declaración de variables Switch, CheckBox, Button
    swconfiguracionSelecciónCoefCalibracion =
(Switch) findViewById(R.id.swconfiguracionSelecciónCoefCalibracion);
    swconfiguracionNombrePaciente =
(Switch) findViewById(R.id.swconfiguracionNombrePaciente);
    swconfiguracionEliminacionEnmascaramientoSonora =
(Switch) findViewById(R.id.swconfiguracionEliminacionEnmascaramientoSono
ra);
    swconfiguracionMonitoreoRuido =
(Switch) findViewById(R.id.swconfiguracionMonitoreoRuido);
    swconfiguracionClasificacionOMS =
(Switch) findViewById(R.id.swconfiguracionClasificacionOMS);
    swconfiguracionBananaHabla =
(Switch) findViewById(R.id.swconfiguracionBananaHabla);
    swconfiguracionGradosPerdidaAudicion =
(Switch) findViewById(R.id.swconfiguracionGradosPerdidaAudicion);
    swconfiguracionNormalEdad =
(Switch) findViewById(R.id.swconfiguracionNormalEdad);
    checkBoxconfiguracion_125Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_125Hz);
    checkBoxconfiguracion_250Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_250Hz);
    checkBoxconfiguracion_500Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_500Hz);
    checkBoxconfiguracion_750Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_750Hz);
    checkBoxconfiguracion_1000Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_1000Hz);
    checkBoxconfiguracion_1500Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_1500Hz);
    checkBoxconfiguracion_2000Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_2000Hz);
    checkBoxconfiguracion_3000Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_3000Hz);
    checkBoxconfiguracion_4000Hz =
(CheckBox) findViewById(R.id.checkBoxconfiguracion_4000Hz);

```

```

        checkBoxconfiguracion_6000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_6000Hz);
        checkBoxconfiguracion_8000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_8000Hz);
        checkBoxconfiguracion_10000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_10000Hz);
        checkBoxconfiguracion_12000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_12000Hz);
        checkBoxconfiguracion_14000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_14000Hz);
        checkBoxconfiguracion_16000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_16000Hz);
        checkBoxconfiguracion_18000Hz =
        (CheckBox) findViewById(R.id.checkBoxconfiguracion_18000Hz);
        botonconfiguracionbuttonPredeterminados =
        (Button) findViewById(R.id.botonconfiguracionbuttonPredeterminados);
        botonconfiguracionbuttonSeleccionarTodas =
        (Button) findViewById(R.id.botonconfiguracionbuttonSeleccionarTodas);
        botonconfiguracionbuttonAceptar =
        (Button) findViewById(R.id.botonconfiguracionbuttonAceptar);

        //Iniciando escucha de las acciones de la pantalla
        botonconfiguracionbuttonPredeterminados.setOnClickListener(this);
        botonconfiguracionbuttonSeleccionarTodas.setOnClickListener(this);
        botonconfiguracionbuttonAceptar.setOnClickListener(this);
        swconfiguracionBananaHabla.setOnClickListener(this);
        swconfiguracionGradosPerdidaAudicion.setOnClickListener(this);
        swconfiguracionNormalEdad.setOnClickListener(this);
        swconfiguracionSelecciónCoefCalibracion.setOnClickListener(this);
        swconfiguracionNombrePaciente.setOnClickListener(this);

        swconfiguracionEliminacionEnmascaramientoSonora.setOnClickListener(this);
    };

    swconfiguracionMonitoreoRuido.setOnClickListener(this);
    swconfiguracionClasificacionOMS.setOnClickListener(this);
}

protected void onResume() {
    super.onResume();
    //Acciones Guardadas Previamente
    SharedPreferences configuraciones=
    getSharedPreferences("CONFIGURACIONES", Context.MODE_PRIVATE);

    checkBoxconfiguracion_125Hz.setChecked(configuraciones.getBoolean("bander
    acheckBoxconfiguracion_125Hz", false));

    checkBoxconfiguracion_250Hz.setChecked(configuraciones.getBoolean("bander
    acheckBoxconfiguracion_250Hz", true));

    checkBoxconfiguracion_500Hz.setChecked(configuraciones.getBoolean("bander
    acheckBoxconfiguracion_500Hz", true));

    checkBoxconfiguracion_750Hz.setChecked(configuraciones.getBoolean("bander
    acheckBoxconfiguracion_750Hz", false));

    checkBoxconfiguracion_1000Hz.setChecked(configuraciones.getBoolean("bande
    rachecheckBoxconfiguracion_1000Hz", true));

    checkBoxconfiguracion_1500Hz.setChecked(configuraciones.getBoolean("bande

```

```

racheckBoxconfiguracion_1500Hz", false));

checkBoxconfiguracion_2000Hz.setChecked(configuraciones.getBoolean("bande
racheckBoxconfiguracion_2000Hz", true));

checkBoxconfiguracion_3000Hz.setChecked(configuraciones.getBoolean("bande
racheckBoxconfiguracion_3000Hz", false));

checkBoxconfiguracion_4000Hz.setChecked(configuraciones.getBoolean("bande
racheckBoxconfiguracion_4000Hz", true));

checkBoxconfiguracion_6000Hz.setChecked(configuraciones.getBoolean("bande
racheckBoxconfiguracion_6000Hz", true));

checkBoxconfiguracion_8000Hz.setChecked(configuraciones.getBoolean("bande
racheckBoxconfiguracion_8000Hz", true));

checkBoxconfiguracion_10000Hz.setChecked(configuraciones.getBoolean("band
eracheckBoxconfiguracion_10000Hz", false));

checkBoxconfiguracion_12000Hz.setChecked(configuraciones.getBoolean("band
eracheckBoxconfiguracion_12000Hz", false));

checkBoxconfiguracion_14000Hz.setChecked(configuraciones.getBoolean("band
eracheckBoxconfiguracion_14000Hz", false));

checkBoxconfiguracion_16000Hz.setChecked(configuraciones.getBoolean("band
eracheckBoxconfiguracion_16000Hz", false));

checkBoxconfiguracion_18000Hz.setChecked(configuraciones.getBoolean("band
eracheckBoxconfiguracion_18000Hz", false));

swconfiguracionSelecciónCoefCalibracion.setChecked(configuraciones.getBoo
lean("banderaswconfiguracionSelecciónCoefCalibracion", false));

swconfiguracionNombrePaciente.setChecked(configuraciones.getBoolean("band
eraswconfiguracionNombrePaciente", true));

swconfiguracionEliminacionEnmascaramientoSonora.setChecked(configuracio
nes.getBoolean("banderaswconfiguracionEliminacionEnmascaramientoSonora"
, false));

swconfiguracionMonitoreoRuido.setChecked(configuraciones.getBoolean("band
eraswconfiguracionMonitoreoRuido", false));

swconfiguracionClasificacionOMS.setChecked(configuraciones.getBoolean("ba
nderaswconfiguracionClasificacionOMS", true));

swconfiguracionBananaHabla.setChecked(configuraciones.getBoolean("bandera
swconfiguracionBananaHabla", false));

swconfiguracionGradosPerdidaAudicion.setChecked(configuraciones.getBoolea
n("banderaswconfiguracionGradosPerdidaAudicion", true));

swconfiguracionNormalEdad.setChecked(configuraciones.getBoolean("banderas
wconfiguracionNormalEdad", false));
}

@Override
public void onClick(View view) {

```

```

//Escucha selectiva de las acciones requeridas, para su
respectivo llamado de actividades
switch (view.getId()){
    case R.id.swconfiguracionBananaHabla:
        if(swconfiguracionBananaHabla.isChecked()==true){
            banderaswconfiguracionBananaHabla=true;
        }else{
            banderaswconfiguracionBananaHabla=false;
        }
        break;
    case R.id.swconfiguracionGradosPerdidaAudicion:
if(swconfiguracionGradosPerdidaAudicion.isChecked()==true){
            banderaswconfiguracionGradosPerdidaAudicion=true;
        }else{
            banderaswconfiguracionGradosPerdidaAudicion=false;
        }
        break;
    case R.id.swconfiguracionNormalEdad:
        if(swconfiguracionNormalEdad.isChecked()==true){
            banderaswconfiguracionNormalEdad=true;
        }else{
            banderaswconfiguracionNormalEdad=false;
        }
        break;
    case R.id.swconfiguracionClasificacionOMS:
        if(swconfiguracionClasificacionOMS.isChecked()==true){
            banderaswconfiguracionClasificacionOMS=true;
        }else{
            banderaswconfiguracionClasificacionOMS=false;
        }
        break;
    case R.id.swconfiguracionEliminacionEnmascararamientoSonora:
if(swconfiguracionEliminacionEnmascararamientoSonora.isChecked()==true){
            banderaswconfiguracionEliminacionEnmascararamientoSonora=true;
        }else{
            banderaswconfiguracionEliminacionEnmascararamientoSonora=false;
        }
        break;
    case R.id.swconfiguracionMonitoreoRuido:
        if(swconfiguracionMonitoreoRuido.isChecked()==true){
            banderaswconfiguracionMonitoreoRuido=true;
        }else{
            banderaswconfiguracionMonitoreoRuido=false;
        }
        break;
    case R.id.swconfiguracionNombrePaciente:
        if(swconfiguracionNombrePaciente.isChecked()==true){
            banderaswconfiguracionNombrePaciente=true;
        }else{
            banderaswconfiguracionNombrePaciente=false;
        }
        break;
    case R.id.swconfiguracionSelecciónCoefCalibracion:
if(swconfiguracionSelecciónCoefCalibracion.isChecked()==true){
            banderaswconfiguracionSelecciónCoefCalibracion=true;

```

```

    }else{
        banderaswconfiguracionSelecciónCoefCalibracion=false;
    }
    break;
//Automaticamente seleccionar todas las frecuencias
case R.id.buttonconfiguracionbuttonSeleccionarTodas:
    checkBoxconfiguracion_125Hz.setChecked(true);
    checkBoxconfiguracion_250Hz.setChecked(true);
    checkBoxconfiguracion_500Hz.setChecked(true);
    checkBoxconfiguracion_750Hz.setChecked(true);
    checkBoxconfiguracion_1000Hz.setChecked(true);
    checkBoxconfiguracion_1500Hz.setChecked(true);
    checkBoxconfiguracion_2000Hz.setChecked(true);
    checkBoxconfiguracion_3000Hz.setChecked(true);
    checkBoxconfiguracion_4000Hz.setChecked(true);
    checkBoxconfiguracion_6000Hz.setChecked(true);
    checkBoxconfiguracion_8000Hz.setChecked(true);
    checkBoxconfiguracion_10000Hz.setChecked(true);
    checkBoxconfiguracion_12000Hz.setChecked(true);
    checkBoxconfiguracion_14000Hz.setChecked(true);
    checkBoxconfiguracion_16000Hz.setChecked(true);
    checkBoxconfiguracion_18000Hz.setChecked(true);
    break;
//Automaticamente seleccionar las frecuencias
prederterminadas
case R.id.buttonconfiguracionbuttonPredeterminados:
    checkBoxconfiguracion_125Hz.setChecked(false);
    checkBoxconfiguracion_250Hz.setChecked(true);
    checkBoxconfiguracion_500Hz.setChecked(true);
    checkBoxconfiguracion_750Hz.setChecked(false);
    checkBoxconfiguracion_1000Hz.setChecked(true);
    checkBoxconfiguracion_1500Hz.setChecked(false);
    checkBoxconfiguracion_2000Hz.setChecked(true);
    checkBoxconfiguracion_3000Hz.setChecked(false);
    checkBoxconfiguracion_4000Hz.setChecked(true);
    checkBoxconfiguracion_6000Hz.setChecked(true);
    checkBoxconfiguracion_8000Hz.setChecked(true);
    checkBoxconfiguracion_10000Hz.setChecked(false);
    checkBoxconfiguracion_12000Hz.setChecked(false);
    checkBoxconfiguracion_14000Hz.setChecked(false);
    checkBoxconfiguracion_16000Hz.setChecked(false);
    checkBoxconfiguracion_18000Hz.setChecked(false);
    break;
//La selección de frecuencias predeterminadas, todas o las
escogidas por el usuario y además encapsular y enviar los datos a la
AnalizandoActivity y clase Grafica
case R.id.buttonconfiguracionbuttonAceptar:
    do {
        vectorFrecuenciasActivas =new ArrayList<String>();
        arrayparametrosmaximoVolumen =new
ArrayList<Integer>();
        tamañoFrecuenciasMenores1000Analizando=0;
        vectorFrecuenciasActivas.add(cont, " ");
        cont++;
        if (checkBoxconfiguracion_125Hz.isChecked() == true)
        {
            arrayparametrosmaximoVolumen.add(12);
            tamañoFrecuenciasMenores1000Analizando++;
            vectorFrecuenciasActivas.add(cont, "125");
            cont++;
            banderacheckBoxconfiguracion_125Hz = true;

```

```

    } else {
        banderacheckBoxconfiguracion_125Hz = false;
    }
    if (checkBoxconfiguracion_250Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(14);
        tamañoFrecuenciasMenores1000Analizando++;
        vectorFrecuenciasActivas.add(cont, "250");
        cont++;
        banderacheckBoxconfiguracion_250Hz = true;
    } else {
        banderacheckBoxconfiguracion_250Hz = false;
    }
    if (checkBoxconfiguracion_500Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(15);
        tamañoFrecuenciasMenores1000Analizando++;
        vectorFrecuenciasActivas.add(cont, "500");
        cont++;
        banderacheckBoxconfiguracion_500Hz = true;
    } else {
        banderacheckBoxconfiguracion_500Hz = false;
    }
    if (checkBoxconfiguracion_750Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(15);
        tamañoFrecuenciasMenores1000Analizando++;
        vectorFrecuenciasActivas.add(cont, "750");
        cont++;
        banderacheckBoxconfiguracion_750Hz = true;
    } else {
        banderacheckBoxconfiguracion_750Hz = false;
    }
    if (checkBoxconfiguracion_1000Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(15);
        tamañoFrecuenciasMenores1000Analizando++;
        vectorFrecuenciasActivas.add(cont, "1000");
        cont++;
        banderacheckBoxconfiguracion_1000Hz = true;
    } else {
        banderacheckBoxconfiguracion_1000Hz = false;
    }
    if (checkBoxconfiguracion_1500Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(15);
        vectorFrecuenciasActivas.add(cont, "1500");
        cont++;
        banderacheckBoxconfiguracion_1500Hz = true;
    } else {
        banderacheckBoxconfiguracion_1500Hz = false;
    }
    if (checkBoxconfiguracion_2000Hz.isChecked() == true)
    {
        arrayparametrosmaximoVolumen.add(16);
        vectorFrecuenciasActivas.add(cont, "2000");
        cont++;
        banderacheckBoxconfiguracion_2000Hz = true;
    } else {
        banderacheckBoxconfiguracion_2000Hz = false;
    }
}

```

```

        if (checkboxconfiguracion_3000Hz.isChecked() == true)
        {
            arrayparametrosmaximoVolumen.add(16);
            vectorFrecuenciasActivas.add(cont, "3000");
            cont++;
            banderacheckBoxconfiguracion_3000Hz = true;
        } else {
            banderacheckBoxconfiguracion_3000Hz = false;
        }
        if (checkboxconfiguracion_4000Hz.isChecked() == true)
        {
            arrayparametrosmaximoVolumen.add(16);
            vectorFrecuenciasActivas.add(cont, "4000");
            cont++;
            banderacheckBoxconfiguracion_4000Hz = true;
        } else {
            banderacheckBoxconfiguracion_4000Hz = false;
        }
        if (checkboxconfiguracion_6000Hz.isChecked() == true)
        {
            arrayparametrosmaximoVolumen.add(15);
            vectorFrecuenciasActivas.add(cont, "6000");
            cont++;
            banderacheckBoxconfiguracion_6000Hz = true;
        } else {
            banderacheckBoxconfiguracion_6000Hz = false;
        }
        if (checkboxconfiguracion_8000Hz.isChecked() == true)
        {
            arrayparametrosmaximoVolumen.add(15);
            vectorFrecuenciasActivas.add(cont, "8000");
            cont++;
            banderacheckBoxconfiguracion_8000Hz = true;
        } else {
            banderacheckBoxconfiguracion_8000Hz = false;
        }
        if (checkboxconfiguracion_10000Hz.isChecked() ==
true) {
            arrayparametrosmaximoVolumen.add(15);
            vectorFrecuenciasActivas.add(cont, "10000");
            cont++;
            banderacheckBoxconfiguracion_10000Hz = true;
        } else {
            banderacheckBoxconfiguracion_10000Hz = false;
        }
        if (checkboxconfiguracion_12000Hz.isChecked() ==
true) {
            arrayparametrosmaximoVolumen.add(15);
            vectorFrecuenciasActivas.add(cont, "12000");
            cont++;
            banderacheckBoxconfiguracion_12000Hz = true;
        } else {
            banderacheckBoxconfiguracion_12000Hz = false;
        }
        if (checkboxconfiguracion_14000Hz.isChecked() ==
true) {
            arrayparametrosmaximoVolumen.add(15);
            vectorFrecuenciasActivas.add(cont, "14000");
            cont++;
            banderacheckBoxconfiguracion_14000Hz = true;
        } else {

```

```

        banderacheckBoxconfiguracion_14000Hz = false;
    }
    if (checkBoxconfiguracion_16000Hz.isChecked() ==
true) {
        arrayparametrosmaximoVolumen.add(15);
        vectorFrecuenciasActivas.add(cont, "16000");
        cont++;
        banderacheckBoxconfiguracion_16000Hz = true;
    } else {
        banderacheckBoxconfiguracion_16000Hz = false;
    }
    if (checkBoxconfiguracion_18000Hz.isChecked() ==
true) {
        arrayparametrosmaximoVolumen.add(15);
        vectorFrecuenciasActivas.add(cont, "18000");
        cont++;
        banderacheckBoxconfiguracion_18000Hz = true;
    } else {
        banderacheckBoxconfiguracion_18000Hz = false;
    }
    vectorFrecuenciasActivas.add(cont, " ");
    cont++;
    if(vectorFrecuenciasActivas.get(1).equals("")) {
        checkBoxconfiguracion_250Hz.setChecked(true);
        checkBoxconfiguracion_500Hz.setChecked(true);
        checkBoxconfiguracion_1000Hz.setChecked(true);
        checkBoxconfiguracion_2000Hz.setChecked(true);
        checkBoxconfiguracion_4000Hz.setChecked(true);
        checkBoxconfiguracion_6000Hz.setChecked(true);
        checkBoxconfiguracion_8000Hz.setChecked(true);
        cont=0;
        mensaje("¡Al menos una frecuencia se debe
seleccionar!\n Se activarán las frecuencias por defecto.");
    }
    }while (vectorFrecuenciasActivas.get(1).equals("")));
    tamañoFrecenciasGraficadas =cont;
    cont=0;
    mensaje("¡Frecuencias Guardadas!");
    break;
default:
    break;
}
}

public void mensaje(String mensaje){
    Toast toast = Toast.makeText(this, mensaje, Toast.LENGTH_LONG);
    toast.show();
}

@Override
protected void onPause() {
    Context context=this;
    SharedPreferences configuraciones=
getSharedPreferences("CONFIGURACIONES", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor=configuraciones.edit();

    editor.putBoolean("banderacheckBoxconfiguracion_125Hz", banderacheckBoxcon
figuracion_125Hz); editor.putBoolean("banderacheckBoxconfiguracion_250Hz",
banderacheckBoxconfiguracion_250Hz);

    editor.putBoolean("banderacheckBoxconfiguracion_500Hz", banderacheckBoxcon

```

```

figuracion_500Hz);editor.putBoolean("banderacheckBoxconfiguracion_750Hz",
banderacheckBoxconfiguracion_750Hz);

editor.putBoolean("banderacheckBoxconfiguracion_1000Hz",banderacheckBoxco
nfiguracion_1000Hz);editor.putBoolean("banderacheckBoxconfiguracion_1500H
z",banderacheckBoxconfiguracion_1500Hz);

editor.putBoolean("banderacheckBoxconfiguracion_2000Hz",banderacheckBoxco
nfiguracion_2000Hz);editor.putBoolean("banderacheckBoxconfiguracion_3000H
z",banderacheckBoxconfiguracion_3000Hz);

editor.putBoolean("banderacheckBoxconfiguracion_4000Hz",banderacheckBoxco
nfiguracion_4000Hz);editor.putBoolean("banderacheckBoxconfiguracion_6000H
z",banderacheckBoxconfiguracion_6000Hz);

editor.putBoolean("banderacheckBoxconfiguracion_6000Hz",banderacheckBoxco
nfiguracion_6000Hz);editor.putBoolean("banderacheckBoxconfiguracion_8000H
z",banderacheckBoxconfiguracion_8000Hz);

editor.putBoolean("banderacheckBoxconfiguracion_10000Hz",banderacheckBoxc
onfiguracion_10000Hz);editor.putBoolean("banderacheckBoxconfiguracion_120
00Hz",banderacheckBoxconfiguracion_12000Hz);

editor.putBoolean("banderacheckBoxconfiguracion_14000Hz",banderacheckBoxc
onfiguracion_14000Hz);editor.putBoolean("banderacheckBoxconfiguracion_160
00Hz",banderacheckBoxconfiguracion_16000Hz);

editor.putBoolean("banderacheckBoxconfiguracion_18000Hz",banderacheckBoxc
onfiguracion_18000Hz);

editor.putBoolean("banderaswconfiguracionSelecciónCoefCalibracion",bander
aswconfiguracionSelecciónCoefCalibracion);editor.putBoolean("banderaswcon
figuracionNombrePaciente",banderaswconfiguracionNombrePaciente);

editor.putBoolean("banderaswconfiguracionEliminacionEnmascaramientoSono
ra",banderaswconfiguracionEliminacionEnmascaramientoSonora);editor.putB
oolean("banderaswconfiguracionMonitoreoRuido",banderaswconfiguracionMonit
oreoRuido);

editor.putBoolean("banderaswconfiguracionClasificacionOMS",banderaswconfi
guracionClasificacionOMS);editor.putBoolean("banderaswconfiguracionGrados
PerdidaAudicion",banderaswconfiguracionGradosPerdidaAudicion);

editor.putBoolean("banderaswconfiguracionBananaHabla",banderaswconfigurac
ionBananaHabla);editor.putBoolean("banderaswconfiguracionNormalEdad",band
eraswconfiguracionNormalEdad);

editor.putInt("tamañoFrecuenciasGraficadas",tamañoFrecuenciasGraficadas);ed
itor.putInt("tamañoFrecuenciasMenores1000Analizando",tamañoFrecuenciasMen
ores1000Analizando);
    editor.putInt("parametromaximovolumenAnalizando",
arrayparametrosmaximoVolumen.size());

    //Guardar Array de parametro maximo del volumen
    StringBuilder stringbuidlerparametrosmaximoVolumen=new
StringBuilder();
    for(Integer s : arrayparametrosmaximoVolumen){
        stringbuidlerparametrosmaximoVolumen.append(s);
        stringbuidlerparametrosmaximoVolumen.append(",");

```

```

    }

    editor.putString("arrayparametrosmaximoVolumen",stringbuiderverparametrosmaximoVolumen.toString());

    //Guardar Vector de Frecuencias Activas
    StringBuilder stringbuidervervectorFrecuenciasActivas=new
    StringBuilder();
    for(String s :vectorFrecuenciasActivas){
        stringbuidervervectorFrecuenciasActivas.append(s);
        stringbuidervervectorFrecuenciasActivas.append(",");
    }

    editor.putString("vectorFrecuenciasActivas",stringbuidervervectorFrecuenciasActivas.toString());
    editor.commit();
    super.onPause();
}
}

```

Descripción: Esta clase se encarga:

- Iniciar las frecuencias por defecto o permitir iniciar con las distintas frecuencias que el usuario desee.
- Activar o desactivar la actividad de pedido de información del usuario.
- Escoger el tipo de respuesta para el diagrama audiométrico

Anexo. G

Archivo XML Activity_Menu.xml (Menú Principal).

Descripción Código del archive ActivityMenu.xml
<pre><?xml version="1.0" encoding="utf-8" ?> <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="horizontal" android:background="@drawable/fondo" tools:context="com.example.paulandreslandazuri.audiometriamovil.MenuActiv ity"> <TextView android:id="@+id/textView" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_marginTop="@dimen/tituloMenulayout_marginTop" android:fontFamily="sans-serif-smallcaps" android:text="@string/menuTitulo" android:textSize="@dimen/tituloMenutextSize" app:layout_constraintLeft_toRightOf="@+id/imageView" app:layout_constraintTop_toTopOf="@+id/imageView" tools:ignore="RtlHardcoded" tools:layout_constraintLeft_creator="1" tools:layout_constraintTop_creator="1" /> <ImageView android:id="@+id/imageView" android:layout_width="@dimen/iconoLogolayout_width" android:layout_height="@dimen/iconoLogolayout_height" android:layout_marginStart="@dimen/iconoLogolayout_marginStart" android:layout_marginTop="@dimen/iconoLogolayout_marginTop" android:contentDescription="" app:layout_constraintTop_toTopOf="parent" app:srcCompat="@mipmap/ic_launcher" tools:ignore="ContentDescription,MissingConstraints,RtlHardcoded" tools:layout_constraintLeft_creator="1" tools:layout_constraintTop_creator="1" tools:layout_editor_absoluteX="24dp" /> <ImageView android:id="@+id/imgbotIniciar" android:layout_width="@dimen/menuimgIconos_width" android:layout_height="@dimen/menuimgIconos_width" android:layout_marginBottom="@dimen/menuiconollayout_marginBottom" android:layout_marginEnd="@dimen/menuiconollayout_marginEnd" android:layout_marginTop="@dimen/menuiconollayout_marginTop" app:layout_constraintBottom_toBottomOf="@+id/linearLayout" app:layout_constraintRight_toRightOf="@+id/linearLayout" app:layout_constraintTop_toTopOf="parent" app:srcCompat="@mipmap/ic_auriculares"</pre>

```

tools:ignore="ContentDescription,RtlHardcoded"
tools:layout_constraintBottom_creator="1"
tools:layout_constraintRight_creator="1"
tools:layout_constraintTop_creator="1"
android:layout_marginRight="@dimen/menuicono1layout_marginRight"
app:layout_constraintVertical_bias="0.363" />

<ImageView
    android:id="@+id/imgAyuda"
    android:layout_width="@dimen/menuimgIconos_width"
    android:layout_height="@dimen/menuimgIconos_width"
    android:layout_marginEnd="@dimen/menuicono5layout_marginEnd"
    android:layout_marginRight="@dimen/menuicono5layout_marginRight"
    android:layout_marginTop="@dimen/menuicono5layout_marginTop"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@mipmap/ic_ayuda"
    tools:ignore="ContentDescription,RtlHardcoded" />

<TextView
    android:id="@+id/txtAyuda"
    style="@android:style/Widget.DeviceDefault.Light.TextView.SpinnerItem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/iceberg"
    android:text="@string/menuAyuda"
    android:textSize="@dimen/menuAyuda"
    app:layout_constraintRight_toRightOf="@+id/imgAyuda"
    app:layout_constraintTop_toBottomOf="@+id/imgAyuda"
    tools:ignore="RtlHardcoded"
    tools:layout_constraintRight_creator="1"
    tools:layout_constraintTop_creator="1" />

<ImageView
    android:id="@+id/imgInformacion"
    android:layout_width="@dimen/menuimgIconos_width"
    android:layout_height="@dimen/menuimgIconos_width"
    android:layout_marginEnd="@dimen/menuicono2layout_marginEnd"
    android:layout_marginLeft="@dimen/menuicono2layout_marginLeft"
    app:layout_constraintBottom_toBottomOf="@+id/linearLayout"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="@+id/imgConfiguracion"
    app:layout_constraintRight_toLeftOf="@+id/linearLayout"
    app:layout_constraintTop_toTopOf="@+id/linearLayout"
    app:layout_constraintVertical_bias="0.566"
    app:srcCompat="@mipmap/ic_informacion"
    tools:ignore="ContentDescription,RtlHardcoded"
    tools:layout_constraintBottom_creator="1"
    tools:layout_constraintLeft_creator="1"
    tools:layout_constraintRight_creator="1"
    tools:layout_constraintTop_creator="1" />

<ImageView
    android:id="@+id/imgConfiguracion"
    android:layout_width="@dimen/menuimgIconos_width"
    android:layout_height="@dimen/menuimgIconos_width"
    android:layout_marginBottom="@dimen/menuicono3layout_marginBottom"
    app:layout_constraintBottom_toBottomOf="parent"
    app:srcCompat="@mipmap/ic_configuracion"

```

```

tools:ignore="ContentDescription,RtlHardcoded"
tools:layout_constraintBottom_creator="1"
tools:layout_constraintLeft_creator="1"
android:layout_marginLeft="@dimen/menuicono3layout_marginLeft"
app:layout_constraintLeft_toLeftOf="parent"
android:layout_marginStart="@dimen/menuicono3layout_marginStart"
/>

<ImageView
    android:id="@+id/imgGuardados"
    android:layout_width="@dimen/menuimgIconos_width"
    android:layout_height="@dimen/menuimgIconos_width"
    app:srcCompat="@mipmap/ic_carpeta"
    tools:ignore="ContentDescription,RtlHardcoded"
    tools:layout_constraintBottom_creator="1"
    tools:layout_constraintRight_creator="1"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginEnd="@dimen/menuicono4layout_marginEnd"
    app:layout_constraintRight_toRightOf="parent"

android:layout_marginBottom="@dimen/menuicono4layout_marginBottom"
    android:layout_marginRight="@dimen/menuicono4layout_marginRight"
    app:layout_constraintLeft_toRightOf="@+id/linearLayout"
    android:layout_marginLeft="@dimen/menuicono4layout_marginLeft"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="@dimen/menuicono4layout_marginTop"
    app:layout_constraintHorizontal_bias="0.872"
    app:layout_constraintVertical_bias="0.969" />

<TextView
    android:id="@+id/txtGuardados"

style="@android:style/Widget.DeviceDefault.Light.TextView.SpinnerItem"
    android:layout_width="@dimen/menutexto4layout_width"
    android:layout_height="@dimen/menutexto4layout_height"

android:layout_marginBottom="@dimen/menutexto4layout_marginBottom"
    android:fontFamily="@font/iceberg"
    android:text="@string/menuGuardados"
    android:textSize="@dimen/menuGuardar"
    app:layout_constraintBottom_toBottomOf="@+id/linearLayout"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintLeft_toRightOf="@+id/imgGuardados"
    app:layout_constraintRight_toRightOf="@+id/txtAyuda"
    app:layout_constraintTop_toTopOf="@+id/imgGuardados"
    app:layout_constraintVertical_bias="0.0"
    tools:ignore="RtlHardcoded"
    tools:layout_constraintBottom_creator="1"
    tools:layout_constraintLeft_creator="1"
    tools:layout_constraintRight_creator="1"
    tools:layout_constraintTop_creator="1" />

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="@dimen/menutextoslayout_width"
    android:layout_height="@dimen/menutextoslayout_height"
    android:orientation="vertical"
    android:weightSum="1"
    tools:ignore="RtlHardcoded"
    tools:layout_constraintLeft_creator="1"
    tools:layout_constraintBottom_creator="1"

```

```

        app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="@dimen/menutextoslayout_marginBottom"
    app:layout_constraintLeft_toRightOf="@+id/imgConfiguracion"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="@dimen/menutextoslayout_marginTop"
    app:layout_constraintVertical_bias="0.967"
    android:layout_marginLeft="@dimen/menutextoslayout_marginLeft">

    <TextView
        android:id="@+id/txtIniciar"

    style="@android:style/Widget.DeviceDefault.Light.TextView.SpinnerItem"
        android:layout_width="@dimen/menutexto1layout_width"
        android:layout_height="@dimen/menutexto1layout_height"
        android:fontFamily="@font/iceberg"
        android:text="@string/menuInicio"
        android:textSize="@dimen/menuInicio" />

    <TextView
        android:id="@+id/txtInformacion"

    style="@android:style/Widget.DeviceDefault.Light.TextView.SpinnerItem"
        android:layout_width="match_parent"
        android:layout_height="@dimen/menutexto2layout_height"
        android:fontFamily="@font/iceberg"
        android:text="@string/menuInformacion"
        android:textSize="@dimen/menuInformacion" />

    <TextView
        android:id="@+id/txtConfiguracion"

    style="@android:style/Widget.DeviceDefault.Light.TextView.SpinnerItem"
        android:layout_width="match_parent"
        android:layout_height="@dimen/menutexto3layout_height"
        android:fontFamily="@font/iceberg"
        android:text="@string/menuConfiguracion"
        android:textSize="@dimen/menuConfiguracion" />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

Este archivo layout permite crear la interface de usuario del menú donde se encuentran las diferentes componentes como botones, TextView, ImageView y cuentan con unos identificadores para dimensiones y nombres para en caso de una actualización se puedan actualizar para multilinguaje y cambio de dimensiones.

Anexo. H

Manual de la Aplicación de “Audiometría de Tonos Móvil”

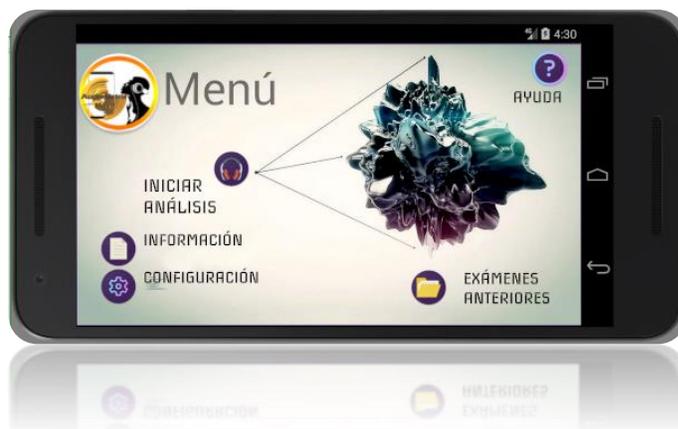
Este manual esta realizado, como un instructivo que paso a paso le va a ir guiando por toda la aplicación, donde se encontrara una breve explicación del funcionamiento de cada actividad, además mencionar que este manual va dirigido a todo usuario final.



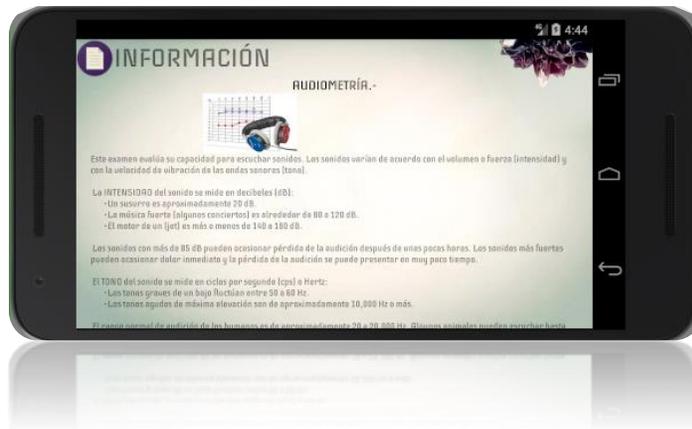
1. Ejecute la aplicación de “AUDIOMETRÍA DE TONOS MÓVIL” en el dispositivo Android y encontrara el video introductorio de la aplicación, a continuación, la previsualización.



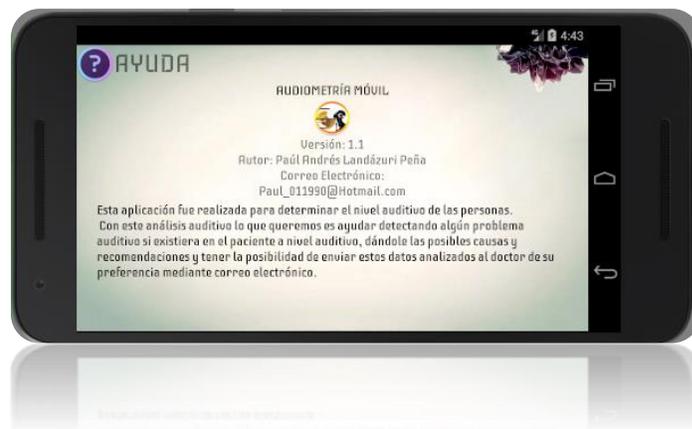
2. Luego se abrirá la ventana de “MENÚ” de la aplicación, en esta ventana podrá observar las distintas actividades a las cuales se puede acceder, a continuación, una previsualización del Menú.



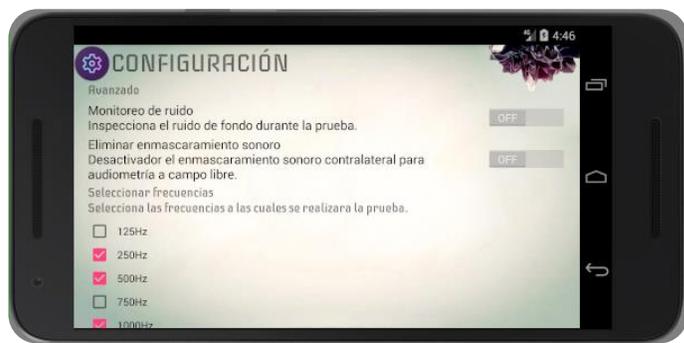
- Actividad de “INFORMACIÓN”, al ingresar en esta ventana podrá obtener información sobre audiometría de tonos puro, consejos sobre cómo usar la aplicación de audiometría, y una explicación de cómo interpretar los resultados.



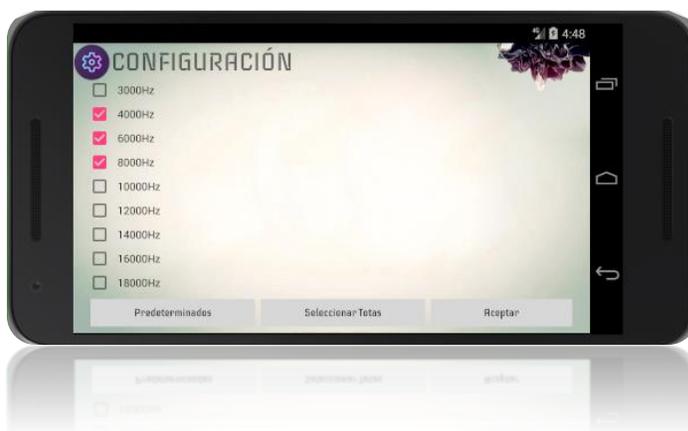
- Actividad de “AYUDA”, en esta ventana podrá obtener información de ayuda para la aplicación, versión de la aplicación, datos de su creación y la página web con información general sobre la aplicación.



- Actividad de “CONFIGURACIÓN”, en esta ventana se podrá realizar la configuración de los parámetros de funcionamiento de la aplicación.



- Para escoger las frecuencias se realizaran la prueba de audiometría, se debe activar las casillas al frente de cada frecuencia y colocar ACEPTAR.



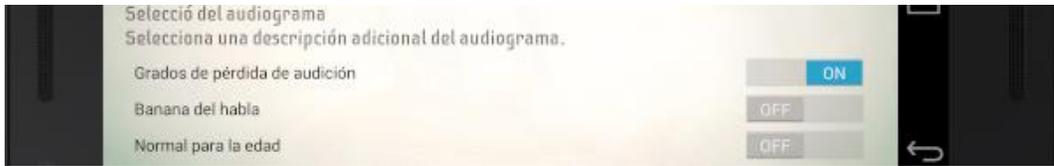
RECORDATORIO

Si desea regresar a las frecuencias predeterminadas. Presione el botón **“Predeterminadas”** en esta misma ventana de configuración y luego presiones aceptar.

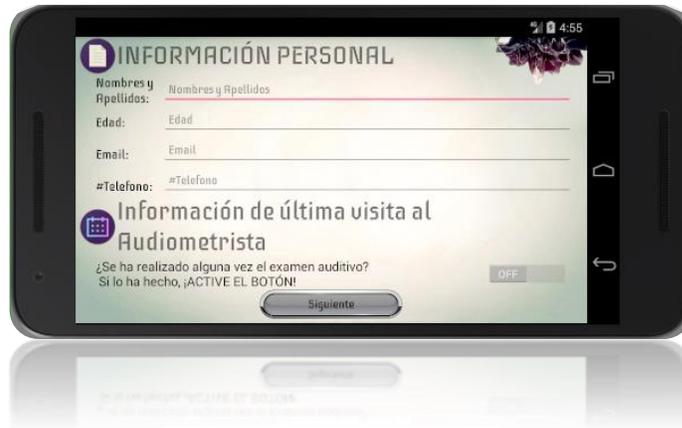
- Para activar o desactivar la ventana donde se pide información del usuario, solo es necesario colocar el switchen ON u OFF respectivamente.



- Para escoger como se presentaran los resultados, se debe escoger activar el switch entre los dos distintos diagramas de audiometría que pueden ser:
 - Niveles de audición
 - Banana auditiva

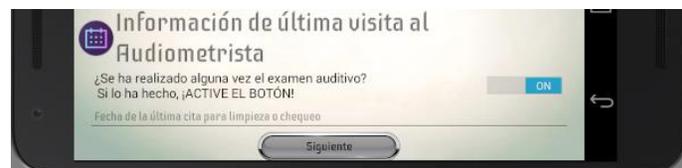


6. Actividad “INICIO PRUEBA AUDITIVA”, al ingresar a esta actividad ingresará a una ventana, donde le pedirá información básica para la audiometría.



ADICIONAL

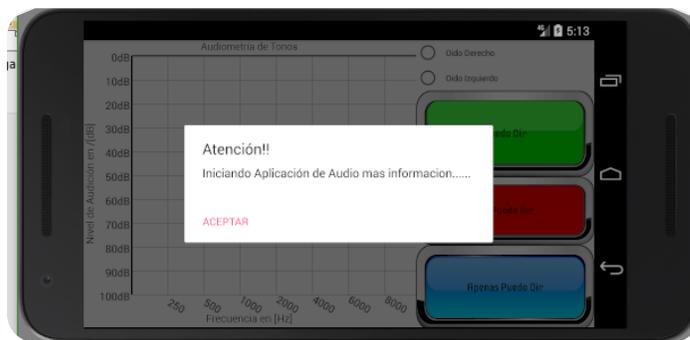
Si usted alguna vez ha ido a un especialista en audición, active la casilla para que pueda ingresar la fecha de su ultimo visita esta opción es opcional.



7. Seguidamente luego de colocar siguiente en la actividad anterior, se abrirá la ventana de “Recomendaciones”, para antes de realizarse el examen, aquí encontrará información de ayuda como, por ejemplo: Que tipo de auriculares usar, como colocarse los auriculares correctamente, además, donde, como y a qué hora poder realizarse el examen.



8. Luego de haber revisado las recomendaciones y presionado el botón de iniciar prueba, se abrirá una nueva actividad, en esta ventana se abrirá un cuadro donde se explicará cómo realizar el examen audiométrico y cómo debe interactuar con la aplicación.



9. Al finalizar la Prueba audiometrica se abrirá la ventana de resultados donde le presentará el diagrama del oído izquierdo y derecho con los niveles en los cuales debe estar tu audición.

10. Actividad de “Guardar Exámenes”, en esta actividad podrá revisar los resultados de las pruebas audiométricas realizadas anteriormente y escogerlas por usuario o fecha, para poderlas comparar.

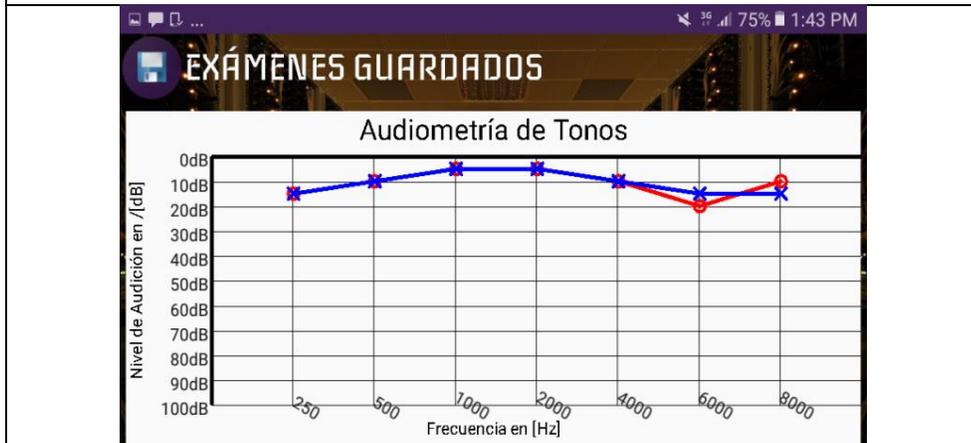
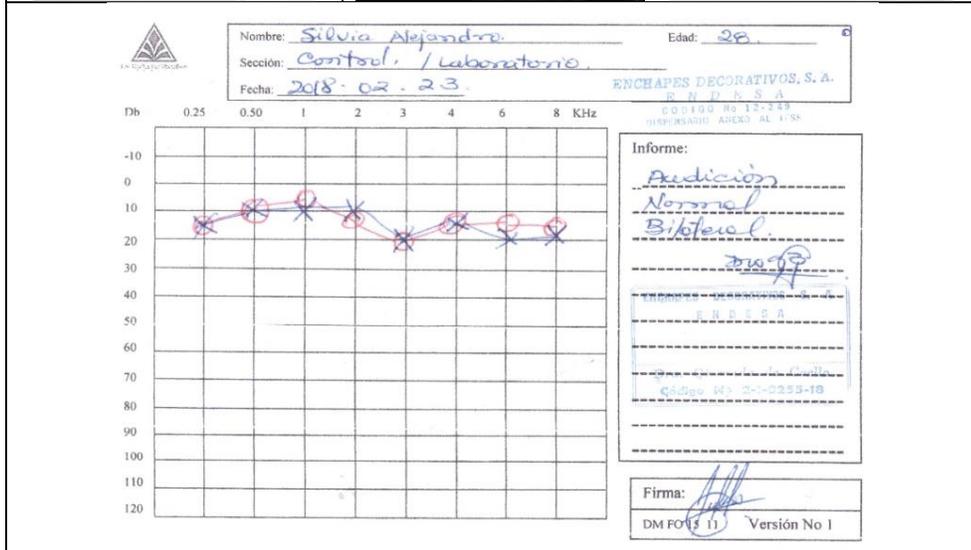


Anexo. I

Prueba Paciente 1

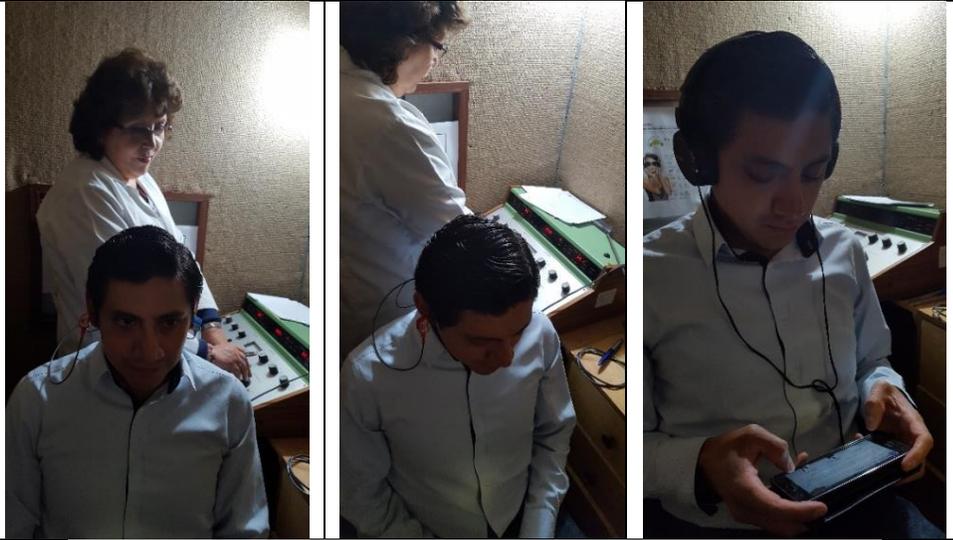


PACIENTE
1

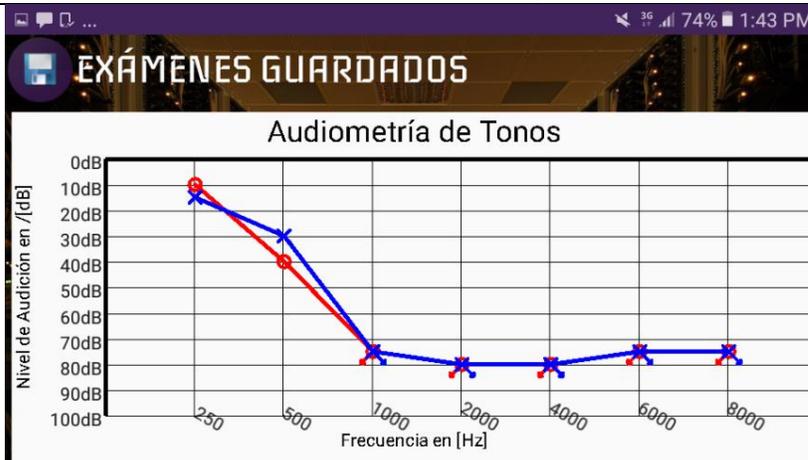
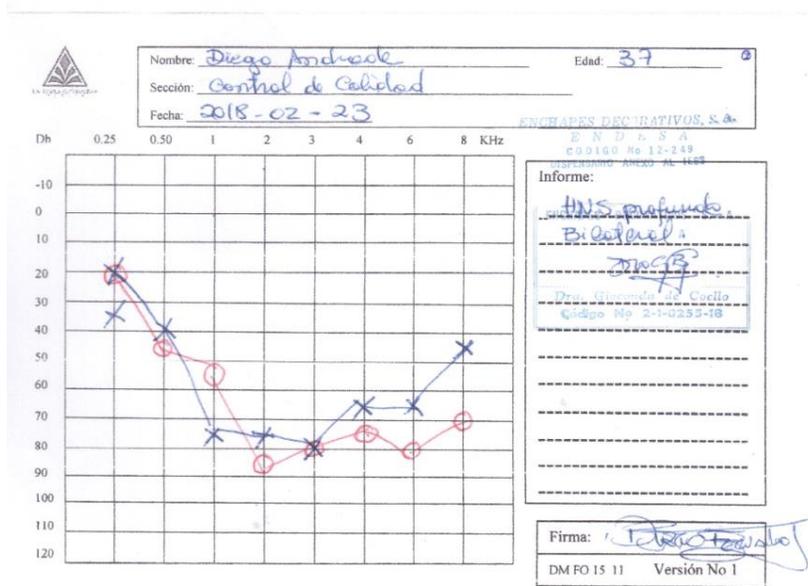


Elaborado por: Mi Autoría

Prueba Paciente 2



PACIENTE
2

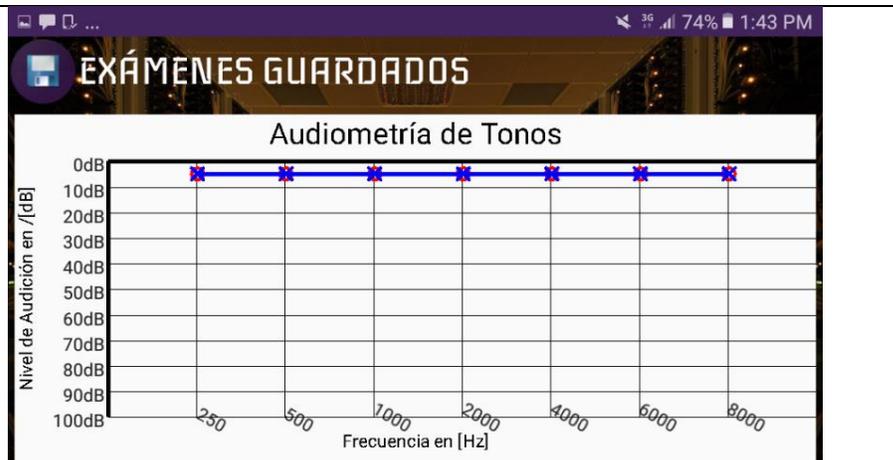
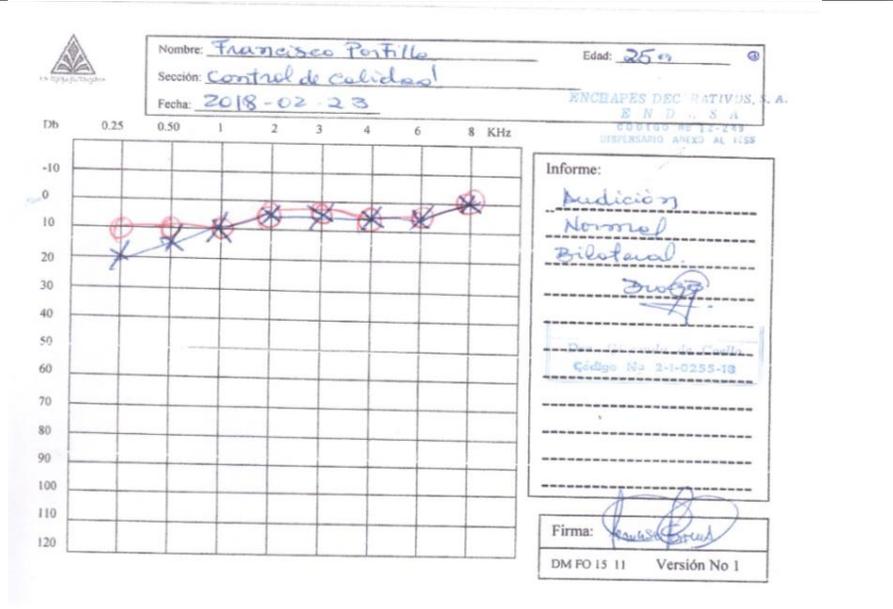


Elaborado por: Mi Autoría

Prueba Paciente 3



PACIENTE
3

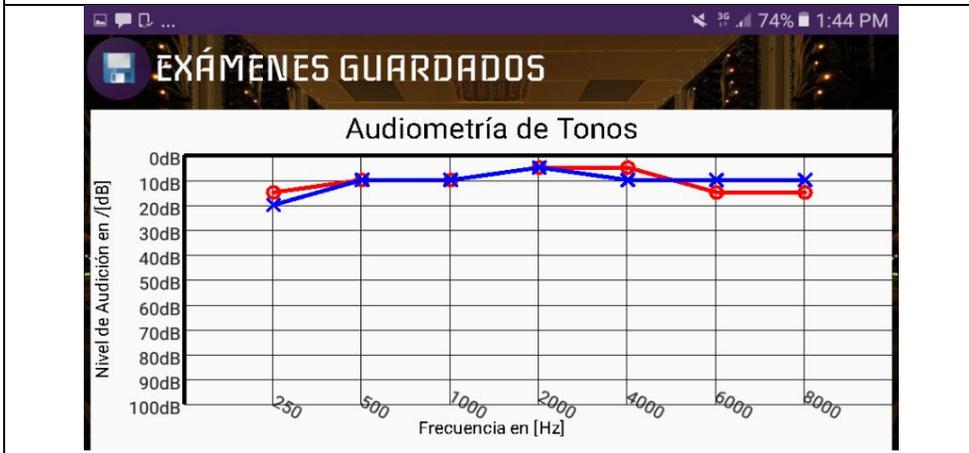
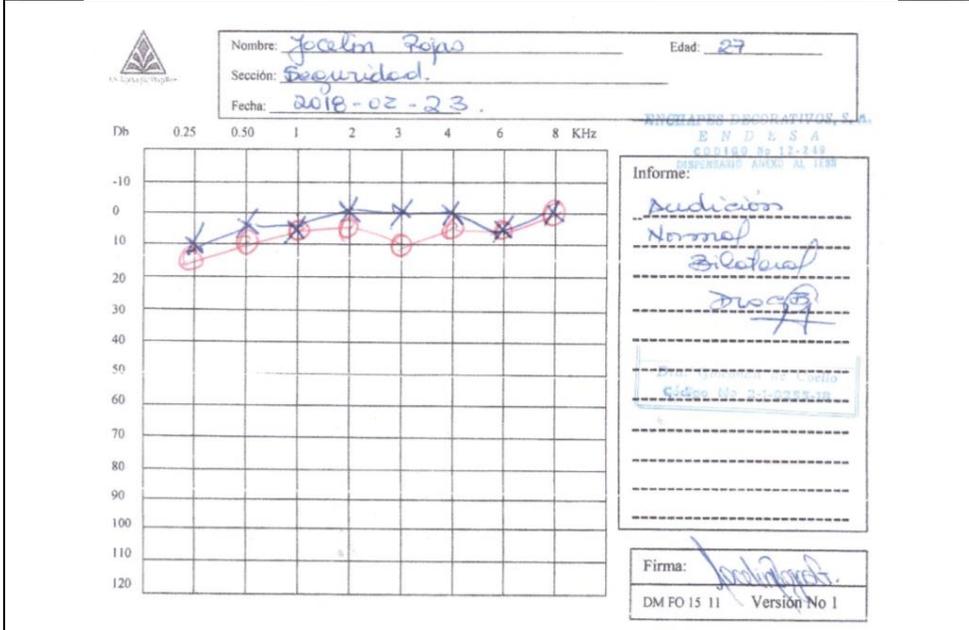


Elaborado por: Mi Autoría

Prueba Paciente 4

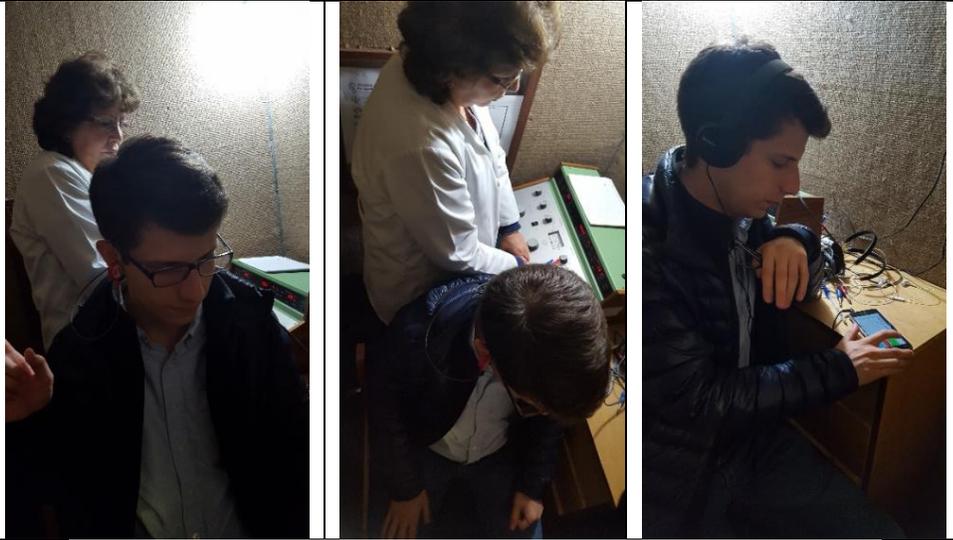


PACIENTE
4

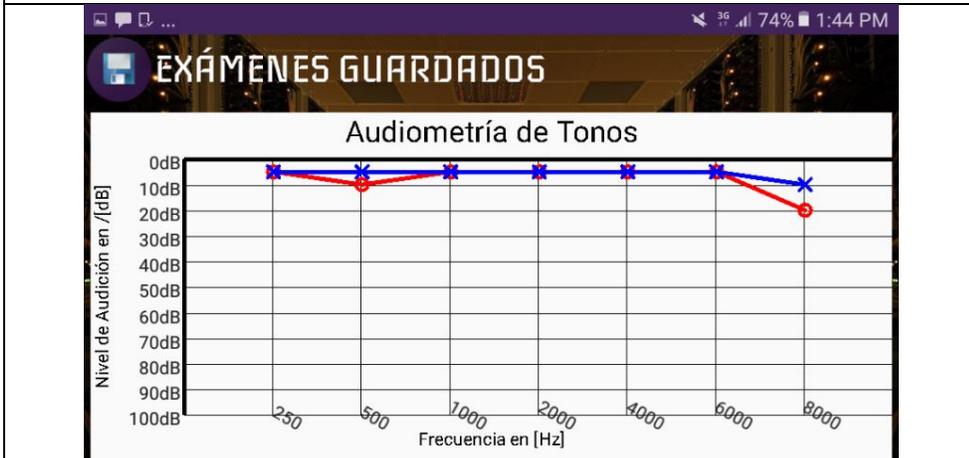
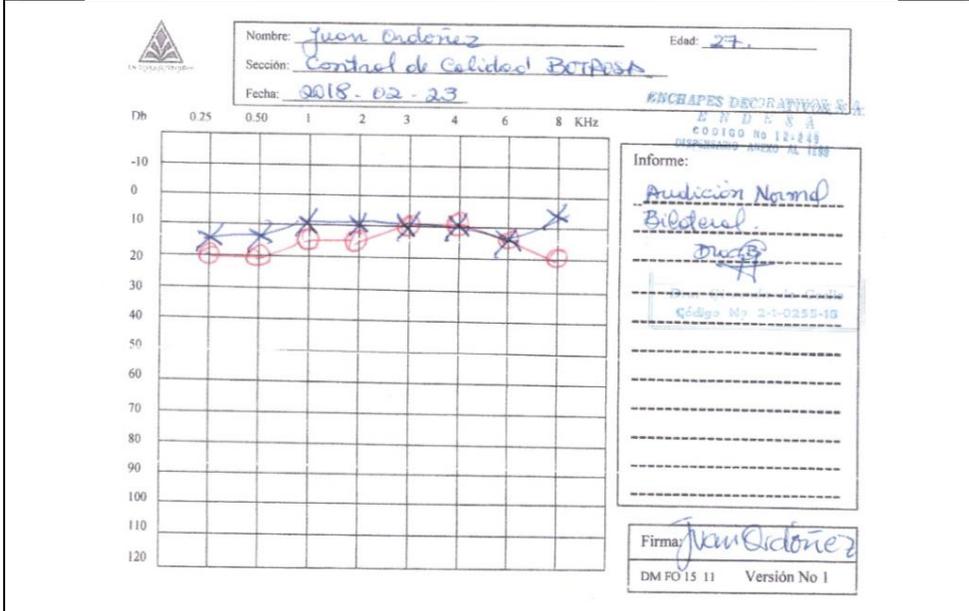


Elaborado por: Mi Autoría

Prueba Paciente 5



PACIENTE
5

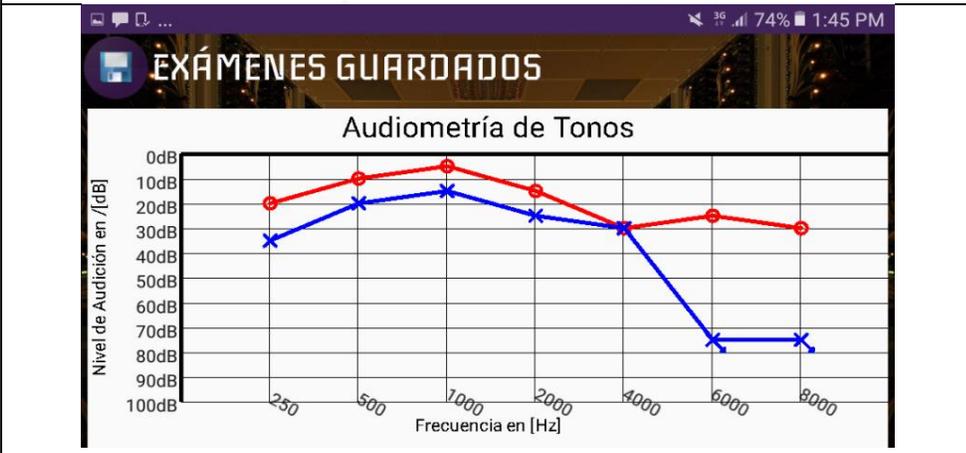
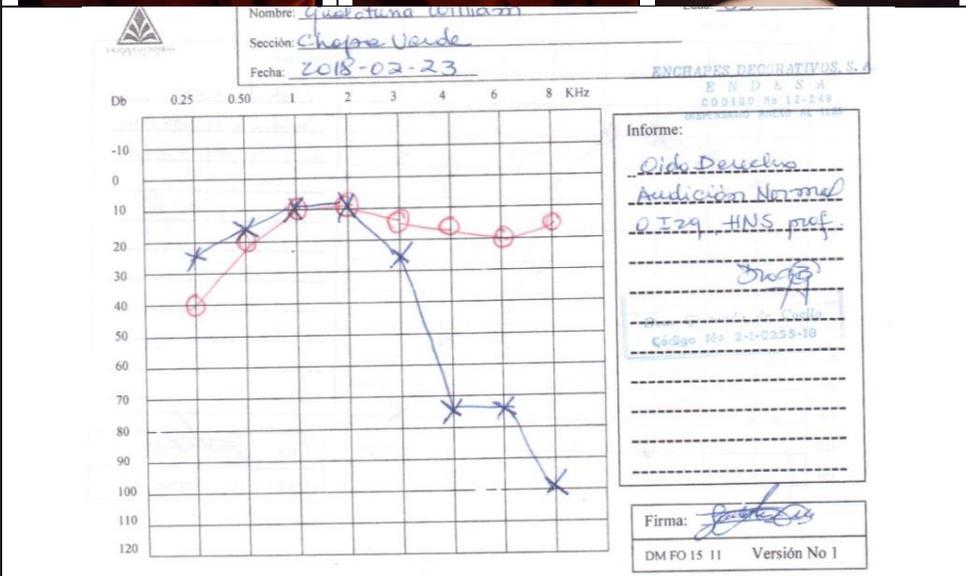


Elaborado por: Mi Autoría

Prueba Paciente 6



PACIENTE
6



Elaborado por: Mi Autoría

Prueba Paciente 7



PACIENTE
7

