RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO DE SONIDO.
2. **TÍTULO:** APLICACIÓN MULTIMEDIA INTERACTIVA PARA DISPOSITIVOS MÓVILES, DE ANÁLISIS DE PARÁMETROS ACÚSTICOS.
3. **AUTOR:** Sergio Leonardo Salamanca Salamanca.
4. **LUGAR:** Bogotá, D.C.
5. **FECHA:** Noviembre de 2013.
6. **PALABRAS CLAVE:** Nivel de Presión Sonora, Nivel Equivalente, Percentil, Retrasos de Señal, Dispositivo Móvil, Aplicación Móvil, Android, Eclipse, Phonegap, jQuery Mobile, Arquitectura de Software.
7. **Descripción del Trabajo:** El objetivo principal de este proyecto es el desarrollo de una aplicación para dispositivos móviles que haga la predicción de niveles de presión sonora, niveles equivalentes, percentiles y retrasos de señal en campo libre, todo lo anterior enfatizando en la arquitectura del software y en un desarrollo iterativo e incremental de la aplicación.
8. **Líneas de Investigación:** Línea de investigación de la Universidad de San Buenaventura: Tecnologías actuales y sociedad. Línea de investigación de la facultad: Análisis y procesamiento de señales.
9. **Fuentes Consultadas:** Belloch Ortí, Consuelo. Desarrollo de aplicaciones multimedia interactivas. Universidad de Valencia: Unidad de Tecnología Educativa (UTE), 2005. Carrión, Antoni. Diseño Acústicos de Espacios Arquitectónicos. Edición de la Universidad Politécnica de Catalunya, 1998. Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado De Desarrollo De Software. PEARSON EDUCACIÓN, 2000. Roxana S. Giandin y Claudia F. Pons. Relaciones entre Casos de Uso en el Unified Modeling Language. Universidad Nacional de La Plata, 2010. Sommerville, Ian. Ingeniería de Software. Pearson Educación S.A, 2005.
10. **Contenidos:** Las aplicaciones para dispositivos móviles se han convertido en utilidades perfectas para la productividad, con estas herramientas las personas pueden realizar cálculos o simular campos de trabajo. La aplicación pretende colocar a disposición una herramienta que sea capaz de generar diferentes puntos de fuentes y puntos de medición para lograr predecir el comportamiento de la energía sonora que se irradia en algún punto en el espacio. Mediante el proyecto no solo se pretende dejar un cuerpo código fuente y su aplicativo, si no todo un producto que contenga el modelo de la arquitectura del sistema, el modelo del diseño, especificaciones no funcionales y casos de prueba que permitan a los usuarios utilizar y modificar el sistema de generación en generación.
11. **Metodología:** La investigación tiene un enfoque **Empírico – Analítico** cuyo interés es el desarrollo ingenieril de una aplicación interactiva enfatizada en una herramienta para los ingenieros de sonido.
12. **Conclusiones**: El algoritmo se desarrolló teniendo en cuenta las formulas teóricas en base a campo libre, este algoritmo no fue desarrollado para medir ruido ambiental por lo que no se utilizaron factores como las absorciones del suelo o la del aire. Por esta razón la validación en el software CadnaA sirvió como un precedente o un comparador pero en ningún momento la funcionalidad de la aplicación se compara a la funcionalidad de CadnaA. Trabajando con el ANDROID SDK se logró la implementación de la aplicación para todos los dispositivos móviles que cumplan con los requerimientos del sistema.

APLICACIÓN MULTIMEDIA INTERACTIVA PARA DISPOSITIVOS MÓVILES, DE ANÁLISIS DE PARÁMETROS ACÚSTICOS

SERGIO LEONARDO SALAMANCA SALAMANCA

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SONIDO

BOGOTÁ D.C. 2013

APLICACIÓN MULTIMEDIA INTERACTIVA PARA DISPOSITIVOS MÓVILES, DE ANÁLISIS DE PARÁMETROS ACÚSTICOS

SERGIO LEONARDO SALAMANCA SALAMANCA

Trabajo presentado como requisito parcial para optar al título de profesional en Ingeniería de Sonido

Asesor:
Ingeniero Miguel Ángel Olivares

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SONIDO

BOGOTÁ D.C. 2013

Este trabajo de grado, en la modalidad de Proyecto de Investigación, es aceptado como uno de los requisitos para obtener el título de Ingeniero de Sonido en la Universidad de San Buenaventura Sede Bogotá.

Nota de Aceptación.

_____

_____

_____

_____

_____

Ingeniero Miguel Olivares
Tutor

_____

Jurado

_____

Jurado

Agradecimientos.

Agradezco a mi familia que me apoyo en todo momento mientras realizaba la investigación, a mi Madre Clara Salamanca a mi Padre Fabio Salamanca y a mi Hermano Fabio Andrés Salamanca ya que sin ellos esto no se hubiera podido haber realizado, Agradezco al Ingeniero Miguel Ángel Olivares por brindarme su apoyo, tiempo, conocimiento y sus consejos durante este proceso, de igual forma al Licenciado Alexander Ortega, al Ingeniero Nelson Rosas y a mi amiga Ana Lucia Díaz quienes me brindaron apoyo en la fundamentación del proyecto.

Por ultimo quisiera agradecer a la familia Sabogal quienes estuvieron pendientes de mí y de lo que necesitase durante el tiempo que realice la investigación.

TABLA DE CONTENIDO

INDICE DE FIGURAS 10
INTRODUCCIÓN 14
1 PLANTEAMIENTO DEL PROBLEMA 14
1.1 ANTECEDENTES 14
1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA 16
1.3 PREGUNTA PROBLEMA 16
1.4 JUSTIFICACIÓN 16
1.5 OBJETIVOS 17
1.5.1 Objetivo General 17
1.5.2 Objetivos Específicos 17
1.6 ALCANCES Y LIMITACIONES DEL PROYECTO 17
1.6.1 Alcances 17
1.6.2 Limitaciones 17
2 MARCO DE REFERENCIA 18
2.1 MARCO TEÓRICO – CONCEPTUAL 18
2.1.1 Aplicación multimedia Interactiva 18
2.1.2 Dispositivo Móvil 18
2.1.3 Smartphone 18
2.1.4 Tablet 18
2.1.5 Plataformas y lenguajes soportados 19
2.1.6 Android 19
2.1.7 Android SDK 19
2.1.8 Eclipse 20
2.1.9 Phonegap 20
2.1.10 jQuery Mobile 20
2.1.11 Java 20

2.1.12 Java Script 20
2.1.13 HTML 5 21
2.1.14 Proceso de Desarrollo de Software 21
2.1.15 Proceso unificado 21
2.1.16 Arquitectura de Software 21
2.1.17 Casos de Uso 22
2.1.18 Modelo de Análisis 22
2.1.19 Modelo de Diseño 22
2.1.20 Modelo de Implementación 23
2.1.21 Modelo de Prueba 23
2.1.22 UML 23
2.1.23 Atributos de calidad 25
2.1.24 Producto 25
2.1.25 Propagación del sonido en campo libre 25
2.1.26 Nivel Continuo Equivalente 26
2.1.27 Niveles Percentiles 26
2.1.28 CadnaA 27
3 METODOLOGÍA 27
3.1 ENFOQUE DE LA INVESTIGACIÓN 28
3.2 LÍNEA DE INVESTIGACIÓN 28
3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN. 28
3.4 HIPÓTESIS 29
3.5 VARIABLES 29
3.5.1 Variables Independientes 29
3.5.2 Variables Dependientes 29
4 DESARROLLO INGENIERIL 30

4.1 MODELO DE CASOS DE USO 30
4.2 ATRIBUTOS DE CALIDAD 33
4.3 MODELO DE ANALISIS 34
4.4 MODELO DE DISEÑO 35
4.5 MODELO DE IMPLEMENTACIÓN 35
4.5.1 Diagrama de Componentes 36
4.6 DIAGRAMA DE SECUENCIA 43
4.7 PRUEBAS DE USABILIDAD 46
4.7.1 Pruebas de Hardware 46
4.7.2 Casos de prueba 46
4.8 VALIDACIÓN CadnaA 47
4.8.1 Nivel de Presión Sonora 47
4.8.2 Nivel de Presión Sonora Total 51
4.8.3 Nivel Equivalente 53
5 ANALISIS DE RESULTADOS 56
5.1 EVALUACIÓN DE USABILIDAD 56
5.1.1 Pruebas de Hardware 56
5.1.2 Casos de Prueba 58
5.2 ATRIBUTOS DE CALIDAD 62
5.2.1 Desempeño 62
5.2.2 Confiabilidad 65
5.3 REQUERIMIENTOS DE SISTEMA 66
5.4 VALIDACIÓN CADNA 67
5.4.1 Nivel de Presión Sonora 67
5.4.2 Nivel de Presión Sonora Total 68
5.4.3 Nivel Equivalente 69
6 CONCLUSIONES 70

7 RECOMENDACIONES 71
GLOSARIO 72
BIBLIOGRAFÍA 75

INDICE DE FIGURAS

Figura 1: Proceso de desarrollo de Software 21

Figura2: Representación Casos de Uso..... 24

Figura 3: Generalización representación UML 24

Figura 4: Extensión representación UML 24

Figura 5: Inclusión representación UML..... 25

Figura 6: Fases de Desarrollo de Software 27

Figura 7: Modelo del Proceso Unificado..... 30

Figura 8: Caso de Uso, Ingresar Objetos de Medición 31

Figura 9: Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos 31

Figura 10: Casos de Uso: Ingresar Tiempos de Medición 32

Figura 11: Caso de Uso: Ingresar Percentil deseado 33

Figura 12: Modelo de Análisis 34

Figura 13: Modelo de Diseño 35

Figura 14: Diagrama de Componentes 36

Figura 15: Componente Numero de Fuentes. 37

Figura 16: Componente Numero de Oyentes. 37

Figura 17: Componente Información. 37

Figura 18: Componente Swipepad 38

Figura 19: Componente Coordenadas. 38

Figura 20: Componente Parámetros Fuente. 38

Figura 21: Componente SPL total. 39

Figura 22: Componente SPL Fuente - Oyente 39

Figura 23: Componente Tiempos de Medición 40

Figura 24: Componente Leq 40

Figura 25: Componente Alerta Leq 41

Figura 26: Componente Percentil..... 41

Figura 27: Componente Alerta Percentil 41

Figura 28: Diagrama de Componentes del Componente Guide.html 42

Figura 29: Diagrama de Secuencia - Caso de Uso: Ingresar Objetos de Medición. 43

Figura 30: Diagrama de Secuencia - Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos. 44

Figura 31: Diagrama de Secuencia - Caso de Uso: Ingresar Tiempo Total de Medición, Caso de Uso: Ingresar Tiempo de Fuentes. 45

Figura 32: Diagrama de Secuencia - Caso de Uso: Ingresar Percentil deseado 45

Figura 33: Representaciones graficas de CadnaA 47

Figura 34: Plano 2D, medición en CadnaA SPL. 48

Figura 35: Configuración de la fuente Puntual 48

Figura 36: Coordenadas Fuente puntual 49

Figura 37: Tabla de Resultados CadnaA SPL 49

Figura 38: Resultados Aplicación SPL fuente – Oyente 50

Figura 39: Plano 2D, medición CadnaA SPL total 51

Figura 40: Tabla de Resultados CadnaA SPL total 52

Figura 41: Resultados Aplicación SPL total 53

Figura 42: Plano 2D, medición CadnaA Niveles Equivalentes 54

Figura 43: Tabla de Resultados CadnaA Nivel equivalente 55

Figura 44: Resultados Aplicación Nivel equivalente 55

Figura 45: Vista de la APP instalada en un SAMSUNG Galaxy ACE 56

Figura 46: Consecuencias de cumplir con el Caso de Uso: Ingresar Objetos de Medición. 59

Figura 47: Consecuencias de incumplir con el Caso de Uso: Ingresar Objetos de Medición. 59

Figura 48: Consecuencias de cumplir con el Caso de Uso: Ingresar parámetros y coordenadas de objetos de medición. 60

Figura 49: Consecuencias de incumplir con el Caso de Uso: Ingresar parámetros y coordenadas de objetos de medición. 60

Figura 50: Consecuencias de cumplir con los Caso de Uso: Ingresar tiempos de medición. 61

Figura 51: Consecuencias de incumplir con los Caso de Uso: Ingresar tiempos de medición. 61

Figura 52: Aplicación abierta por primera vez en perfil fuera de línea 66

INDICE DE TABLAS

Tabla 1: Caso de Uso Ingresar Objetos de Medición 31

Tabla 2: Caso de Uso Ingresar Parámetros y Coordenadas de Objetos 32

Tabla 3: Caso de Uso Ingresar Tiempo Total de Medición 32

Tabla 4: Caso de Uso Ingresar Tiempo de Fuentes 32

Tabla 5: Caso de Uso Ingresar Percentil deseado 33

Tabla 6: Coordenadas, receptores medición SPL. 49

Tabla 7: Coordenadas de Fuentes Sonoras. 51

Tabla 8: Coordenadas, receptores medición SPL total. 52

Tabla 9: Coordenadas de Fuentes, Medición de nivel equivalente 54

Tabla 10: Coordenadas de Receptores, Medición de nivel equivalente 54

Tabla 11: Tiempos de Operación de Fuentes, Medición de Nivel Equivalente 55

Tabla 12: Prueba de Instalación en dispositivos móviles 57

Tabla 13: Prueba de Configuración. 57

Tabla 14: Prueba de Tensión. 58

Tabla 15: Tiempos de Apertura por Primera Vez. 62

Tabla 16: Tiempos de Apertura. 63

Tabla 17: Tiempos de cambio de ventanas por medio de Swipepad. 64

Tabla 18: Tiempos de cambio de ventanas por medio de Pad. 65

Tabla 19: Requerimientos del Sistema. 67

Tabla 20: Niveles de Presión Sonora CadnaA - Aplicación. 68

Tabla 21: Niveles de Presión Sonora total CadnaA - Aplicación 68

Tabla 22: Niveles Equivalentes CadnaA - Aplicación 69

INTRODUCCIÓN

El campo de la acústica, como muchos otros campos de la ciencia, es extremadamente amplio. No en vano engloba disciplinas tan diversas como la acústica ambiental, la acústica musical, la psico-acústica y la acústica arquitectónica, entre otras. El proyecto se centra exclusivamente en el desarrollo de una aplicación multimedia interactiva que realice la predicción de parámetros acústicos que intervienen en campo libre, a la hora de realizar un proyecto.

El análisis de parámetros sonoros entre espacio cerrado a campo libre varía ya que en campo libre no intervienen factores como superficies reflejantes así que no se encuentran fenómenos sonoros como las reflexiones, entre otros, pero aun así hay parámetros importantes a medir en este entorno como por ejemplo la disipación del sonido por el espacio, los retardos de señales por distancia o la suma de varias señales en un punto y lo que esto representa a un oyente.

Las aplicaciones móviles proveen acceso al usuario o cliente a una enorme variedad de herramientas o información en cualquier lugar con tan solo tener un dispositivo móvil, con el proyecto se pretende generar una herramienta para que los usuarios interesados en predecir parámetros acústicos puedan tener una aplicación móvil que desarrolle esto a su disposición.

El proyecto no solo cuenta con el aplicativo ya que el producto se basa en un desarrollo iterativo e incremental que contiene todos los análisis y diagramas del software para que futuros desarrolladores puedan seguir aumentando las posibilidades de esta aplicación ya conociendo su arquitectura.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

El mercado de dispositivos móviles dio un vuelco al emerger las primeras aplicaciones móviles a finales del siglo pasado, estas aparecieron en forma de agendas, juegos,

editores de ring toné etc. cumplían funciones muy elementales y su diseño era bastante simple.[1]

Gracias a las innovaciones en tecnología WAP (protocolo de aplicaciones inalámbricas) y el gran desarrollo de celulares cada vez más sofisticados y potentes la evolución de las aplicaciones se dio rápidamente, [2] pero su expansión se dio cuando la empresa Apple lanza el iPhone por el año 2007 continuo de la aparición del sistema operativo Android en el mismo año. Seguido a esto Apple lanza el AppStore el 10 de julio de 2008, un lugar en el cual el usuario puede acceder, bajar o pagar por las aplicaciones de su interés, y Google quien compro Android[3] lanza su Play Store el 22 de octubre del mismo año el cual tiene las mismas funciones del AppStore.

El mercado de Apple o Google no se basaba en desarrollar aplicaciones así que facilitaron la posibilidad de que cualquier persona tuviera acceso a los SDK (Software Development Kits), así dejando entrar a nuevas empresas para ofertar sus productos desarrollados para Android, iOS o para otros sistemas que saldrían para hacerle competencia a estas dos potencias.[4]

En el mercado hay cantidad de aplicaciones como cantidad de necesidades, hay aplicaciones que son útiles para los ingenieros de sonido en el mercado por ejemplo para medir SPL como la aplicación de Logitech Inc. "UE SPL" y sonómetros como el "Sound Meter" de Smart Tools co. Entre otras muchas otras aplicaciones.[5]

En Colombia iniciativas gubernamentales están apoyando y capacitando el sector de desarrollo en tecnología de la comunicación como lo son los desarrolladores de aplicaciones móviles, además estas campañas han ayudado en la consolidación de 60

---

[1] Basterretche, Juan. *Dispositivos Móviles.* Trabajo de Adscripción, Universidad Nacional del Nordeste, Argentina 2007.

[2] Alonso Baz, Arturo; Álvarez Rodríguez, María y GARCIA BANIELLO, Rosana. *Dispositivos Móviles*, Universidad de Oviedo, Ingeniería de Telecomunicaciones, Pg. 3.

[3] Beauperthuy Taibo, José Luis y Malavé Polanco, Kristel. *"Android" el sistema operativo de Google para dispositivos móviles*. Fundación Miguel Unamuno y Jugo Venezuela, Vol. 7, núm. 19, 2011, Pg. 79-96.

[4] Martínez González, Felipe Luis. *Aplicaciones para dispositivos móviles*. Universidad Politécnica de Valencia, 2010 – 2011.

[5] Pérez, José; Fernández Mireya y Ramos Juan. *Análisis de Características Técnicas en Dispositivos Móviles Aplicado a la Instrumentación.* Cuarto congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica, 2010.

empresas nacionales jóvenes dedicadas a la creación de software. [6] El ministerio de Tecnologías de la información y las comunicaciones MinTIC consciente de que el Gobierno es el mayor demandante de aplicaciones, está en la tarea de fortalecer este ecosistema. [7]

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

Los ingenieros de sonido deben abordar el campo de la acústica y poder medir y analizar claramente el comportamiento sonoro en cualquier recinto o lugar en donde se quiera realizar una investigación al detalle, La tecnología que ofrecen los dispositivos móviles, nos permiten tener aplicaciones a la mano que pueden ser herramientas útiles para evaluar parámetros de manera rápida, es por esto que se desarrollara una aplicación multimedia interactiva, que pueda generar distintas configuraciones entre puntos de medición y fuentes sonoras para evaluar parámetros acústicos en campo libre.

1.3 PREGUNTA PROBLEMA

¿Cómo realizar una herramienta interactiva que por medio de la predicción de parámetros acústicos permita al usuario analizar el comportamiento de estos, en campo libre?

1.4 JUSTIFICACIÓN

Las aplicaciones para teléfonos inteligentes y tabletas se han convertido en utilidades perfectas para el entretenimiento o la productividad, con estas herramientas los ingenieros de sonido pueden tener a su disposición aplicaciones que puedan mejorar la productividad al momento de realizar cálculos o de simular campos de trabajo. La aplicación que se va a desarrollar pretende colocar a disposición una herramienta que sea capaz de generar diferentes puntos de fuentes y puntos de medición para lograr predecir el comportamiento de la energía que se irradia en un punto.

[6] Las TIC para en el sector educativo, Impacto y lecciones aprendidas de programas apoyados por el IICD
[7] *Que es apps.co. 2013,de* https://apps.co/about/apps-co-que-es/

1.5 OBJETIVOS

1.5.1 Objetivo General

Desarrollar una aplicación multimedia interactiva en una plataforma de dispositivo móvil, la cual permita predecir parámetros acústicos evaluados en un entorno de campo libre.

1.5.2 Objetivos Específicos

- Diseñar y desarrollar el algoritmo de la aplicación, que calcule los parámetros acústicos a medir (SPL, SPL total, Nivel equivalente, Percentiles y retrasos de señal) de forma abierta e interactiva.
- Implementar la aplicación para dispositivo móvil
- Validar el sistema desarrollado comparando su funcionamiento con el software CadnaA.

1.6 ALCANCES Y LIMITACIONES DEL PROYECTO

1.6.1 Alcances

- Con el desarrollo del proyecto tanto los estudiantes y profesionales de ingeniería de sonido como las demás personas interesadas pueden obtener una aplicación donde puedan predecir el comportamiento de parámetros acústicos desde sus dispositivos móviles.

1.6.2 Limitaciones

- Durante el desarrollo de la aplicación se generaran determinadas fuentes y puntos de medición, esta información ocupara espacio que puede superar el espacio que un dispositivo móvil puede almacenar.
- Por esto la cantidad de fuentes y puntos de medición serán limitadas, que dependerán de las características técnicas del dispositivo móvil.
- Introducir todas las variables a la aplicación.

2 MARCO DE REFERENCIA

2.1 MARCO TEÓRICO – CONCEPTUAL

2.1.1 Aplicación multimedia Interactiva

Son aplicaciones que permite interactuar con el ordenador utilizando diferentes códigos en la presentación de la información (texto, imagen, sonido,...). [8]

2.1.2 Dispositivo Móvil

La línea entre lo que es un dispositivo móvil y lo que no lo es puede ser un poco difusa, pero en general, se pueden definir como aquellos micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma. Los ordenadores portátiles no se consideran como dispositivos móviles, ya que consumen más batería y suelen ser un poco más pesados de lo que se espera de algo pensado para llevar siempre encima.

2.1.3 Smartphone

Es un dispositivo entre un teléfono móvil clásico y una agenda electrónica personal ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional pero además incluye características cercanas a las de un ordenador personal. Los teléfonos inteligentes se distinguen por muchas características, entre las que destacan las pantallas táctiles, un sistema operativo así como la conectividad a Internet y el acceso al correo electrónico[9].

2.1.4 Tablet

Es un dispositivo con capacidad de procesamiento de información y navegación en Internet similares o ligeramente inferiores a la de un computador portátil. Sus principales características son batería de larga duración (en el orden de 8 horas),

[8] Consuelo Belloch, Ortí. *Aplicaciones multimedia interactivas: Clasificación*. Unidad de Tecnología Educativa. Universidad de Valencia.

[9] Alonso Baz, Arturo; Álvarez Rodríguez, María y García Baniello, Rosana. *Dispositivos Móviles*. Universidad de Oviedo, Ingeniería de Telecomunicaciones, Pg. 3.

pantalla táctil, bajo peso (alrededor de los 500 gramos) y tamaño (hasta 10”) lo que mejora la portabilidad.[10]

2.1.5 Plataformas y lenguajes soportados

Cada una de las plataformas tiene sus características únicas, no sólo en diseño, procesamiento y arquitectura, sino también a la hora de desarrollar aplicaciones para las mismas, ya que no es lo mismo programar una aplicación para i OS que para Android, entre otras cosas porque no todas las plataformas soportan los mismos lenguajes de programación como java o html5 entre una y otra varia el lenguaje para programar.

- BlackBerry OS.
- Windows Mobile.
- iPhone OS: Para dispositivos móviles como iPad y iPhone.
- Android: Para dispositivos móviles de Samsung.
- Symbian OS

2.1.6 Android

Es un compilador para dispositivos móviles que incluye sistema operativo denominado con el mismo nombre basado en Linux, los desarrolladores de aplicaciones en esta plataforma tienen pleno acceso a la misma framework APIs utilizada por las aplicaciones base y todas las aplicaciones están escritas en lenguaje de programación Java. Esta plataforma es utilizada para dispositivos móviles de Samsung, Sony, LG, Motorola entre otros

2.1.7 Android SDK

Software Developer Kit Es un paquete de software que proporciona Android para los desarrolladores donde se encuentran bibliotecas API y herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android.

[10] Mares, Laura. *Tablets en educación oportunidades y desafíos en políticas uno a uno*. red Latinoamérica portales educativos, 2012, pg 5

2.1.8 Eclipse

Es un entorno de desarrollo multiplataforma para desarrollar aplicaciones en lenguaje java, con la facilidad de implementar librerías y herramientas para este que permitan trabajar en HTML 5 y JavaScript. Eclipse es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

2.1.9 Phonegap

Es un framework Open Source que permite crear aplicaciones móviles utilizando Aplicaciones web estándar para la plataforma que uno desee, (iOS, Android, BlackBerry, Windows Phone, palm webOS, Symbian…).

2.1.10 jQuery Mobile

Es un sistema unificado, basado en una interfaz de usuario HTML5 para que en lugar de escribir aplicaciones únicas para cada dispositivo móvil o sistema operativo, el marco móvil jQuery permite diseñar un sitio web altamente calificado o aplicación que funcione en todas las plataformas de teléfonos inteligentes o tablets.

2.1.11 Java

Es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos.

2.1.12 Java Script

Es un lenguaje de scripting basado en objetos que se pueden integrar directamente en las páginas HTML, permite crear aplicaciones basadas en la Web dinámicas e

interactivas que se ejecutan por completo dentro de un navegador Web - JavaScript es el lenguaje de elección para el desarrollo de aplicaciones.[11]

2.1.13 HTML 5

HyperText Markup Language (**HTML**), es el lenguaje que todos los navegadores usan para presentar información en la World Wide Web (WWW). Este es un lenguaje muy sencillo que se basa en el uso de etiquetas, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares (< >).

2.1.14 Proceso de Desarrollo de Software

Un proceso define quien está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo. En la ingeniería de software el objetivo es construir un producto o mejorar uno existente, un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad, lo que es necesario para que sirva como guía tanto para clientes, usuarios, y desarrolladores.[12]

2.1.15 Proceso unificado

El proceso Unificado es un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software, se resume en tres frases clave: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

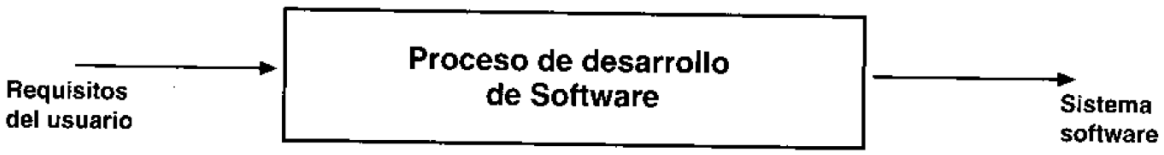

Figura 1: Proceso de desarrollo de Software

Fuente: Jacobson, Ivar. Booch, Grady y Rumbaugh, James. El Proceso Unificado De Desarrollo De Software, Pearson Educación Madrid, 2000.

2.1.16 Arquitectura de Software

[11] Flanagan, David. *Java Script Pocket Reference*. O´Reilly Media Inc. Segunda Edición.
[12] Jacobson, Ivar. Booch, Grady. Rumbaugh, James. *El Proceso Unificado De Desarrollo De Software*, PEARSON EDUCACIÓN Madrid, 2000, prefacio XVI.

El concepto de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades, como las perciben los usuarios y los desarrolladores, y se refleja en los casos de uso. Sin embargo, también se ve influida por muchos otros factores, como la plataforma en la que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión de base de datos, protocolos para comunicaciones en red), consideraciones de implantación, sistemas heredados, y requisitos no funcionales (por ejemplo, rendimiento, fiabilidad). La arquitectura es una vista de diseño completo con las características más importantes resaltadas, dejando los detalles de lado.[13]

2.1.17 Casos de Uso

Describe que hace un sistema pero no específica como lo hace, es una secuencia de acciones que el sistema lleva a cabo para ofrecer algún resultado de valor para un usuario, en pocas palabras los Casos de Uso representan los requisitos funcionales.

Todos los Casos de Uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad total del sistema y son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

2.1.18 Modelo de Análisis

Tiene dos propósitos: refinar los casos de uso con detalle y establecer la asignación inicial de funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento

2.1.19 Modelo de Diseño

Define la estructura estática del sistema en la forma de subsistemas, clases e interfaces y los casos de uso como colaboraciones. En el diseño modelamos el sistema

[13] Jacobson, Ivar. Booch, Grady. Rumbaugh, James. *El Proceso Unificado De Desarrollo De Software*, PEARSON EDUCACIÓN Madrid, 2000, Pg.6.

y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones que se le suponen.[14]

Debe ser mantenido durante todo el ciclo de vida del software.

2.1.20 Modelo de Implementación

Incluye componentes (que representan el código fuente) y la correspondencia de las clases con los componentes.

2.1.21 Modelo de Prueba

Verifica que el sistema implemente de verdad la funcionalidad descrita en los casos de uso y que satisfaga los requisitos del sistema. El modelo de prueba se compone de casos de prueba los cuales definen una colección de entradas, condiciones de ejecución, y resultados esperados, desarrollados para un objetivo concreto, tal como probar un camino concreto a través de un caso de uso, o verificar que se cumple un requisito especifico.

El caso de prueba a de especificar la entrada, el resultado esperado y otras condiciones relevantes para la verificación del escenario del caso de uso[15]

2.1.22 UML

Es un lenguaje estándar para construir, especificar, visualizar y documentar planos de un sistema pero no dicen que modelos se deben crear ni cuando se los debe crear, sus siglas significan Lenguaje unificado de modelado.[16]

En la representación gráfica de los Casos de Uso se utilizan actores que representan a los usuarios y Casos de Uso que representan los requisitos funcionales.

---

[14] Jacobson, Ivar. Booch, Grady. Rumbaugh, James. *El Proceso Unificado De Desarrollo De Software*, PEARSON EDUCACIÓN Madrid, 2000, Pg. 205.

[15] Jacobson, Ivar. Booch, Grady. Rumbaugh, James. *El Proceso Unificado De Desarrollo De Software*, PEARSON EDUCACIÓN Madrid, 2000, Pg. 284.

[16] Guiandin, Roxana. Pons, Claudia. *Relaciones entre Casos de Uso en el Unified Modeling Language*. REVISTA COLOMBIANA DE COMPUTACIÓN, Volumen 1, Numero 1, pg 73-90.

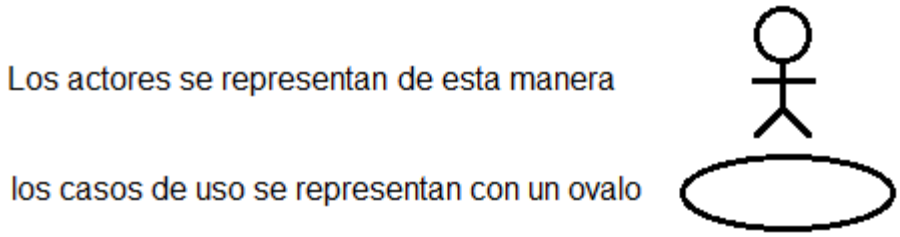

Figura2: Representación Casos de Uso

Fuente: Guiandin, Roxana y Pons, Claudia. Relaciones entre Casos de Uso en el Unified Modeling Language. Revista Colombiana de Computación, Volumen 1, Numero 1.

Un actor y un Caso de Uso se pueden comunicar a través de una asociación en donde cada uno de ellos puede enviar y recibir mensajes. Hay tres tipos de relaciones:

- Generalización (*generalization*): Es una relación que amplía la funcionalidad de un Caso de Uso o refina su funcionalidad original mediante el agregado de nuevas operaciones y/o atributos y/o secuencias de acciones. [17]

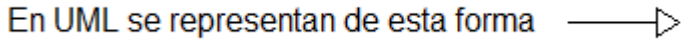

Figura 3: Generalización representación UML

Fuente: Guiandin, Roxana y Pons, Claudia. Relaciones entre Casos de Uso en el Unified Modeling Language. Revista Colombiana de Computación, Volumen 1, Numero 1.

- Extensión (*extend*): Sera utilizado cuando un Caso de Uso sea similar a otro pero con ciertas variaciones.

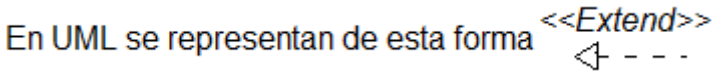

Figura 4: Extensión representación UML

Fuente: Guiandin, Roxana y Pons, Claudia. Relaciones entre Casos de Uso en el Unified Modeling Language. Revista Colombiana de Computación, Volumen 1, Numero 1.

- Inclusión (*include*): Es una relación mediante el cual se re-usa un Caso de Uso encapsulado en distintos contextos a través de su invocación desde otros Casos de Uso.[18]

[17] Guiandin, Roxana. Pons, Claudia. *Relaciones entre Casos de Uso en el Unified Modeling Language*. REVISTA COLOMBIANA DE COMPUTACIÓN, Volumen 1, Numero 1, pg 2.

En UML se representan de esta forma <<*Include*>>

Figura 5: Inclusión representación UML

Fuente: Guiandin, Roxana y Pons, Claudia. Relaciones entre Casos de Uso en el Unified Modeling Language. Revista Colombiana de Computación, Volumen 1, Numero 1.

2.1.23 Atributos de calidad

Estos atributos no están asociados directamente con lo que el software hace, en si refleja su comportamiento durante su ejecución y en la estructura y organización del programa fuente (algunas veces llamados atributos no funcionales), por ejemplo el tiempo de respuesta del software a una pregunta del usuario.[19] Existen pautas y estándares para que los atributos de calidad no se escriban de forma ambigua, las más representativas son la ISO – 9126 y la IEEE1061.

El conjunto específico de atributos que se espera de un sistema depende de la aplicación que se le da a este.

2.1.24 Producto

Consta de un cuerpo código fuente incluido en componentes que puede compilarse y ejecutarse, además de manuales y otros productos asociados. Sin embargo, el producto terminado no sólo debe ajustarse a las necesidades de los usuarios, sino también a las de todos los interesados, es decir, toda la gente que trabajará con el producto. El producto software debería ser algo más que el código máquina que se ejecuta.

El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye el modelo de la arquitectura y el modelo visual permiten a los usuarios utilizar y modificar el sistema de generación en generación.

2.1.25 Propagación del sonido en campo libre

---

[18] Guiandin, Roxana. Pons, Claudia. *Relaciones entre Casos de Uso en el Unified Modeling Language*. REVISTA COLOMBIANA DE COMPUTACIÓN, Volumen 1, Numero 1, pg 2.

[19] Sommerville, Ian. *Ingeniería de Software*. Séptima edición, Pearson Educación S.A, Madrid 2005, pg11.

El sonido en espacio libre se propaga de tal forma que a un oyente ubicado en un punto cualquiera le llega solo la energía directa desde la fuente sin ningún tipo de interferencia. A medida que el receptor se aleja de la fuente, la energía sonora se convierte en calor por lo que el nivel de presión sonora en cada punto va disminuyendo progresivamente. En concreto, la disminución del mismo varía según el frente de onda de la señal.[20]

2.1.26 Nivel Continuo Equivalente

Dado que el sonido no permanece constante a lo largo del tiempo, para valorar lo que ha ocurrido acústicamente a lo largo de un período de medida determinado en un punto del espacio, se usan diferentes índices o niveles de valoración. El nivel sonoro continuo equivalente, es uno de estos, se define como la media energética del nivel de señal promediado en el intervalo de tiempo de medida. Este parámetro no tiene sentido si no va acompañado de una base de tiempo o intervalo de observación.[21]

Este parámetro es básico para cualquier medida de ruido. Su definición se encuentra en la mayoría de las normas de medida de ruido y de la legislación actual sobre protección acústica.

2.1.27 Niveles Percentiles

Se utilizan para tener una idea de las fluctuaciones de sonido a lo largo del tiempo, se suelen usar como índices estadísticos. El nivel percentil n, Ln, es el nivel que se sobrepasa durante el n% del periodo de medida, se suelen usar los percentiles L1, L5, L10, L50, L90, L95 y L99.

Por ejemplo si el tiempo de medida son 10 minutos, y L90 es igual a 80dB y L10 es igual a 90dBA, quiere decir que durante 9 minutos hemos tenido niveles superiores a 80dBA, y que durante 1 minuto hemos tenido niveles superiores a 90dBA.

---

[20] Carrión, Antoni. *Diseño Acústicos de Espacios Arquitectónicos*. Ediciones de la Universidad Politécnica de Catalunya, 1998, pg. 48.

[21] *Qué es el nivel continuo equivalente Leq*. (s.f.). de http://www.inasel.com/Acustipedia/Conceptos-generales/%C2%BFQue-es-el-nivel-continuo-equivalente-Leq.html

2.1.28 CadnaA

El software CadnaA de la firma DATAKUSTIK, es un software de predicción y evaluación de ruido ambiental, que permite la gestión de la inmisión de ruido de acuerdo con normativas internacionales, se pueden modelar todo tipo de emisores acústicos (industria, tráfico rodado, tráfico ferroviario, etc.), siendo una herramienta fundamental para proyectos urbanísticos y estudios de Impacto Ambiental (EIA).

Además, DATAKUSTIK CadnaA permite la realización de mapas estratégicos de ruido en aglomeraciones según las prescripciones de leyes de Ruido Ambiental.

3 METODOLOGÍA

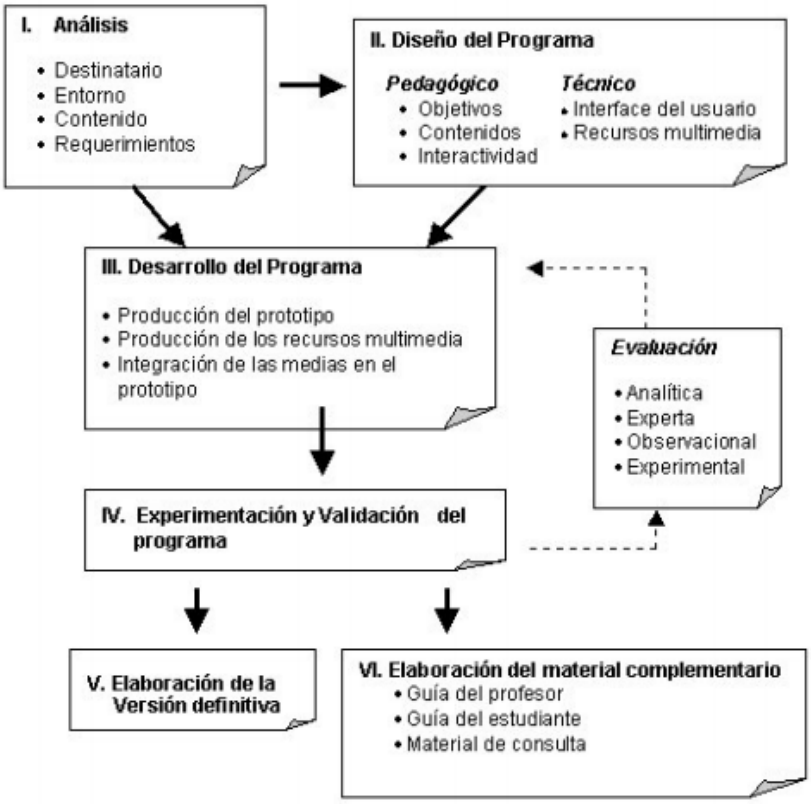

Figura 6: Fases de Desarrollo de Software

Fuente: Consuelo Belloch, Ortí. Aplicaciones multimedia interactivas: Clasificación. Unidad de Tecnología Educativa. Universidad de Valencia.

La metodología de desarrollo de software va de la mano con el desarrollo iterativo e incremental que se plantea para el proyecto la relación parte desde el análisis viéndolo

desde el punto de los modelos de caso de uso y el modelo de análisis valga la redundancia, continuando con el diseño del programa que se plantea en los modelos de diseño, y el desarrollo del programa que se ve plasmado en el modelo de implementación, hasta llegar a la evaluación, experimentación y validación del programa bajo los modelos de prueba especificando atributos de calidad y evaluación del usabilidad para obtener la versión definitiva y el material complementario que se encuentra en el producto final.

3.1 ENFOQUE DE LA INVESTIGACIÓN

La investigación tiene un enfoque **Empírico – Analítico** cuyo interés es el desarrollo ingenieril de una aplicación interactiva enfatizada en un problema de los ingenieros de sonido, el cual es analizar parámetros acústicos en campo libre, mediante la retroalimentación de la teoría con la experimentación interactiva en una aplicación de dispositivo móvil.

3.2 LÍNEA DE INVESTIGACIÓN

La línea de investigación a la cual pertenece es **análisis y procesamiento de señales** porque se están implementando teorías acústicas modernas a la vez del estudio del comportamiento de señales producidas por fuentes sonoras con características tales como su factor de Directividad y potencia eléctrica para poder obtener predicciones de parámetros acústicos en campo libre.

3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN.

Esta fase tiene por finalidad recabar información sobre determinadas características de los usuarios, metodología, hardware, etc. que no formando parte integra del diseño de la aplicación, es necesario tener presente siempre para que la aplicación sea adecuada.

- Características de los usuarios
- Características del entorno de implementación
- Análisis del contenido

- Requerimientos técnicos

Debido a que el proyecto es totalmente teórico en la parte de acústica, la técnica de recolección de información utilizada, son las especificaciones determinadas por otros profesionales en la materia que sean consultados o que están consignadas en textos especializados.

3.4 HIPÓTESIS

Por medio de la aplicación multimedia interactiva se posibilito la opción de predecir y analizar parámetros acústicos introduciendo las variables requeridas por el sistema desde cualquier lugar con tan solo tener un dispositivo móvil a la mano.

3.5 VARIABLES

3.5.1 Variables Independientes

- Tiempo de procesamiento, complejidad computacional.

3.5.2 Variables Dependientes

- La plataforma en la cual sea compilada la aplicación.
- Cantidad de parámetros que se van a predecir en la aplicación.
- Diseño sonoro y entorno grafico del producto final.
- Cantidad de fuentes y puntos de medición que se podrán evaluar, ya que esto depende de la capacidad de memoria y procesador que tenga el dispositivo móvil.

4 DESARROLLO INGENIERIL

El proyecto se realizó bajo la metodología propuesta de desarrollo de software y desplegado mediante el modelo de proceso unificado.

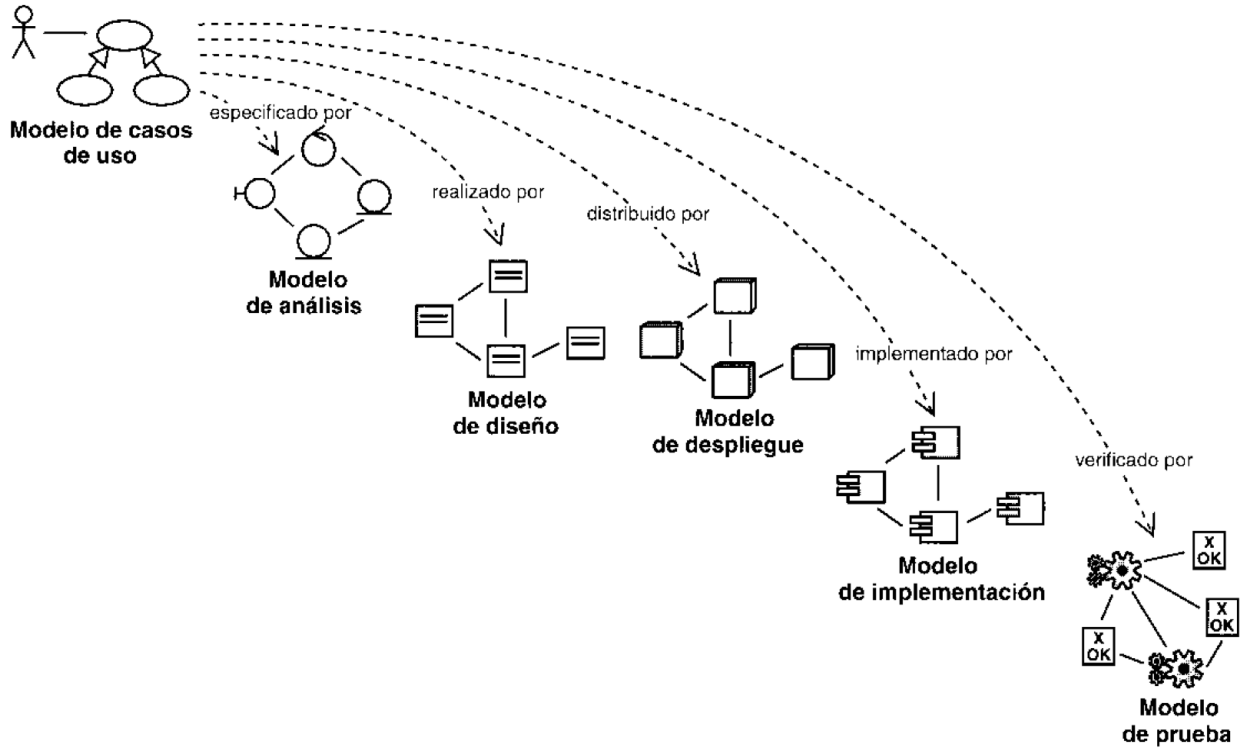

Figura 7: Modelo del Proceso Unificado.

Fuente: Jacobson, Ivar. Booch, Grady y Rumbaugh, James. El Proceso Unificado De Desarrollo De Software, PEARSON EDUCACIÓN Madrid, 2000.

4.1 MODELO DE CASOS DE USO

En el software de predicción de parámetros acústicos solo existe un usuario el cual tiene todos los permisos para manipular la aplicación, la persona que tenga la aplicación se convertirá en el usuario el primer y único actor. El sistema tiene que subministrar cualquier medición que el actor requiera y debe definir las reglas para que el actor trabaje adecuadamente con la aplicación.

Las características de cada caso de uso o la “tabla de casos de uso” se pueden observar a continuación de cada diagrama de Caso de Uso.

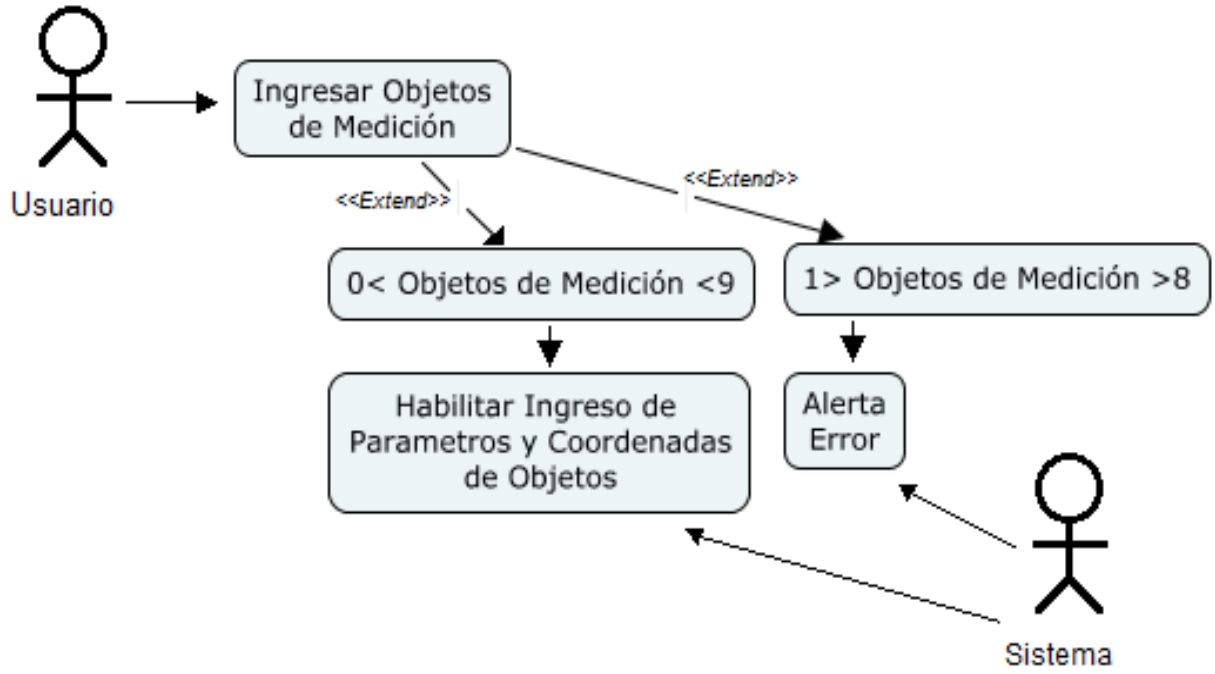

Figura 8: Caso de Uso, Ingresar Objetos de Medición
Fuente: Propia

Tabla 1:
Caso de Uso: Ingresar Objetos de Medición

Caso de Uso: Ingresar Objetos de Medición	
Actor: Usuario	
Curso Normal	Alternativas
El usuario puede registrar correctamente los datos	El usuario no puede registrar datos por digitar un dato que no es permitido
El sistema habilita el Ingreso de Parámetros y coordenadas de Objetos	El sistema envía un alerta y digita el número máximo permitido

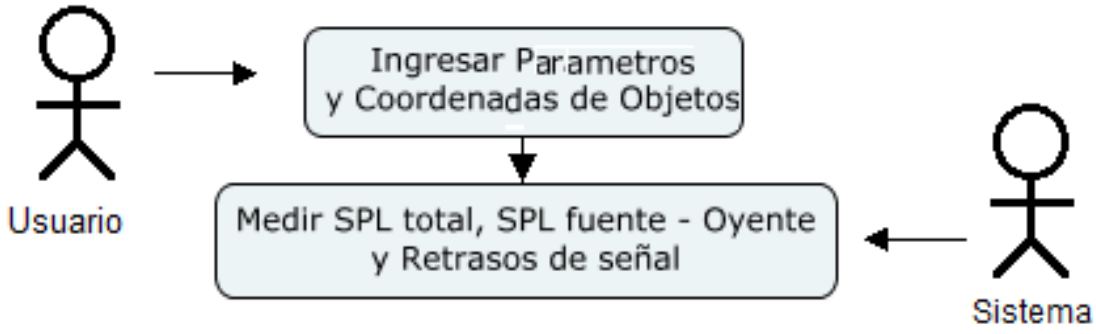

Figura 9: Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos
Fuente: Propia

Tabla 2:

Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos

Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos	
Actor: Usuario	
Curso Normal	Alternativas
El usuario puede registrar correctamente los datos	
El sistema mide y muestra SPL total, SPL fuente - Oyente y Retrasos de señal	

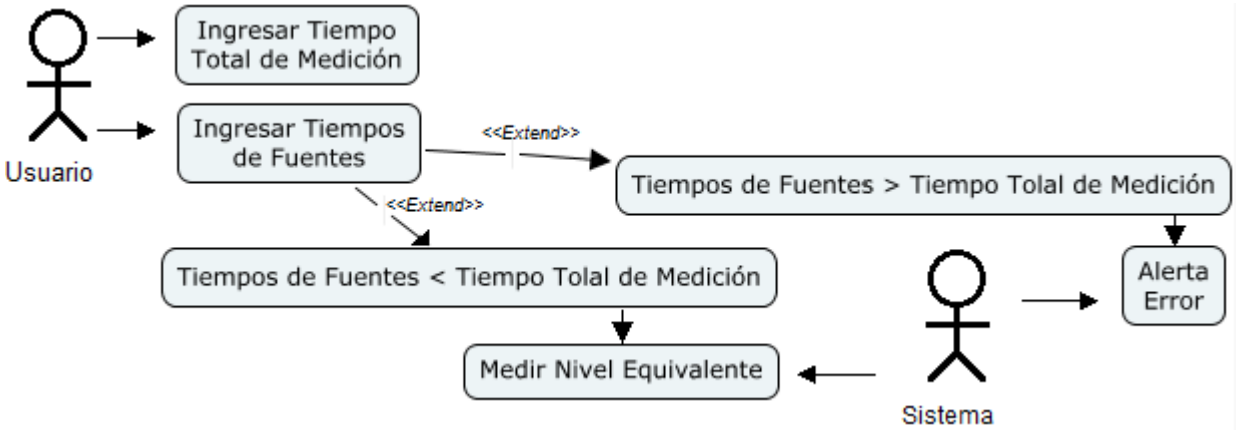

Figura 10: Casos de Uso: Ingresar Tiempos de Medición

Fuente Propia

Tabla 3:

Caso de Uso: Ingresar Tiempo Total de Medición

Caso de Uso: Ingresar Tiempo Total de Medición	
Actor: Usuario	
Curso Normal	Alternativas
El sistema registra el dato	

Tabla 4:

Caso de Uso: Ingresar Tiempo de Fuentes

Caso de Uso: Ingresar Tiempo de Fuentes	
Actor: Usuario	
Curso Normal	Alternativas
El usuario puede registrar correctamente el dato	El usuario no puede registrar datos por digitar un dato que no es permitido
El sistema mide y muestra nivel equivalente	El sistema envía un alerta pidiendo que se digite de nuevo el dato

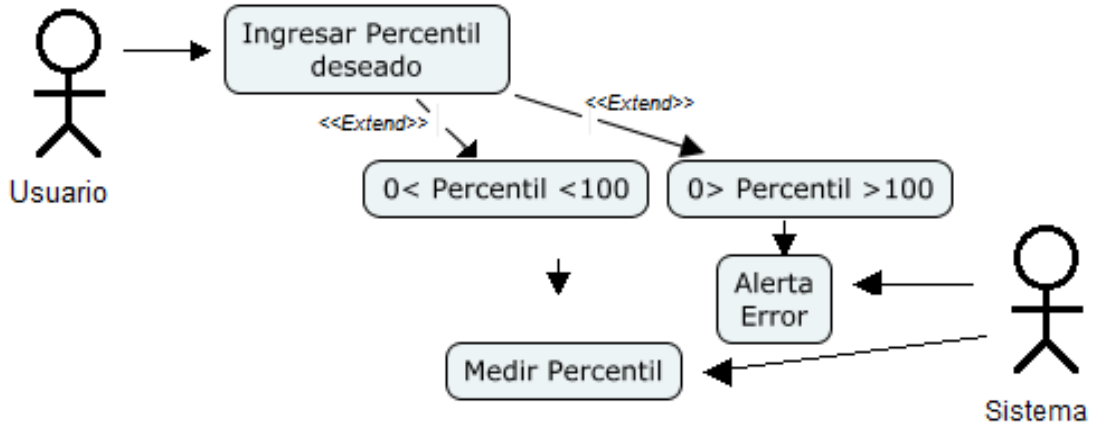

Figura 11: Caso de Uso: Ingresar Percentil deseado
Fuente: Propia

Tabla 5:
Caso de Uso: Ingresar Percentil deseado

Caso de Uso: Ingresar Percentil deseado	
Actor: Usuario	
Curso Normal	Alternativas
El usuario puede registrar correctamente el dato	El usuario no puede registrar datos por digitar un dato que no es permitido
El sistema mide y muestra el percentil deseado	El sistema envía un alerta pidiendo que se digite de nuevo el dato

4.2 ATRIBUTOS DE CALIDAD

- Desempeño
  - Se espera que la aplicación abra en menos de cinco segundos.
  - El tiempo de pasar de una ventana a otra debe ser menor a dos segundos, aun si se pasa por ventanas que muestren mediciones.
  - Se espera que la aplicación no conlleve a usar más del 15.00% de la memoria del dispositivo.
- Confiabilidad
  - Disponibilidad – Se espera que el sistema puede ser usado en cualquier tipo de configuración que tenga el dispositivo.

4.3 MODELO DE ANALISIS

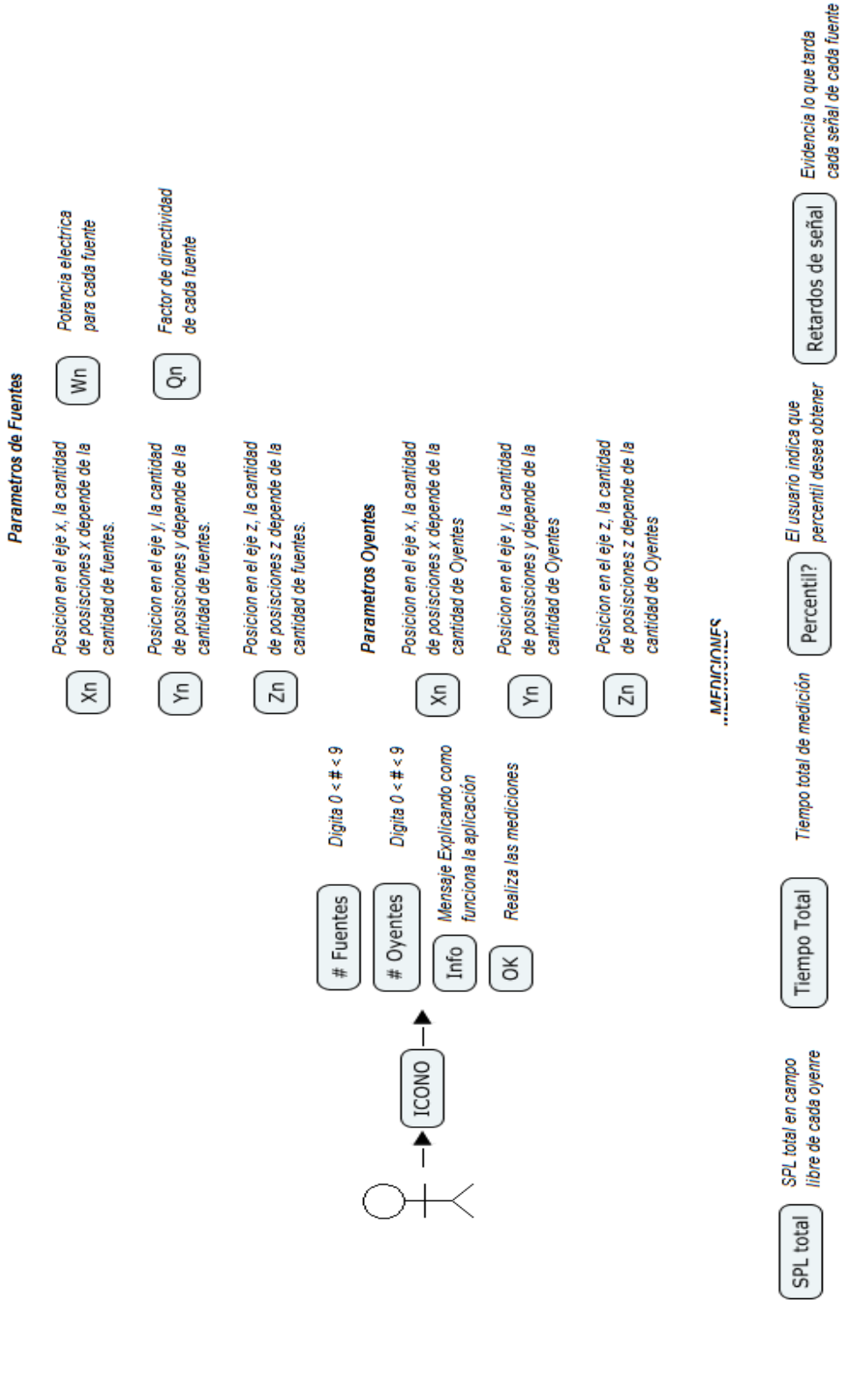

Figura 12: Modelo de Análisis
Fuente: Propia

4.4 MODELO DE DISEÑO

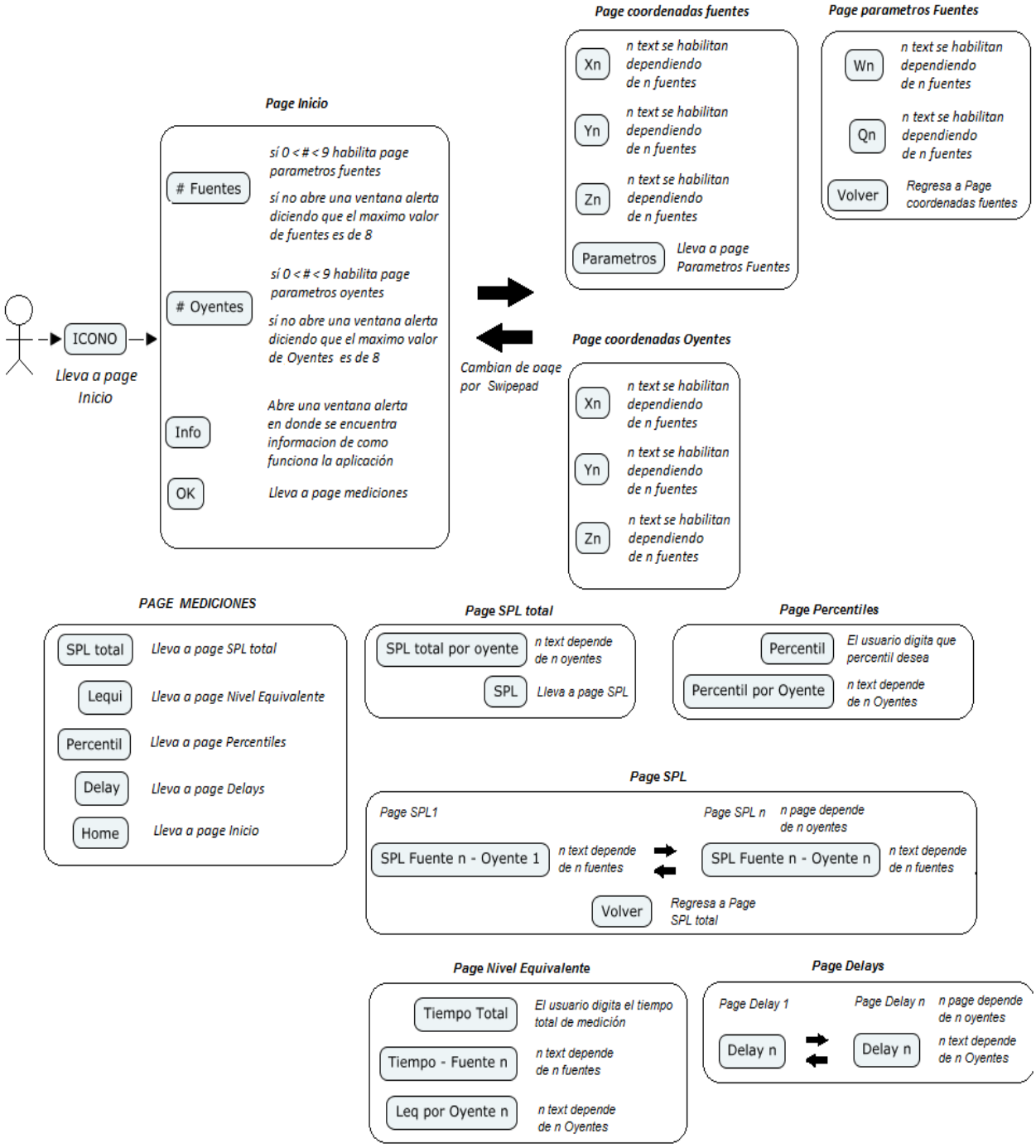

Figura 13: Modelo de Diseño
Fuente: Propia

4.5 MODELO DE IMPLEMENTACIÓN

Con el modelo de análisis y el modelo de diseño se estableció una asignación inicial de funcionalidad del sistema para qué soporte todos los requisitos de la aplicación, con

esto se pudo sacar una conclusión de que tipo de sistema convenía para comenzar la implementación.

Para el desarrollo de la aplicación se trabajara por medio del Android SDK ya que permite realizar cada requisito funcional y el diseño planteado en los modelos ya nombrados, además de ser el compilador base de Android este se encuentra gratuito en el mercado.

Al mismo tiempo de utilizar el Android SDK se trabajara con Phonegap ya que permite realizar el diseño en html5 lo que cede que el desarrollo sea iterativo e incremental ya que así se podrá manejar el mismo diseño en diferentes plataformas.

4.5.1 Diagrama de Componentes

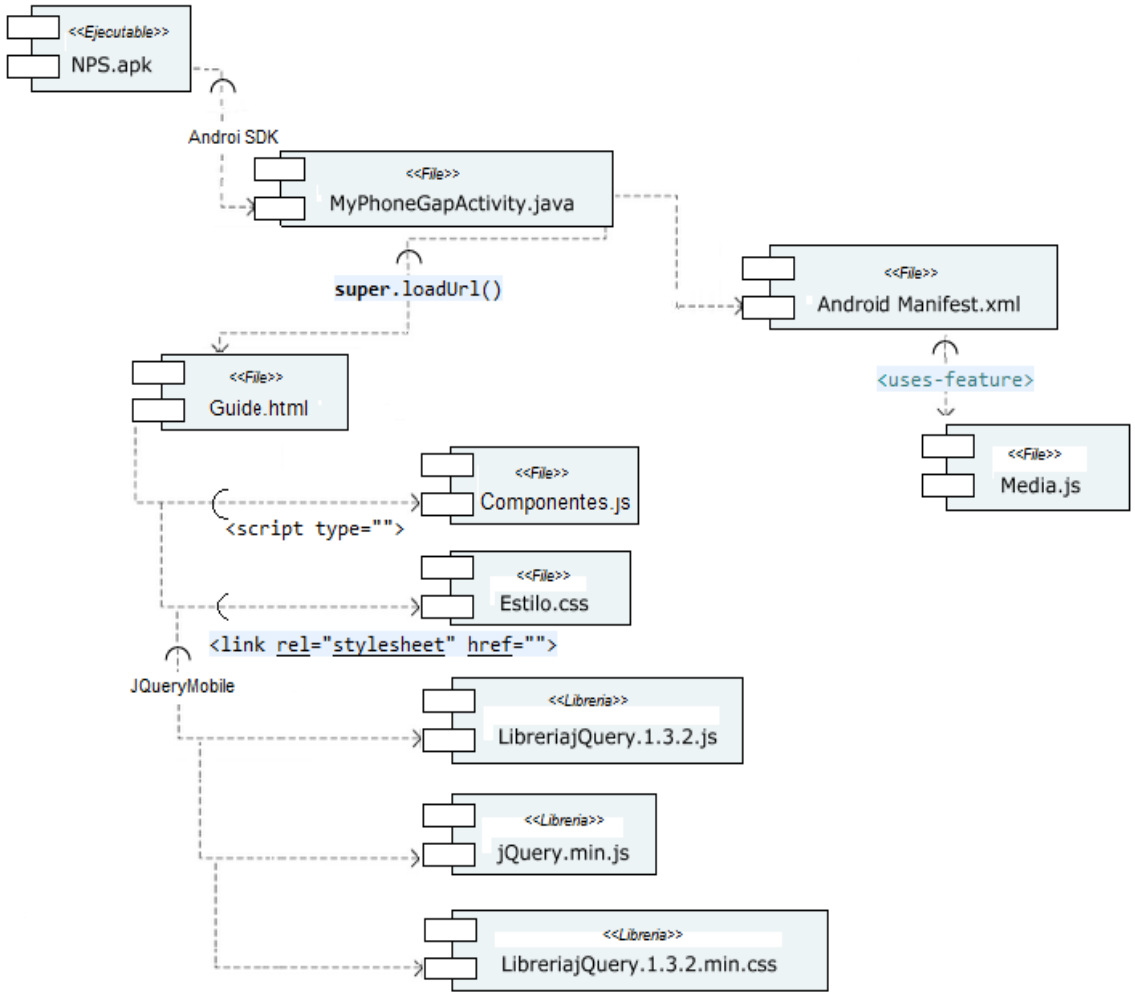

Figura 14: Diagrama de Componentes
Fuente: Propia

El componente.js es la clase en la que se encuentra la programación funcional de la aplicación; los “componentes” dentro de este componente .js se ven referenciados desde la figura 15 hasta la figura 22, se desarrolló de esta forma para no crear una clase por cada componente de estos, si no tenerlos todos en una sola clase.

Numero de Fuentes

```
$("#cafu").on( "input", function( event, ui ) {
var vafu = document.getElementById("cafu").value;
   if (vafu>8){
      alert("En esta aplicacion la mayor cantidad de fuentes a medir son 8");
      document.getElementById("cafu").value =8;
      vafu=8;
   }
   if (vafu<1){
      alert("En esta aplicacion la menor cantidad de fuentes a medir es una");
      document.getElementById("cafu").value =1;
      vafu=1;
   }
})
```

Figura 15: Componente Numero de Fuentes.
Fuente: Propia

Numero de Oyentes

```
$("#caoy").on( "input", function( event, ui ) {
var vaoy = document.getElementById("caoy").value;
   if (vaoy>8){
      alert("En esta aplicacion la mayor cantidad de oyentes en medicion son 8");
      document.getElementById("caoy").value =8;
      vaoy=8;
   }
   if (vaoy<1){
      alert("En esta aplicacion la menor cantidad de oyentes en medicion es de uno");
      document.getElementById("caoy").value =1;
      vaoy=1;
   }
})
```

Figura 16: Componente Numero de Oyentes.
Fuente: Propia

Información

```
<div data-role="popup" id="popupDialog" data-overlay-theme="">
      </div>
            <h3 class="ui-title">texto</h3>
      </div>
</div>

$('#info').on('click', function(e){
    alert("text");
});
```

Figura 17: Componente Información.
Fuente: Propia

swipepad

```
$(document).on('swiperight','#pageinicial', function(){
        $.mobile.changePage("#pagedestino");
    });
$(document).on('swipeleft','#pageinicial', function(){
        $.mobile.changePage("#pagedestino");
});
```

Figura 18: Componente Swipepad

Fuente: Propia

Coordenadas

```
for (var j = 1; j <= vafu; j++) {
var suma= j + 1;
   if (vafu==j){
      for (var i = 1; i < suma; i++) {
         $("#x" + i).css('display', 'block');
         $("#y" + i).css('display', 'block');
         $("#z" + i).css('display', 'block');
         }
      }
      for (var i = suma; i <= 8; i++) {
         $("#x" + i).css('display', 'none');
         $("#y" + i).css('display', 'none');
         $("#z" + i).css('display', 'none');
      }
   }
```

Figura 19: Componente Coordenadas.

Fuente: Propia

Parametros Fuentes

```
for (var j = 1; j <= vafu; j++) {
var suma= j + 1;
   if (vafu==j){
      for (var i = 1; i < suma; i++) {
         $("#q" + i).css('display', 'block');
         $("#w" + i).css('display', 'block');
      }
      for (var i = suma; i <= 8; i++) {
         $("#q" + i).css('display', 'none');
         $("#w" + i).css('display', 'none');
      }
   }
```

Figura 20: Componente Parámetros Fuente.

Fuente: Propia

SPL total

```
$('#total').on("pageshow", function(){
   n=document.getElementById('cafu').value;
   m=document.getElementById('caoy').value;

   for (var i = 1; i <= n; i++) {
      var w =document.getElementById('w' + i).value;
      var q =document.getElementById('q' + i).value;
      var x = document.getElementById('x'+ i).value;
      var y = document.getElementById('y'+ i).value;
      var z = document.getElementById('z'+ i).value;

      var r =[]
      for (var j = 1; j <= m; j++) {
         var xo1 = document.getElementById('xo'+ j).value;
         var yo1 = document.getElementById('yo'+ j).value;
         var zo1 = document.getElementById('zo'+ j).value;

         r[j]=((x-xo1)*(x-xo1))+((y-yo1)*(y-yo1))+((z-zo1)*(z-zo1));
      }

      for (var k = 1; k <= m; k++) {
      document.getElementById('spl' + i + k).value = (120)+(10*(Math.log(w)/Math.LN10))+(10*(Math.log(q/(4*3.1416*(r[k])))/Math.LN10));
      }
   }
   document.getElementById('d11').value =Math.sqrt(r[1]);
});
```

Figura 21: Componente SPL total.

Fuente: Propia

SPL
fuente - Oyente

```
$('#total').on("pageshow", function(){
   n=document.getElementById('cafu').value;
   m=document.getElementById('caoy').value;

   var spl1 =[];var spl2 =[];var spl3 =[];var spl4 =[];var spl5 =[];var spl6 =[];var spl7 =[];var spl8 =[]
   for (var i = 1; i <= n; i++) {
      spl1[i] = Math.pow(10,((document.getElementById('spl' + i + 1).value)/10));
      spl2[i] = Math.pow(10,((document.getElementById('spl' + i + 2).value)/10));
      spl3[i] = Math.pow(10,((document.getElementById('spl' + i + 3).value)/10));
      spl4[i] = Math.pow(10,((document.getElementById('spl' + i + 4).value)/10));
      spl5[i] = Math.pow(10,((document.getElementById('spl' + i + 5).value)/10));
      spl6[i] = Math.pow(10,((document.getElementById('spl' + i + 6).value)/10));
      spl7[i] = Math.pow(10,((document.getElementById('spl' + i + 7).value)/10));
      spl8[i] = Math.pow(10,((document.getElementById('spl' + i + 8).value)/10));

   }

   total1 = eval(spl1.join("+"));
   document.getElementById('st1').value =(10*(Math.log(total1)/Math.LN10));
   total2 = eval(spl2.join("+"));
   document.getElementById('st2').value =(10*(Math.log(total2)/Math.LN10));
   total3 = eval(spl3.join("+"));
   document.getElementById('st3').value =(10*(Math.log(total3)/Math.LN10));
   total4 = eval(spl4.join("+"));
   document.getElementById('st4').value =(10*(Math.log(total4)/Math.LN10));
   total5 = eval(spl5.join("+"));
   document.getElementById('st5').value =(10*(Math.log(total5)/Math.LN10));
   total6 = eval(spl6.join("+"));
   document.getElementById('st6').value =(10*(Math.log(total6)/Math.LN10));
   total7 = eval(spl7.join("+"));
   document.getElementById('st7').value =(10*(Math.log(total7)/Math.LN10));
   total8 = eval(spl8.join("+"));
   document.getElementById('st8').value =(10*(Math.log(total8)/Math.LN10));

});
```

Figura 22: Componente SPL Fuente - Oyente

Fuente: Propia

Tiempos de Medición

```
$('#leqvalue').on("pageshow", function(){

   ttotal=document.getElementById('timetotal').value;

   time1=document.getElementById('time1').value;
   time2=document.getElementById('time2').value;
   time3=document.getElementById('time3').value;
   time4=document.getElementById('time4').value;
   time5=document.getElementById('time5').value;
   time6=document.getElementById('time6').value;
   time7=document.getElementById('time7').value;
   time8=document.getElementById('time8').value;
});
```

Figura 23: Componente Tiempos de Medición

Fuente: Propia

Leq

```
n=document.getElementById('cafu').value;
m=document.getElementById('caoy').value;

var aspl1 =[];var aspl2 =[];var aspl3 =[];var aspl4 =[];var aspl5 =[];var aspl6 =[];var aspl7 =[];var aspl8 =[];
for (var i = 1; i <= n; i++) {
   aspl1[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 1).value)/10))),
   aspl2[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 2).value)/10))),
   aspl3[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 3).value)/10))),
   aspl4[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 4).value)/10))),
   aspl5[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 5).value)/10))),
   aspl6[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 6).value)/10))),
   aspl7[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 7).value)/10))),
   aspl8[i] = document.getElementById('time' + i).value*(Math.pow(10,((document.getElementById('spl' + i + 8).value)/10))),
}

tl1 = eval(aspl1.join("+"));
document.getElementById('leq1').value = (10*(Math.log(tl1/ttotal)/Math.LN10));
tl2 = eval(aspl2.join("+"));
document.getElementById('leq2').value = (10*(Math.log(tl2/ttotal)/Math.LN10));
tl3 = eval(aspl3.join("+"));
document.getElementById('leq3').value = (10*(Math.log(tl3/ttotal)/Math.LN10));
tl4 = eval(aspl4.join("+"));
document.getElementById('leq4').value = (10*(Math.log(tl4/ttotal)/Math.LN10));
tl5 = eval(aspl5.join("+"));
document.getElementById('leq5').value = (10*(Math.log(tl5/ttotal)/Math.LN10));
tl6 = eval(aspl6.join("+"));
document.getElementById('leq6').value = (10*(Math.log(tl6/ttotal)/Math.LN10));
tl7 = eval(aspl7.join("+"));
document.getElementById('leq7').value = (10*(Math.log(tl7/ttotal)/Math.LN10));
tl8 = eval(aspl8.join("+"));
document.getElementById('leq8').value = (10*(Math.log(tl8/ttotal)/Math.LN10));
```

Figura 24: Componente Leq

Fuente: Propia

Alerta Leq

```
$("#time" +i).on( "input", function( event, ui ) {
    var ttotal=document.getElementById('timetotal').value;
    var time1=document.getElementById('time' + i).value;
    if ((ttotal/time + i)<1){
        alert("EL tiempo de la fuente debe ser menor al tiempo total de medicion")
        document.getElementById('time' + i).value =ttotal;
        time + i=ttotal;
    }
});
```

Figura 25: Componente Alerta Leq

Fuente: Propia

Percentil

```
$("#pd1").on( "input", function( event, ui ) {
n=document.getElementById('cafu').value;

ttotal=document.getElementById('timetotal').value;
time1=document.getElementById('time1').value;

tx1=(time1*100)/ttotal;

var pd1=document.getElementById('pd1').value;

p11=document.getElementById('spl11').value;
p12=document.getElementById('spl21').value;
p13=document.getElementById('spl31').value;


if (tx1>pd1){sp1=(Math.pow(10,(p11/10)));}
else {sp1=0}
if (tx2>pd8){sp2=(Math.pow(10,(p12/10))); }
else {sp2=0}
if (tx3>pd8){sp3=(Math.pow(10,(p13/10)));}
else {sp3=0}

if(n==1){
if(n==1 && sp1>0){document.getElementById('p8').value=(10*(Math.log((sp1))/Math.LN10));}
else{document.getElementById('p8').value=0;}
}
if(n==2){
if(n==2 && (sp1+sp2)>0){document.getElementById('p8').value=(10*(Math.log((sp1)+(sp2))/Math.LN10));}
else{document.getElementById('p8').value=0;}
}
if(n==3){
if(n==3 && (sp1+sp2+sp3)>0){document.getElementById('p8').value=(10*(Math.log((sp1)+(sp2)+(sp3))/Math.LN10));}
else{document.getElementById('p8').value=0;}
```

Figura 26: Componente Percentil

Fuente: Propia

Alerta Percentil

```
var pd8=document.getElementById('pd8').value;

if(pd8 > 99){
alert("El valor maximo es 99");
document.getElementById("pd8").value =99;
pd8=99;
}
if(pd8 < 1){
alert("El valor minimo es 1")
document.getElementById("pd8").value =1;
pd8=1;
}
```

Figura 27: Componente Alerta Percentil

Fuente: Propia

Los componentes del guide de la aplicación se encuentran en el archivo Guide.html y se pueden ver en la Figura 23.

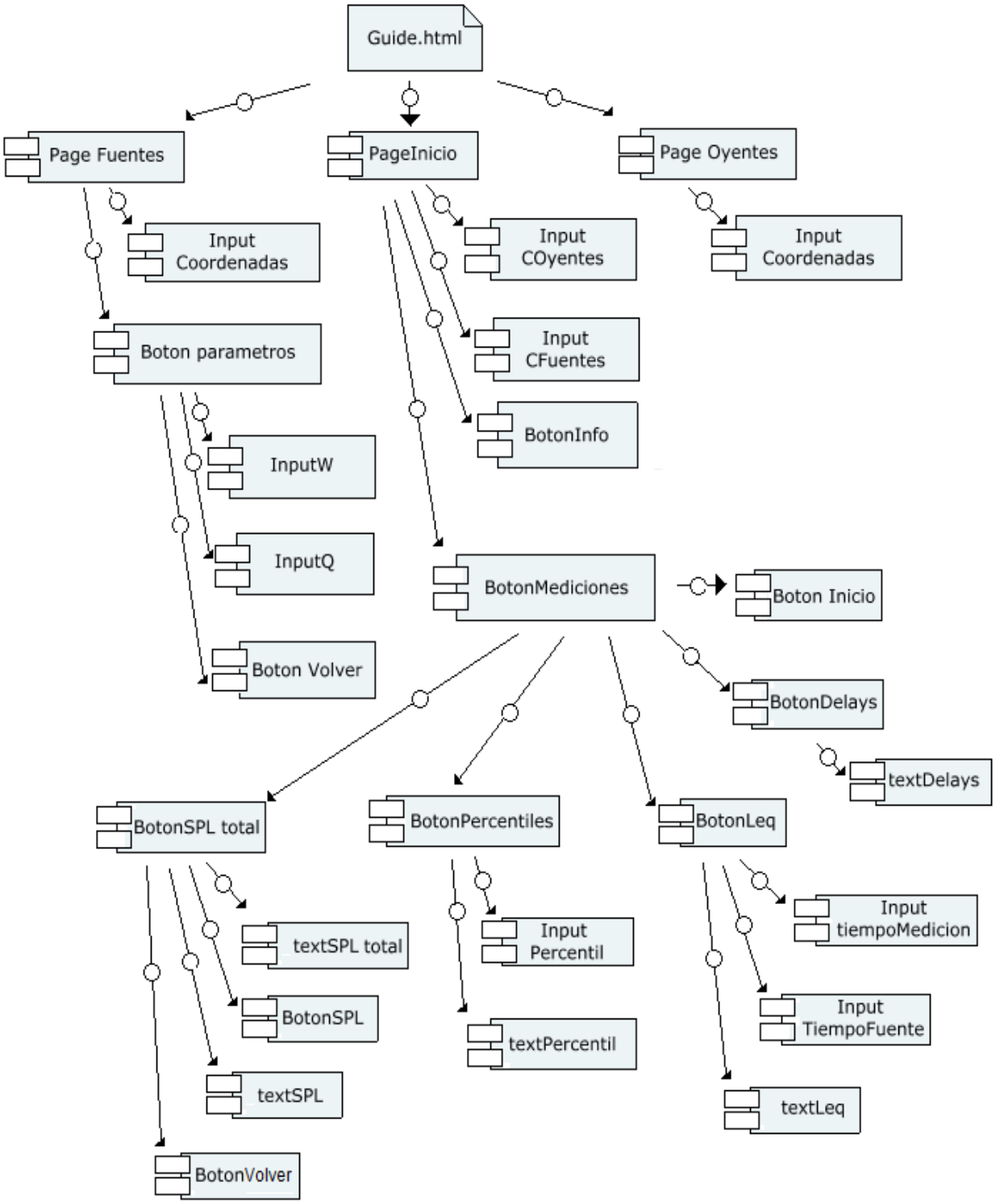

Figura 28: Diagrama de Componentes del Componente Guide.html
Fuente: Propia

4.6 DIAGRAMA DE SECUENCIA

Los diagramas de Secuencia van de acuerdo a los Casos de Uso planteados en el capítulo 4.1. Formulan los pasos en orden secuencial de los métodos y los componentes utilizados para cumplir con los requisitos funcionales.

Los componentes en estos diagramas se referencian a los que están dentro del componente.js

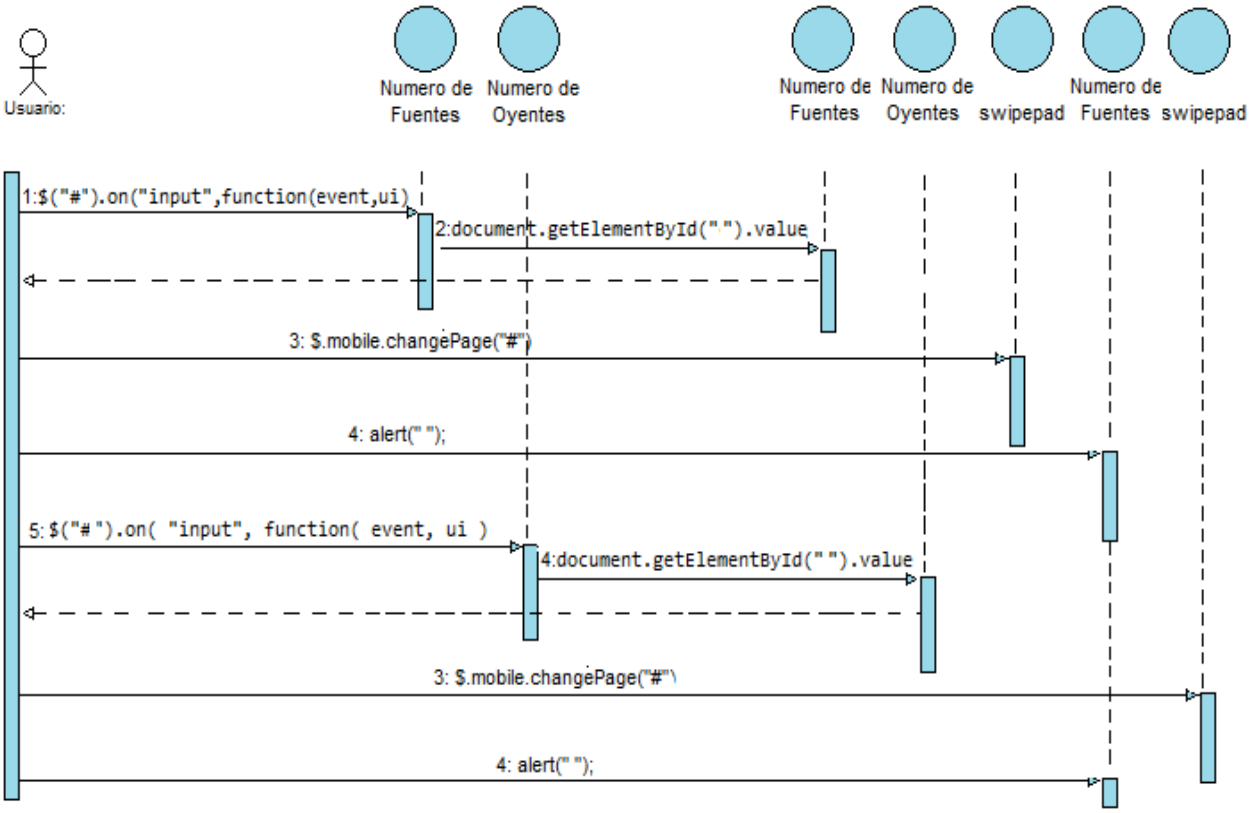

Figura 29: Diagrama de Secuencia - Caso de Uso: Ingresar Objetos de Medición.
Fuente: Propia

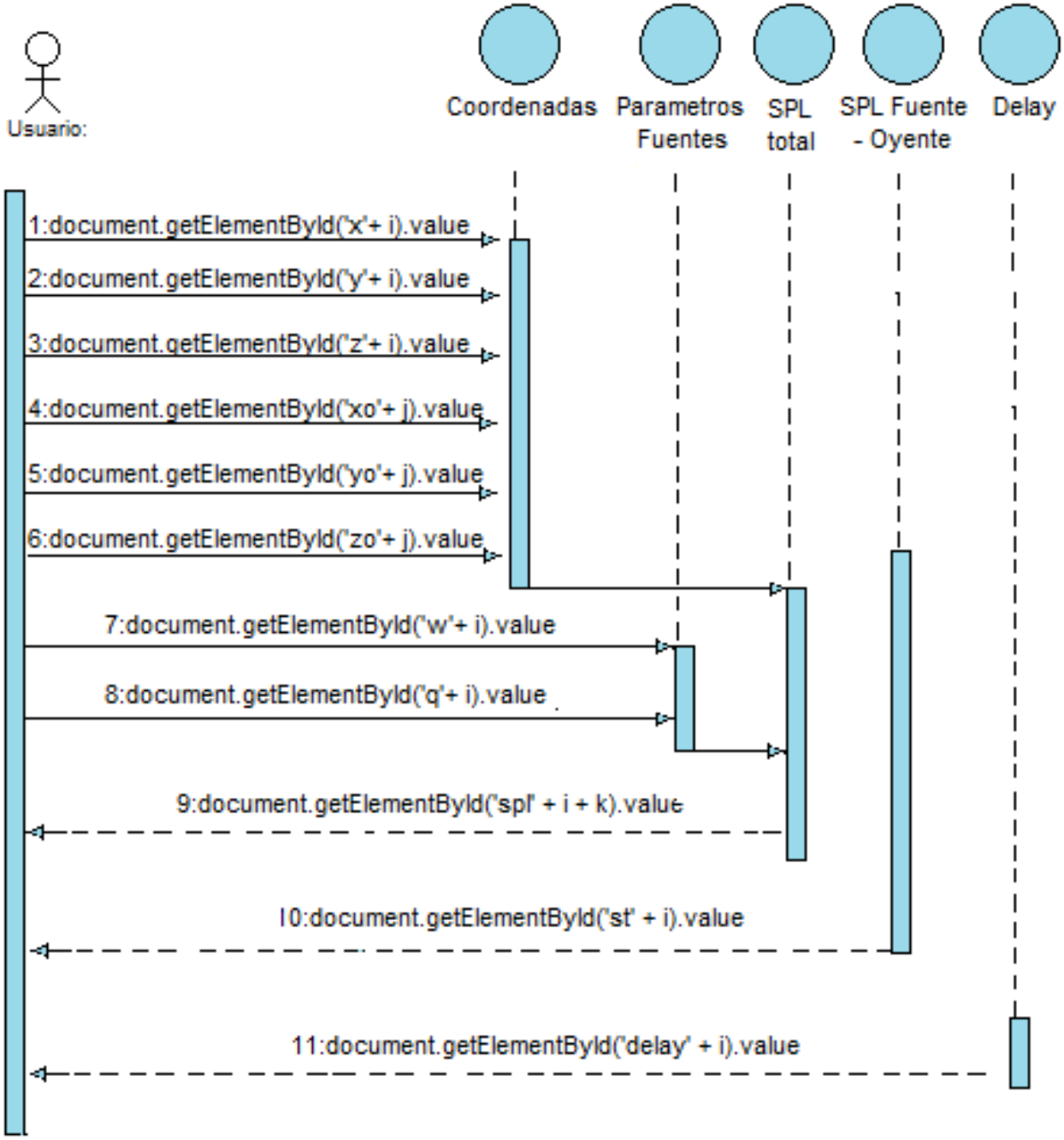

Figura 30: Diagrama de Secuencia - Caso de Uso: Ingresar Parámetros y Coordenadas de Objetos.

Fuente: Propia

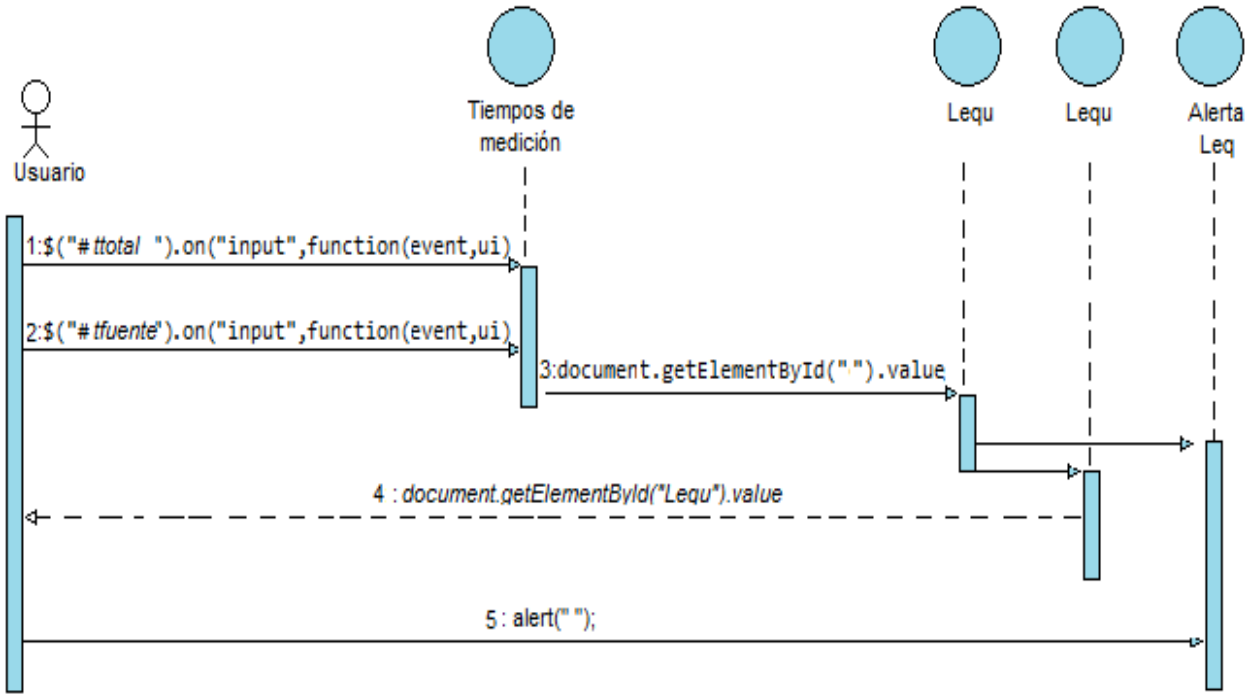

Figura 31: Diagrama de Secuencia - Caso de Uso: Ingresar Tiempo Total de Medición, Caso de Uso: Ingresar Tiempo de Fuentes.
Fuente: Propia

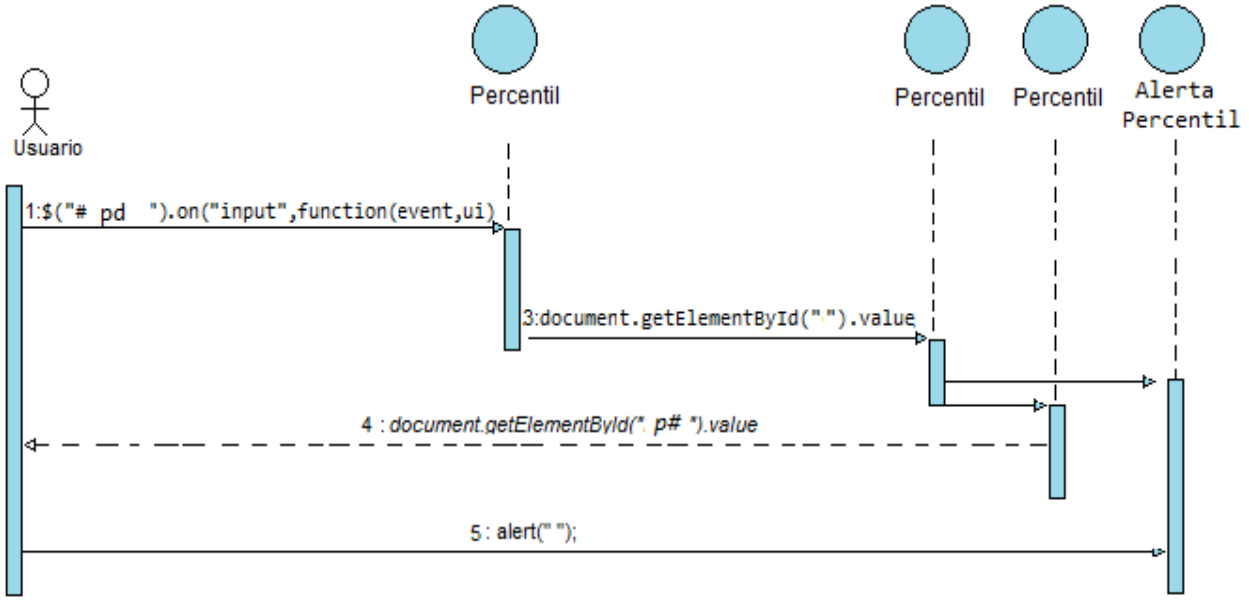

Figura 32: Diagrama de Secuencia - Caso de Uso: Ingresar Percentil deseado
Fuente: Propia

4.7 PRUEBAS DE USABILIDAD

Los casos de prueba fueron diseñados para verificar que el sistema pueda hacer lo que los usuarios quieren que haga es decir que ejecuten los casos de uso, pero también se desarrollan pruebas para poner a prueba el hardware donde la aplicación va a ser implementada.

4.7.1 Pruebas de Hardware

- *Prueba de instalación* Verificar que el sistema puede ser instalado en la plataforma Android y que funcione correctamente cuando sea instalado.

- *Prueba de configuración* Verificar que el sistema funciona correctamente en diferentes configuraciones como lo pueden ser diferentes configuraciones red.

- *Pruebas de tensión o estrés* Provocar que el sistema falle con configuraciones no deseadas, carga de trabajo imposible o competencia por los recursos.

4.7.2 Casos de prueba

Entrada

- Ingresar objetos de medición.
- Ingresar parámetros y coordenadas de objetos de medición.
- Ingresar tiempos de medición.
- Ingresar Percentil deseado

Resultado

- Habilitar ingreso de parámetros y coordenadas de objetos.
- Medir SPL TOTAL, Medir SPL FUENTE – OYENTE.
- Medir Nivel Equivalente
- Medir Percentil

Condiciones

- Tanto el número de fuentes como el número de oyentes en medición no debe ser superior a 8 o inferior a 0.
- Deben ser digitados todos los parámetros que exija el sistema, cada fuente y cada oyente debe tener por completo sus coordenadas x, y, z y las fuentes deben tener sus factores de Directividad y sus potencias eléctricas.
- El tiempo de exposición de cada fuente por separado no debe superar al tiempo total de medición y estos dos parámetros deben estar digitados.
- El percentil ingresado debe ser mayor a 1 y menor a 99.

4.8 VALIDACIÓN CadnaA

Aunque la aplicación desarrollada no tiene la misma funcionalidad que el software de ruido ambiental CadnaA y no presenta características ni variables exactamente iguales y además las dos parten sus mediciones de manera distinta, el software CadnaA permite hacer una comparación de algunas mediciones que realiza la aplicación desarrollada ya que permite medir niveles de presión sonora en campo libre y niveles equivalentes tal y como lo hace la aplicación.

4.8.1 Nivel de Presión Sonora

Mediante las herramientas del software CadnaA, emisor puntual y receptor se realizaron mediciones de nivel de presión sonora con banda única tal y como lo hace la aplicación, primero se efectuó con una única fuente y 8 oyentes en diferentes posiciones.

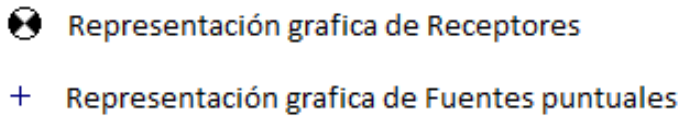

Figura 33: Representaciones graficas de CadnaA
Fuente: Propia

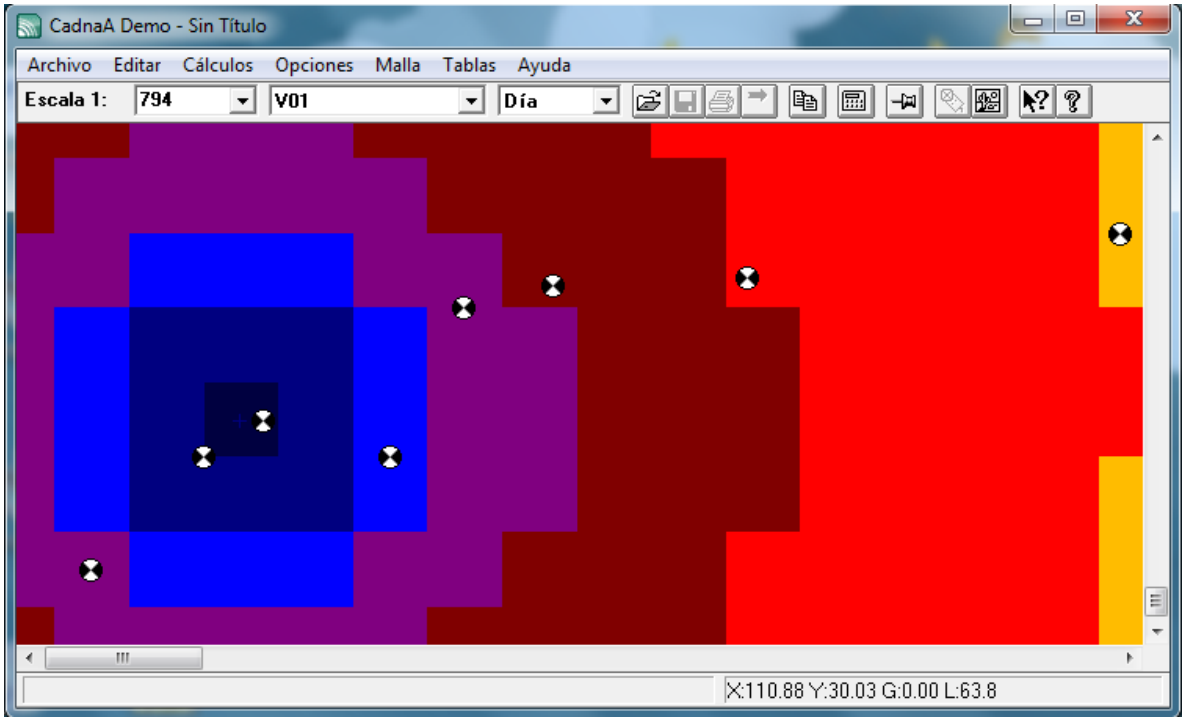

Figura 34: Plano 2D, medición en CadnaA SPL.
Fuente: CadnaA

La fuente fue colocada en las coordenadas cartesianas X: 30 metros, Y: 30 metros y Z: 2 metros con ningún nivel de atenuación o de corrección además de operar durante todo el tiempo con la misma intensidad sonora, ya que en este cálculo no se necesita de tiempos de medición, y los receptores se colocaron sin límites de presión sonora.

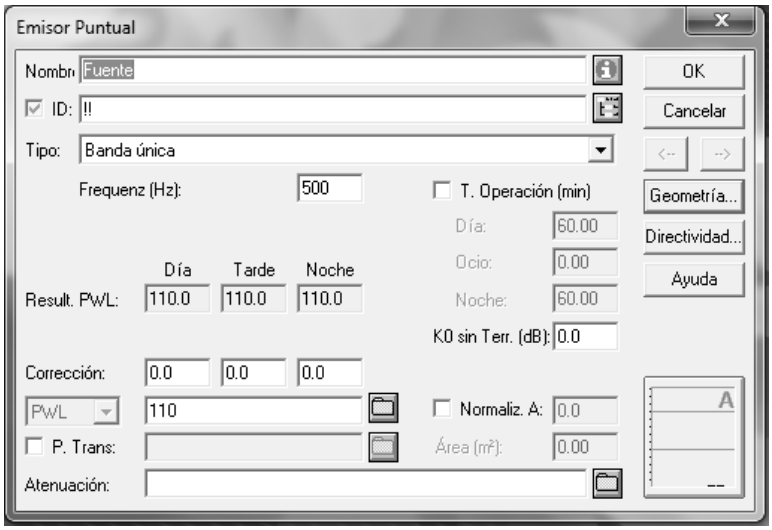

Figura 35: Configuración de la fuente Puntual
Fuente: CadnaA

Figura 36: Coordenadas Fuente puntual
Fuente: CadnaA

Tabla 6:
Coordenadas, receptores medición SPL.

	Coordenadas		
Receptor	x [m]	y [m]	z [m]
1	33	30	2
2	25	25	2
3	50	25	2
4	10	10	2
5	60	45	2
6	72	48	2
7	98	49	2
8	148	55	2

Tabla de Resultados

Cerrar | Editar... | Sinc. Gráfico | Copiar... | Imprimir... | Fuente... | Ayuda

Receptor		Uso de suelo	Valor Límite		Eje. Rel			Lr sin Control de Ruido		dL req.		Lr con Control de Ruido		Exceso		NC pasivo
Nombre	ID		Día	Noche	Estación	Distancia	Altura rel.	Día	Noche	Día	Noche	Día	Noche	Día	Noche	
			dB(A)	dB(A)	m	m	m	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)
4			0	0				64.2	64.2	64.2	64.2	0.0	0.0	-	-	-
5			0	0				62.4	62.4	62.4	62.4	0.0	0.0	-	-	-
2			0	0				77.6	77.6	77.6	77.6	0.0	0.0	-	-	-
1			0	0				85.4	85.4	85.4	85.4	0.0	0.0	-	-	-
3			0	0				67.4	67.4	67.4	67.4	0.0	0.0	-	-	-
6			0	0				59.2	59.2	59.2	59.2	0.0	0.0	-	-	-
7			0	0				54.7	54.7	54.7	54.7	0.0	0.0	-	-	-
8			0	0				49.3	49.3	49.3	49.3	0.0	0.0	-	-	-

Figura 37: Tabla de Resultados CadnaA SPL
Fuente: CadnaA

Se evaluó el mismo caso en la aplicación, una única fuente omnidireccional con las coordenadas descritas en la Figura 32 y ocho receptores con las coordenadas de la tabla 6.

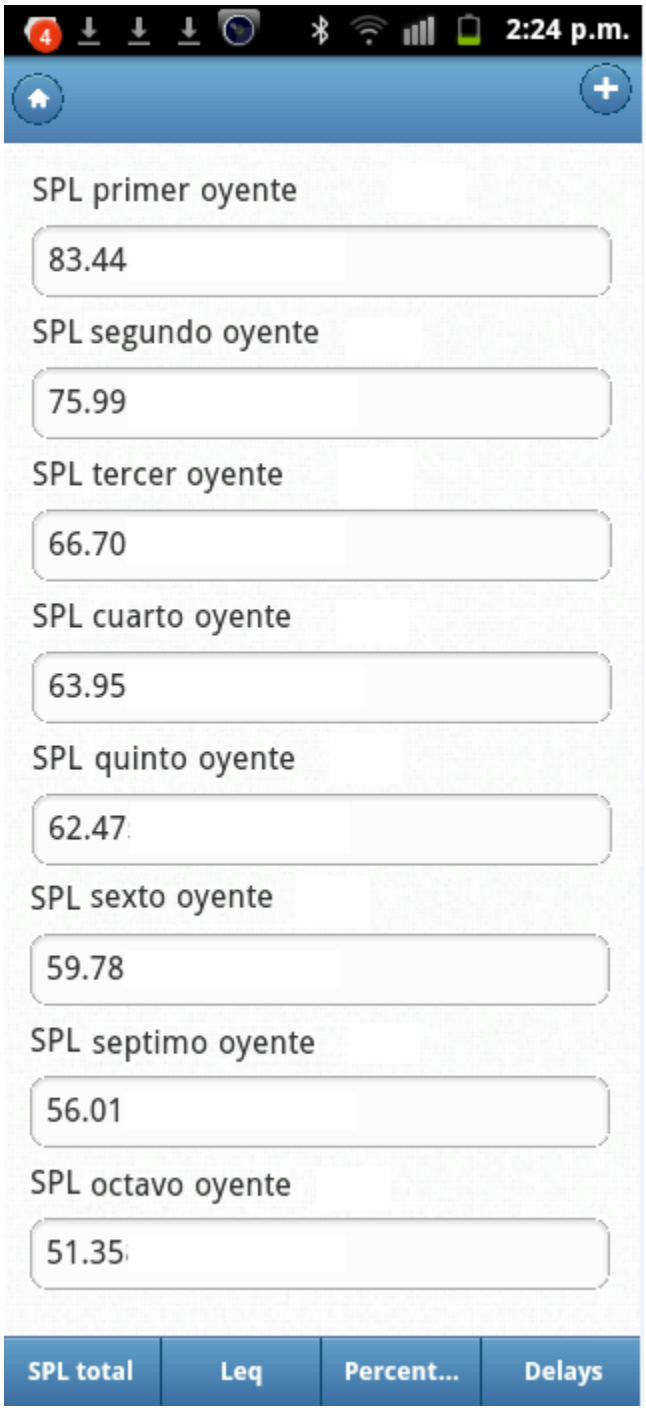

Figura 38: Resultados Aplicación SPL fuente – Oyente
Fuente: Propia

4.8.2 Nivel de Presión Sonora Total

Para medir el nivel de presión sonora total se realizó el cálculo con cuatro fuentes en diferentes coordenadas todas irradiando energía al mismo tiempo y con las mismas características que en la medición de nivel de presión sonora, dejando la primera fuente a utilizar en las mismas coordenadas que en el capítulo 4.8.1.

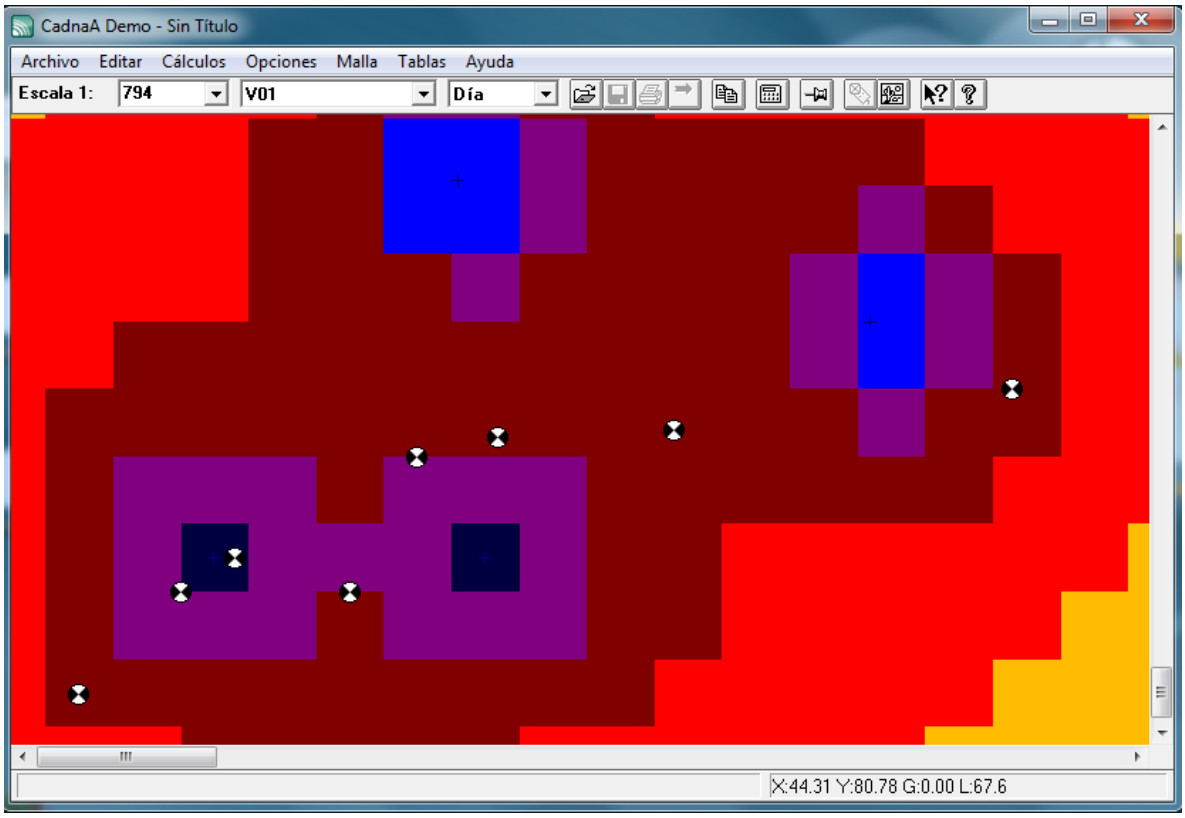

Figura 39: Plano 2D, medición CadnaA SPL total
Fuente: CadnaA

Tabla 7:
Coordenadas de Fuentes Sonoras.

	Coordenadas		
Fuente	x [m]	y [m]	z [m]
1	30	30	2
2	70	30	2
3	127	65	2
4	66	86	2

Tabla de Resultados

Cerrar | Editar... | Sinc. Gráfico | Copiar... | Imprimir... | Fuente... | Ayuda

Receptor		Uso de suelo	Valor Límite		Eje. Rel			Lr sin Control de Ruido		dL req.		Lr con Control de Ruido		Exceso		NC pasivo
Nombre	ID		Día	Noche	Estación	Distancia	Altura rel.	Día	Noche	Día	Noche	Día	Noche	Día	Noche	
			dB(A)	dB(A)	m	m	m	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)
4			0	0				64.0	64.0	64.0	64.0	0.0	0.0	-	-	-
5			0	0				69.1	69.1	69.1	69.1	0.0	0.0	-	-	-
2			0	0				76.6	76.6	76.6	76.6	0.0	0.0	-	-	-
1			0	0				84.3	84.3	84.3	84.3	0.0	0.0	-	-	-
3			0	0				69.5	69.5	69.5	69.5	0.0	0.0	-	-	-
6			0	0				68.9	68.9	68.9	68.9	0.0	0.0	-	-	-
7			0	0				65.5	65.5	65.5	65.5	0.0	0.0	-	-	-
8			0	0				65.5	65.5	65.5	65.5	0.0	0.0	-	-	-

Figura 40: Tabla de Resultados CadnaA SPL total
Fuente: CadnaA

Tabla 8:
Coordenadas, receptores medición SPL total.

	Coordenadas		
Receptor	x [m]	y [m]	z [m]
1	33	30	2
2	25	25	2
3	50	25	2
4	10	10	2
5	60	45	2
6	72	48	2
7	98	49	2
8	148	55	2

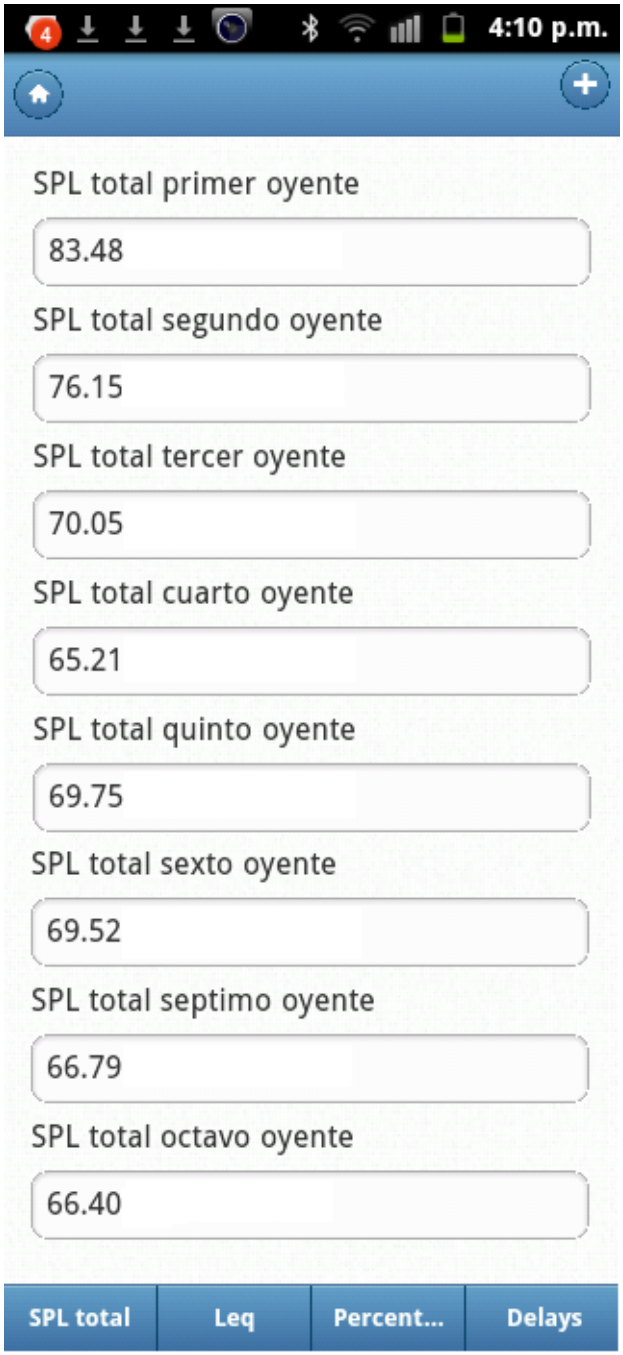

Figura 41: Resultados Aplicación SPL total
Fuente: Propia

4.8.3 Nivel Equivalente

Las mediciones de niveles equivalentes se generaron con cuatro fuentes en diferentes coordenadas y con diferentes tiempos de radiación en el horario del día, y se utilizaron tres receptores distribuidos dentro del espacio de medición.

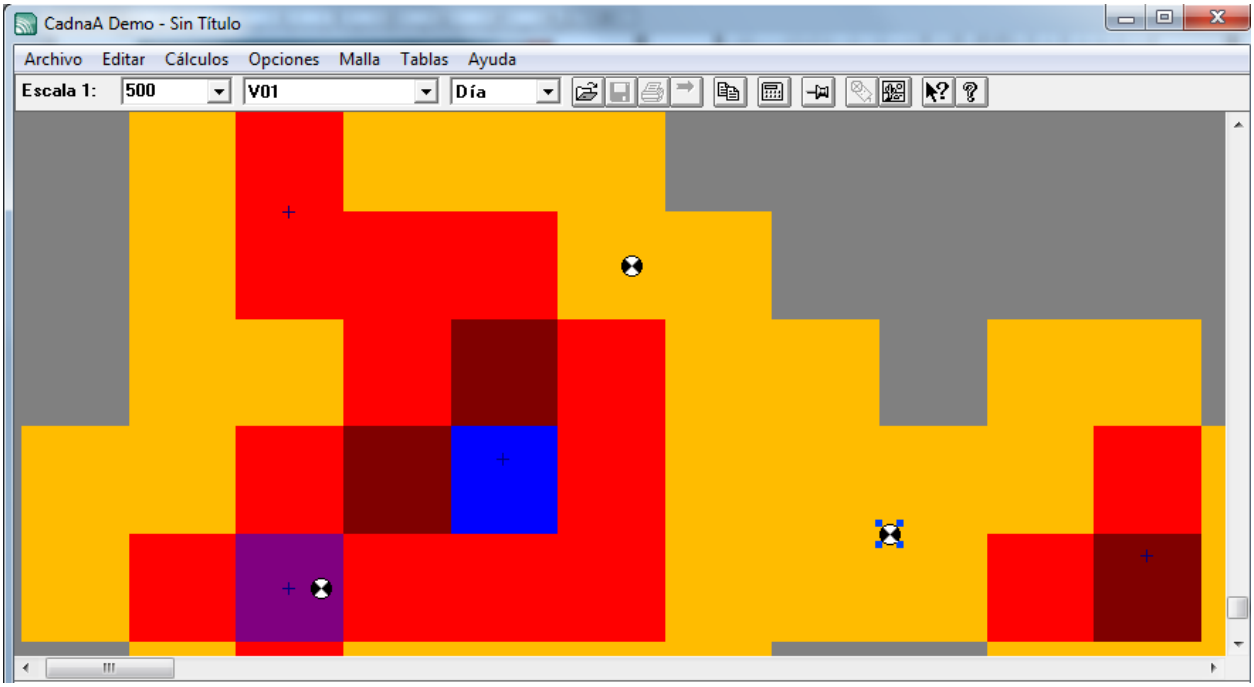

Figura 42: Plano 2D, medición CadnaA Niveles Equivalentes
Fuente: CadnaA

Tabla 9:
Coordenadas de Fuentes, Medición de nivel equivalente

	Coordenadas		
Fuente	x [m]	y [m]	z [m]
1	30	30	2
2	50	42	1,5
3	30	65	2
4	110	33	3

Tabla 10:
Coordenadas de Receptores, Medición de nivel equivalente

	Coordenadas		
Receptor	x [m]	y [m]	z [m]
1	33	30	2
2	62	60	4
3	86	35	1

Tabla 11:

Tiempos de Operación de Fuentes, Medición de Nivel Equivalente

Fuente	Tiempo de Operación [m]
1	15
2	60
3	10
4	20

Tabla de Resultados

Cerrar | Editar... | Sinc. Gráfico | Copiar... | Imprimir... | Fuente... | Ayuda

Receptor		Uso de suelo	Valor Límite		Eje. Rel			Lr sin Control de Ruido		dL req.		Lr con Control de Ruido		Exceso		NC pasivo
Nombre	ID		Día	Noche	Estación	Distancia	Altura rel.	Día	Noche	Día	Noche	Día	Noche	Día	Noche	
			dB(A)	dB(A)	m	m	m	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)	dB(A)
1			0	0				70.3	-88.0	70.3	-	0.0	0.0	-	-	-
2			0	0				58.2	-88.0	58.2	-	0.0	0.0	-	-	-
3			0	0				52.0	-88.0	52.0	-	0.0	0.0	-	-	-

Figura 43: Tabla de Resultados CadnaA Nivel equivalente

Fuente: CadnaA

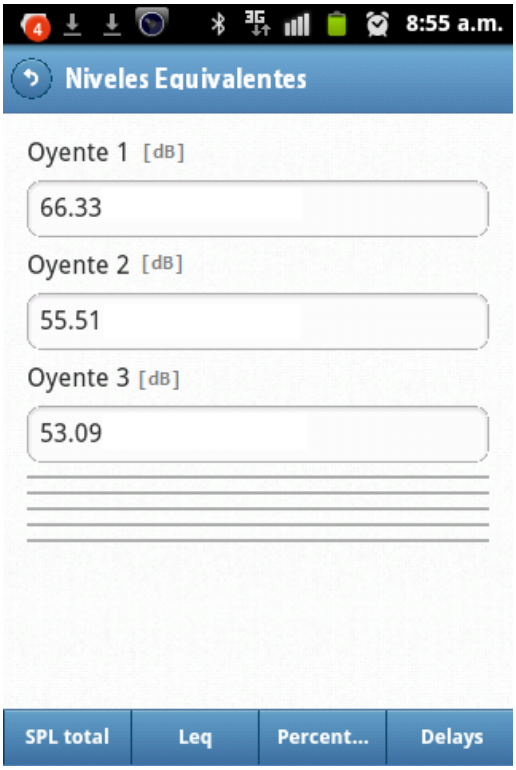

Figura 44: Resultados Aplicación Nivel equivalente

Fuente: Propia

5 ANALISIS DE RESULTADOS

5.1 EVALUACIÓN DE USABILIDAD

Para la evaluación se va a tener en cuenta el planteamiento que se realizó en el capítulo 4.7 pruebas de usabilidad.

5.1.1 Pruebas de Hardware

Para la instalación de la aplicación, se descargó directamente el archivo apk de eclipse a dispositivos móviles con sistema operativo Android, los cuales debían tener al menos 1 kilobyte de espacio disponible en la memoria.

Para terminar la instalación y comprobar que el sistema funciona correctamente el usuario debe permitir los permisos que necesita la aplicación, los cuales son:

- Comunicación por Red: Acceso a Internet sin límite.
- Herramientas del sistema: Enviar difusión permanentemente.

Después de esto como se puede ver en la Ilustración 15 se logró instalar correctamente la aplicación.

Figura 45: Vista de la APP instalada en un SAMSUNG Galaxy ACE
Fuente: Propia

Estos pasos se ejecutaron para instalar la aplicación en 4 tablets y 4 celulares con versiones del sistema operativo diferentes y con diferentes características de hardware.

Tabla 12:
Prueba de Instalación en dispositivos móviles.

Dispositivo	Marca	Versión del sistema Operativo	Instalación
Celular	Sony Xperia ZL	Android v 4.0	OK
Celular	Samsung Galaxy S4	Android v 4.2.2	OK
Celular	LG Nexus 4	Android v 4.2	OK
Celular	Samsung ACE	Android v 2.3.6	OK
Tablet	LENOVO A2107	Android v 4	OK
Tablet	AOC Tablet	Android v 4	OK
Tablet	Samsung Galaxy Tab 2	Android v 4	OK
Tablet	Sony Tablet Xperia S	Android v 4	OK

Se realizaron pruebas de configuración en los dispositivos para ver en que configuraciones la aplicación funcionaba correctamente.

Tabla 13:
Prueba de Configuración.

Configuración	Funcionamiento
Perfil en línea	Completo
Perfil Fuera de línea	Carece en la estructura del diseño
Perfil Silencioso	Completo
Perfil con Sonio	Completo
Bluetooth On	Completo
Bluetooth Off	Completo
GPS On	Completo
GPS Off	Completo

Cuando el Dispositivo se encuentra fuera de línea la estructura de diseño varia ya que para desarrollarla se utilizó una librería JQueri Mobile subida a un servidor, por este motivo cuando el celular esta fuera de línea no la encuentra y produce este error, Sin embrago la aplicación sigue haciendo los cálculos.

Las pruebas de tensión o estrés se realizaron en el celular Samsung ACE el cual tiene un procesador Single Core con Velocidad de 832 MHz, este fue escogido ya que es el dispositivo móvil de prueba de los cuales ya se han referenciado que tiene las características técnicas más bajas en procesamiento y en sistema operativo.

Para estas pruebas se ejerció tensión en el sistema con aplicaciones que se mantuvieran trabajando aun si estas no permanecían abiertas.

Tabla 14:
Prueba de Tensión.

Aplicaciones activas	Uso de CPU
Música	0.00%
Facebook	19.45%
WhatsApp Messenger	7.03%
LINE	6.89%

La aplicación funciono correctamente, no se generó mayor tensión por que la CPU trabaja de acuerdo a las funciones que realizan las aplicaciones así que cada vez que uno abre una aplicación y la aparta para abrir otra esa aplicación deja de gastar procesamiento en la CPU, y las aplicaciones activas en el estudio realizado solo gastan procesador cuando se recibe un mensaje así que la tensión no es continua, lo que concluye en que la aplicación no va a interrumpirse o dejar de trabajar de manera normal a causa de que el celular este trabajando conjuntamente con otras aplicaciones.

5.1.2 Casos de Prueba

Cada caso de prueba según las acciones que haga el usuario dentro de la aplicación se pueden validar o no, y para el software esto significa ir por un lado o por otro hasta conseguir que se cumpla cada caso de uso, a continuación se presenta lo que ocurre en caso de efectuar o de incumplir las condiciones de los casos de prueba.

- Ingresar objetos de medición.

En la validación se puede ver que cumpliendo con las condiciones de este caso se llega al resultado deseado.

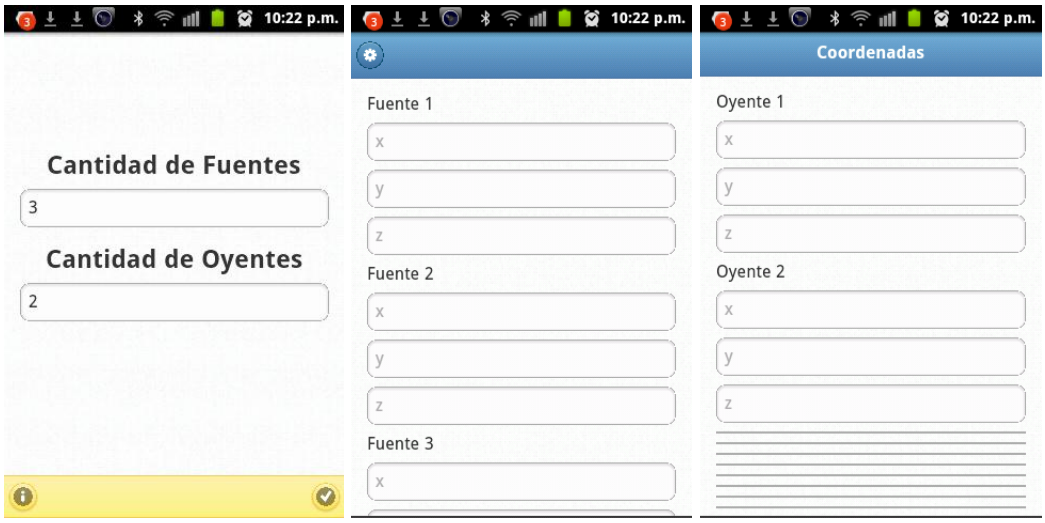

Figura 46: Consecuencias de cumplir con el Caso de Uso: Ingresar Objetos de Medición.
Fuente: Propia

En caso de que no cumpla las condiciones la aplicación desarrolla un sistema que permite enviar al usuario por buen camino para continuar utilizando la aplicación.

Figura 47: Consecuencias de incumplir con el Caso de Uso: Ingresar Objetos de Medición.
Fuente: Propia

- Ingresar parámetros y coordenadas de objetos de medición.

En la validación se puede ver que cumpliendo con las condiciones de este caso se llega al resultado deseado.

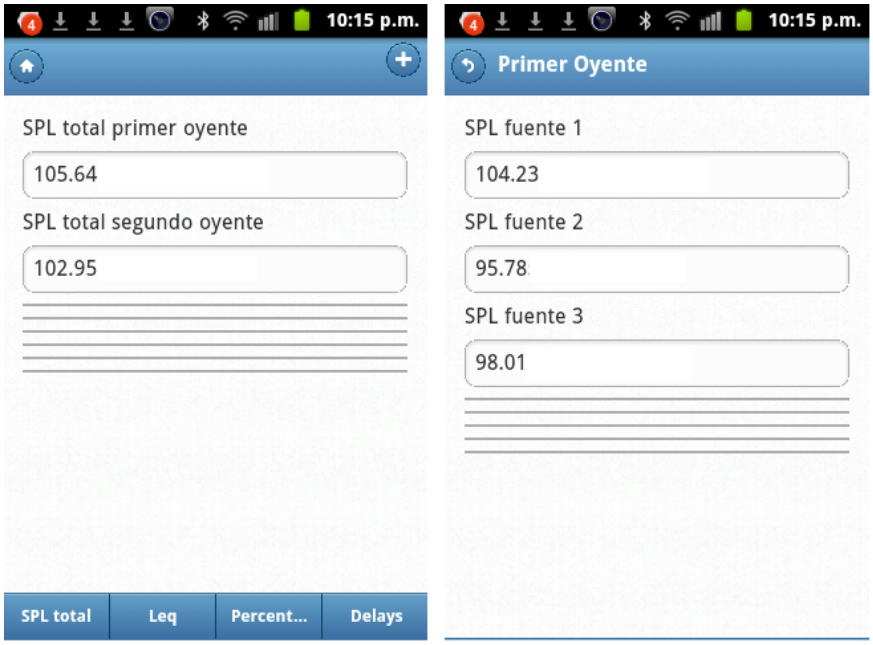

Figura 48: Consecuencias de cumplir con el Caso de Uso: Ingresar parámetros y coordenadas de objetos de medición.

Fuente: Propia

Figura 49: Consecuencias de incumplir con el Caso de Uso: Ingresar parámetros y coordenadas de objetos de medición.

Fuente: Propia

- Ingresar tiempos de medición.

En la validación se puede ver que cumpliendo con las condiciones de este caso se llega al resultado deseado.

Figura 50: Consecuencias de cumplir con los Caso de Uso: Ingresar tiempos de medición.
Fuente: Propia

Figura 51: Consecuencias de incumplir con los Caso de Uso: Ingresar tiempos de medición.
Fuente: Propia

- Ingresar Percentil deseado

5.2 ATRIBUTOS DE CALIDAD

Se evaluara los atributos de calidad según lo planeado en el capítulo 4.2 Atributos de Calidad.

5.2.1 Desempeño

- Tiempo de Apertura de la aplicación por primera vez:: esto se refiere a que la aplicación al abrir por primera vez le toma un tiempo diferente al tiempo de apertura comúnmente, esto pasa por que el sistema empieza a buscar librerías que ya al momento de abrirlas no las vuelve a buscar al momento de abrir la aplicación nuevamente.

Tabla 15:
Tiempos de Apertura por Primera Vez.

Numero de Prueba	Tiempo [s]
1	8,112
2	7,564
3	7,801
4	7,798
5	8,004
6	7,613
7	7,546
8	8,113
9	8,085
10	7,658
11	7,554
12	8,095
13	7,349
14	7,786
15	8,115
16	8,039
17	8,045
18	7,958
19	8,112
20	7,895

La media del tiempo de apertura por primera vez es de 7.8621 mayor a los 5 segundos esperados en los atributos de calidad pero esto sucede en una ocasión especial que es la apertura de la aplicación por vez primera.

- Tiempo de Apertura: Según lo visto en la Tabla 16 se estableció la media del tiempo de apertura de la aplicación el cual resulto siendo de 4.075 segundos, lo que resulta siendo menor a lo esperado lo cual era un tiempo de 5 segundos.

Tabla16:
Tiempos de Apertura.

Numero de Prueba	Tiempo [s]
1	4,113
2	4,008
3	3,839
4	4,159
5	3,856
6	3,992
7	4,232
8	4,375
9	3,888
10	4,416
11	4,08
12	4,145
13	4,008
14	3,944
15	4,112
16	4,008
17	3,936
18	4,039
19	4,464
20	3,887

- Tiempo de cambio de ventanas por medio de Swipepad: Al utilizar Swipepad se llama un algoritmo diferente al que se utiliza para cambiar de ventana por medio de botones, así que primero se evaluó el tiempo de cambio por medio de este método dando a conocer los tiempos como se ven en la Tabla 17, se obtuvo una

media de 1.070 segundos, lo que resulta siendo menor a lo esperado lo cual era un tiempo de 2 segundos.

Tabla 17:
Tiempos de cambio de ventanas por medio de Swipepad.

Numero de Prueba	Tiempo [s]
1	1,032
2	1,448
3	1,408
4	1,072
5	1,008
6	1,008
7	0,944
8	0,856
9	0,988
10	1,216
11	1,012
12	0,945
13	1,162
14	1,055
15	0,998
16	1,009
17	1,059
18	1,245
19	0,896
20	1,048

- Tiempo de cambio de ventanas por medio de Pad: Según los tiempos en la Tabla18 la media de cambio de ventanas por medio de Pad es de 1.4525 segundos, lo que resulta siendo menor a lo esperado lo cual era un tiempo de 2 segundos.

Tabla 18:
Tiempos de cambio de ventanas por medio de Pad.

Numero de Prueba	Tiempo [s]
1	1,52
2	1,552
3	1,673
4	1,152
5	1,272
6	1,808
7	1,64
8	1,455
9	1,339
10	1,496
11	1,335
12	1,458
13	1,326
14	1,315
15	1,423
16	1,408
17	1,4
18	1,456
19	1,367
20	1,655

- Se espera que la aplicación no conlleve a usar más del 15.00% de la memoria del dispositivo.

5.2.2 Confiabilidad

- La aplicación puede estar disponible en cualquier tipo de configuración pero al momento de abrirla por primera vez el perfil debe estar en línea o no se encontrara la librería par la estructura del diseño de la app, así que la aplicación se verá como se muestra en la figura 52.

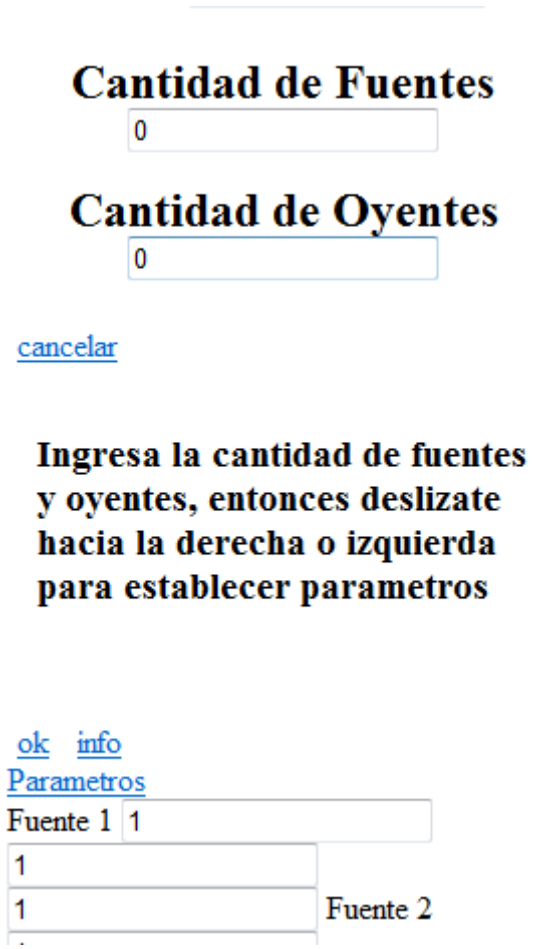

Figura 52: Aplicación abierta por primera vez en perfil fuera de línea
Fuente: Propia

Si se abre la aplicación por primera vez en perfil en línea no abra problemas futuros con la estructura del diseño, así se trabaje en perfil fuera de línea.

5.3 REQUERIMIENTOS DE SISTEMA

Configuración del sistema: Por lo visto en las pruebas de usabilidad el sistema puede trabajar con cualquier tipo de configuración del usuario, pero la versión a su 100% en diseño se vería con perfil en línea.

Memoria RAM: La aplicación viene en un único archivo .apk el cual tiene un tamaño de 1200 bites así que lo justo sería que el dispositivo móvil tuviera 1500 bites de memoria libre disponible.

Sistema Operativo: Ya que la aplicación fue realizada en el SDK de Android el único sistema capaz de soportar esta aplicación es Android.

Versión del sistema operativo: Por motivos de elección se desarrolló la aplicación en el sistema más antiguo que soportaba el SDK ANDROID para que este funcionase desde ahí hasta las versiones más recientes así que se entablo desde la versión 2.3.

Tabla 19:
Requerimientos del Sistema.

Requerimientos del Sistema	
Configuración	Perfil en línea
Memoria RAM	1500 bites
Sistema Operativo	Android
Versión	2.3 - 4.3

5.4 VALIDACIÓN CADNA

Con el desarrollo de los cálculos que se hicieron en CadnaA se pudo observar que este software trabaja a profundidad con las características de las fuentes y de los oyentes en medición, ya que es un software de ruido ambiental las mediciones se realizan bajo normas establecidas sobre ruido, trabajan con ponderaciones, horarios de exposición, y otros factores que no maneja la aplicación desarrollara en el proyecto, Como se ha expresado el software no pretende tener las mismas funcionalidades que CadnaA pero sus mediciones en campo libre permiten analizar algunos resultados que expone la aplicación.

5.4.1 Nivel de Presión Sonora

CadnaA permite ubicar los objetos de medición de forma similar a como lo hace la aplicación del proyecto ya que da un espacio libre en donde se encuentran los ejes x, y, z, donde se pueden colocar las fuentes y receptores en las posiciones y ordenes de antojo. Esto es de gran importancia para medir los niveles sonoros adecuadamente, y compararlos entre sí. Según los datos obtenidos en el capítulo 4.8.2 Nivel de presión sonora donde se realizaron las mediciones de SPL se obtiene la Tabla 20.

Tabla 20:
Niveles de Presión Sonora CadnaA - Aplicación.

Receptor	SPL CadnaA [dB]	SPL Aplicación [dB]
1	85,4	83,4
2	77,6	75,9
3	67,4	66,7
4	64,2	63,9
5	62,4	62,4
6	59,2	59,7
7	54,7	56
8	49,3	51,3

Gracias a los datos de la Tabla 20: Niveles de Presión Sonora CadnaA – Aplicación, se logró evidenciar la media de el margen de diferencia que existe entre las mediciones de estos dos software la cual es de 1.0625 [dB]. Este margen se puede deber a que la forma de realizar las mediciones de cada software es diferente, CadnaA utiliza variables como el PWL, Correcciones, atenuaciones y ponderaciones, mientras que el software del proyecto realiza esta medición bajo la influencia del factor de directividad y la potencia eléctrica de la fuente.

5.4.2 Nivel de Presión Sonora Total

El nivel de Presión Sonora Total se mide de la misma manera en los dos software, mediante suma de niveles.

Tabla 21:
Niveles de Presión Sonora total CadnaA - Aplicación

Receptor	SPL total CadnaA [dB]	SPL total Aplicación [dB]
1	84,3	83,4
2	76,6	76,1
3	69,5	70
4	64	65,2
5	69,1	69,7
6	68,9	69,5
7	65,5	66,7
8	65,5	66,4

Con los datos de la Tabla 21: Niveles de Presión Sonora total CadnaA – Aplicación, se logró evidenciar la media de él margen de diferencia que existe entre CadnaA y la aplicación desarrollada el cual es de 0.8 [dB]. Se puede observar que aunque sean dos software muy distintos con diferentes funcionalidades y diferentes características la media de diferencia de niveles de presión sonora no son altos en si se presenta menos de 1 decibel de diferencia comprobando así que los datos de la aplicación se asemejan bastante a los datos de un software renombrado y profesional como lo es CadnaA.

5.4.3 Nivel Equivalente

La aplicación desarrollada mide niveles equivalentes, teniendo en cuanta el tiempo de radiación de cada fuente, el tiempo de medición total y el nivel de presión sonora que aporta cada fuente a cada oyente, CadnaA sigue reglamentos internacionales de medición de ruido ya que se trata de un software con ese propósito, ruido ambiental, así que los niveles equivalentes los establecen según el horario de medición de día o de noche y en las mediciones intervienen factores como la absorción de suelo o la del aire.

En la medición con la aplicación se escogió un tiempo total de medición de 840 minutos, que es el tiempo total diurno que utiliza CadnaA, y se utilizaron solo 4 fuentes de las 8 disponibles en la aplicación y 3 oyentes de los 8 disponibles de la aplicación, con el fin de comprobar que la aplicación funcionaba aun si no todas las fuentes y oyentes estuvieran activas, con esto los resultados de CadnaA y la aplicación se ven en la tabla 22.

Tabla 22:
Niveles Equivalentes CadnaA - Aplicación

Receptor	Leq CadnaA [dB]	Leq Aplicación [dB]
1	70,3	66,3
2	58,2	55,5
3	52	53

Con los datos de la Tabla 22 Niveles Equivalentes CadnaA – Aplicación, se logró evidenciar la media de él margen de diferencia que existe entre CadnaA y la aplicación

desarrollada la cual es de 2.56 [dB]. Esta es la media de diferencia más grande que ocurrió entre los datos de CadnaA y la aplicación, y se debe a que entre más complejos sean los cálculos que se hagan en CadnaA esté utiliza más variables o condicionales que la aplicación no tiene en cuenta por que el software desarrollado no fue diseñado con la intención de cumplir normas de medición de ruido y no utiliza las variables exactas que utiliza CadnaA.

6 CONCLUSIONES

Repasando los objetivos inicialmente marcados se presentan las conclusiones finales.

La predicción de parámetros acústicos en esta aplicación depende solamente de datos como la ubicación de las fuentes y de los oyentes, de la potencia eléctrica, y el factor de Directividad de las fuentes, y de tiempos de medición. Es una predicción que no evalúa factores externos como factores del campo, no maneja a profundidad características del entorno así se trabaje en campo libre no maneja absorción de piso, de aire… entre otras absorciones como lo hace CadnaA ya que este no es un software diseñado para ruido ambiental, sino que es un software diseñado para obtener valores puramente teóricos brindando rapidez al momento de que se necesite evaluarlos en un sitio en donde no esté la instrumentación necesaria para realizarlos de manera profunda.

Al haber optado por trabajar con eclipse, Phonegap y JQuery Mobile se pudo trabajar por medio de HTML, lo que permitió capacidad en el diseño de la aplicación, y desarrollo iterativo entre HTML, java y JavaScript lo que permitió conseguir los objetivos presentados sin tener que tener conocimientos muy avanzados de programación de objetos.

CadnaA es una herramienta de trabajo profesional que realiza mediciones de ruido ambiental, y por ello su diseño va todo enfocado en cumplir con esa necesidad, al validar el sistema desarrollado se pudo realizar una comparación, mas no se esperaba que los datos fueran iguales si se esperaba que los datos fueran cercanos aunque las variables para llegar a las mediciones fueran distintas, se comprobó que la media de variación en niveles de presión sonora es de 1.062[dB], la media de variación del nivel

de presión sonoro total es de 0.8[dB] y la media de variación del nivel equivalente es de 2.56 [dB], observando que es el de mayor variación debido a que hay más factores que afectan esta medición en CadnaA.

7 RECOMENDACIONES

En comparación con los objetivos marcados inicialmente se presentan las recomendaciones para posibles trabajos futuros.

La aplicación se creó para un número en específico de oyentes y fuentes de medición, este número se puede agrandar según las especificaciones del dispositivo, así que con los futuros avances y mejoras en las capacidades de los dispositivos móviles se puede agrandar la cantidad de elementos de medición.

La aplicación fue creada por medio de jQuery Mobile y Phonegap lo que permite expandir este proyecto a diferentes sistemas operativos por si se quiere hacer funcionar la aplicación en i OS, Symbian o cualquier otro sistema operativo

Al trabajar con jQueryMovile se pueden realizar temas de aplicaciones, se puede trabajar en mejorar el diseño de la aplicación para que se vea mas atrayente hacia el público.

GLOSARIO

ACUSTICA: Ciencia que estudia las cuestiones relativas al sonido, especialmente la generación y recepción de las ondas sonoras.

ALGORITMO: Palabra que viene del nombre del matemático árabe Al-Khwarizmi (780 - 850 aprox.). Define el conjunto de instrucciones que sirven para ejecutar una tarea o resolver un problema.

API: Siglas de Application Programming Interface, en español Interfaz de Programación de Aplicaciones. Consiste en un conjunto de llamadas que ofrecen acceso a funciones y procedimientos, representando una capa de abstracción para el desarrollador.

APP: Un programa autónomo o una pieza de software diseñado para cumplir un propósito en particular, la aplicación, especialmente va para usuarios con dispositivos móvil.

ASCII: Siglas en inglés para American Standard Code for Information Interchange, es decir Código Americano Estándar para el intercambio de Información, es un estándar para el código utilizado por computadoras para representar todas las letras mayúsculas, minúsculas, letras latinas, números, signos de puntuación, etc.

COMPILADOR: programa que transforma el lenguaje de alto nivel al lenguaje de máquina.

COMPLEJIDAD COMPUTACIONAL: estudia la eficiencia de los algoritmos estableciendo su efectividad de acuerdo al tiempo de corrida y al espacio requerido en la computadora o almacenamiento de datos, ayudando a evaluar la viabilidad de la implementación práctica en tiempo y costo.

DECIBELIO: Escala convenida habitualmente para medir la magnitud del sonido. El número de decibelios de un sonido equivale a 10 veces el valor del logaritmo decimal de la relación entre la energía asociada al sonido y una energía que se toma como referencia.

FACTOR DE DIRECTIVIDAD: es una expresión matemática de la Directividad que resulta de comparar el nivel en el eje con el nivel medio de todos los puntos de la esfera de medida.

FRAMEWORK: Término con el que se define un amplio conjunto de elementos que permite desarrollar y organizar software utilizando un determinado lenguaje, sistema o tecnología. Habitualmente incluye bibliotecas, programas de desarrollo o manuales.

FRENTE DE ONDA: conjunto de puntos del medio alcanzados por el movimiento ondulatorio en el mismo instante; o dicho con mayor precisión, el lugar geométrico de todos los puntos del medio con igual fase de vibración.

GUIDE: Es una manera de nombrar a la interfaz gráfica de cualquier tipo de software.

HARDWARE: Maquinaria. Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

INTERFAZ: En computación, una interfaz se refiere a una abstracción que un determinado elemento ofrece de sí mismo al exterior, facilitando de esta forma su acceso y uso por otros elementos de hardware o software.

ITERATIVO: Se aplica a la palabra que denota una acción que se repite

OPEN SOURCE: Código fuente abierto software libre, se refiere a un programa cuyo código fuente está disponible al público general, gratis, para usar y modificar.

PROCESADOR: es el cerebro del PC. Se encarga de hacer funcionar a las aplicaciones y el sistema operativo dando respuesta a las órdenes que le envías a través de los periféricos de entrada como el teclado o el ratón.

RAM: Siglas de Random Access Memory, o en español Memoria de Acceso Aleatorio. Componente de memoria volátil, con palabras de acceso individual mediante una dirección de memoria específica y con rápida ejecución de operaciones de lectura y escritura

SCRIPT: El término *script* hace referencia a todos aquellos ficheros o secciones de código escritas en algún lenguaje de programación**,** como Visual Basic Script (VBScript), JavaScript, etc.

SISTEMA OPERATIVO: Programa cuya finalidad principal es simplificar el manejo y explotación de un elemento con capacidad computacional, gestionando sus recursos, ofreciendo servicios a las demás aplicaciones y ejecutando mandatos del usuario.

SOFTWARE: Se refiere a programas en general, aplicaciones, juegos, sistemas operativos, utilitarios, antivirus, etc. Lo que se pueda ejecutar en la computadora.

SPL: Nivel de Presión Sonora (Sound Preassure Level SPL), se refiere a la desviación en la presión del aire respecto a la presión atmosférica causada por una onda sonora.

SWIPEPAD: Es una aplicación que permite deslizarse de una lugar a otro en las pantallas de dispositivos electrónicos presionando la pantalla y haciendo un corrimiento con el dedo al lugar que se quiera llegar.

WAP: siglas de Wireless Application Protocol, en español o Protocolo de Aplicaciones Inalámbricas. Es un estándar para aplicaciones que utilizan las comunicaciones inalámbricas, como el acceso a servicios de Internet desde un teléfono móvil.

BIBLIOGRAFÍA

- Alonso Baz, Arturo; Álvarez Rodríguez, María y Garcia Baniello, Rosana. *Dispositivos Móvile.* Universidad de Oviedo, Ingeniería de Telecomunicaciones

- Basterretche, Juan. *Dispositivos Móviles.* Trabajo de Adscripción, Universidad Nacional del Nordeste, Argentina 2007.

- Beauperthuy Taibo, José Luis y Malave Polanco, Kristel. *"Android" el sistema operativo de Google para dispositivos móviles*. Fundación Miguel Unamuno y Jugo Venezuela, Vol. 7, núm. 19, 2011, Pg. 79-96.

- Belloch Ortí, Consuelo*. Desarrollo de aplicaciones multimedia interactivas*. Universidad de Valencia: Unidad de Tecnología Educativa (UTE), 2005.

- Carrión, Antoni. *Diseño Acústicos de Espacios Arquitectónicos*. Edición de la Universidad Politécnica de Catalunya, 1998

- Chaparro Escobar, Juan Pablo. *Diseño de 5 módulos interactivos para enseñar principios básicos de acústica.* Universidad Austral de Chile: Ingeniería acústica, 2003.

- De Barrio Mora, Julio Ignacio. *Sistemas de predicción de parámetros que caracterizan el campo sonoro en iglesias.* Universidad Politécnica de Madrid, 2008.

- Gómez Martín, Marco Antonio. *Arquitectura y metodología para el desarrollo de sistemas educativos basados en videojuegos.* Universidad Complutense de Madrid: Ingeniería de Software e Inteligencia Artificial, 2003.

- Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Proceso Unificado De Desarrollo De Software*. PEARSON EDUCACIÓN Madrid, 2000.

- Mares, Laura. *Tablets en educación oportunidades y desafíos en políticas uno a uno*. red Latinoamérica portales educativos, 2012

- Martínez González, Felipe Luis. *Aplicaciones para dispositivos móviles*. Universidad Politécnica de Valencia, 2010 – 2011.

- Padilla Zea, Natalia. *Metodología para el diseño de videojuegos educativos sobre una arquitectura para el análisis del aprendizaje colaborativo.* Universidad de Granada: Ingeniería de lenguajes y sistemas informáticos, 2011.

- Robles Muñoz, Antonio. *Simulación del campo sonoro producido por el tráfico rodado en vías urbanas mediante el método de imágenes acústicas.* Universidad Oviedo, Ingeniería Informática, 2008.

- Roxana S. Giandin y Claudia F. Pons. Relaciones entre *Casos de Uso en el Unified Modeling Language.* Universidad Nacional de La Plata, Volumen 1, número 1

- Sommerville, Ian. *Ingeniería de Software*. Séptima edición, Pearson Educación S.A, Madrid 2005.

- Pérez, José; Fernández Mireya y Ramos Juan. *Análisis de Características Técnicas en Dispositivos Móviles Aplicado a la Instrumentación.* Cuarto congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica, 2010.