

UNIVERSIDAD
PANAMERICANA

Campus Bonaterra

ESCUELA DE INGENIERÍA

TESIS

**Prototipo de un reproductor MP3 con interfaz de audio para
dispositivos Android**

PRESENTADO POR

LUIS FELIPE CASUSO MALDONADO

PARA OBTENER EL GRADO DE

MAESTRÍA EN INGENIERÍA

Dirección de Tesis:

M.I. Juan Antonio de Alba Alcocer

Aguascalientes, Ags. 2015

Agradecimientos

Agradezco a mis padres por su apoyo y sacrificio por lo cual pude concluir mis estudios en esta universidad y alcanzar el grado de maestría. Así como sus frecuentes y valiosos consejos que me acompañaran toda la vida.

Agradezco a los catedráticos de la Universidad Panamericana la oportunidad que me brindaron de aprender y mejorar mis competencias sobre las distintas áreas impartidas en mi carrera de ingeniería.

Agradezco también a los maestros y estudiantes del CAM 6 de Aguascalientes por su participación en la encuesta de esta tesis, y por su excelente retroalimentación.

A mis compañeros, ahora ingenieros, que me apoyaron con algunos temas para su comprensión.

A la M.C. Claudia Nallely Sánchez Gómez por brindar su apoyo para la mejora de esta tesis.

A mi Director de Tesis, el M.I. Juan Antonio de Alba Alcocer, por su apoyo, guía, y paciencia durante todo este tiempo en su proceso de elaboración a todos mil gracias.

Resumen

En esta investigación se presenta el desarrollo de una aplicación para un dispositivo Android, cuya finalidad es ser empleada como interfaz celular o móvil por personas invidentes o de visión reducida, como alternativa a los productos existentes.

La aplicación consistirá principalmente en un menú que funcionará con instrucciones por audio.

Se diseñará la interfaz de la aplicación de forma sencilla e intuitiva. En este proyecto se aprovechará la tecnología táctil de estos dispositivos para crear una aplicación poco convencional que no requerirá del uso de la vista para su manejo.

Se tomarán en cuenta aspectos de diseño y funcionales, por ejemplo cómo reaccionarán los botones en el momento previo a una transición de un menú a otro.

Contenido

1. Introducción	7
1.1 Motivación	8
2. Marco Teórico	10
2.1 Introducción	10
2.2 Tecnología existente	11
2.2.1 Aplicaciones Existentes	11
2.2.2 Productos Electrónicos	16
2.3 Sistema Operativo del Dispositivo	18
2.4 Lenguaje y herramientas de desarrollo	19
2.4.1 Componentes de una aplicación Android	19
2.4.2 Recursos de una aplicación Android	20
2.4.3 Eclipse con ADT	21
3. Diseño de la Aplicación	23
3.1 Problemática	23
3.2 Aplicación propuesta	24
3.3 Análisis y requerimientos de la aplicación	25
3.3.1 Requerimientos funcionales	25
3.3.2 Requerimientos no funcionales	27
3.3.3 Requerimientos del usuario	28
3.4 Características del diseño de la aplicación	30
3.4.1 Funcionamiento de la Interfaz	30
3.4.2 Acciones del layout	32
3.5 Versiones	35
3.5.1 Versión 1.0	35
3.6 Modelo de Menús	36
4. Desarrollo de la Aplicación	38
4.1 Lógica y desarrollo del programa	38

4.1.1 Funcionamiento de las distintas pantallas	41
4.2 Experimentos y Resultados	48
4.2.1 Resultados de la Encuesta Realizada.....	49
5. Conclusiones y trabajo futuro	54
5.1 Conclusiones.....	55
5.2 Costos del proyecto.....	57
5.3 Trabajo Futuro.....	58
6. Referencias.....	62

Biblioteca UP Bonaterra

Índice de Figuras

2.1 Aplicación NoLED.....	12
2.2 Aplicación Magnify para Android.....	14
2.3 Interfaz de la aplicación Georgie.....	15
2.4 Diseño conceptual <i>Squibble</i>	16
3.1 Esquema de menú	24
3.2 Diagrama de casos de uso.....	29
3.3 Esquema de la interfaz.....	31
3.4 Elementos del menú.....	33
3.5 Esquema de la interfaz 2.....	34
3.6 Diagrama de flujo principal.....	37
4.1 Eventos principales de las actividades	39
4.2 Diagrama de flujo. Menú principal	42
4.3 Diagrama de flujo. Lista de archivos de audio	44
4.4 Diagrama de flujo. Actividad de configuración	45
4.5 Diagrama de flujo. Reproductor de audio	47

1. Introducción

Los productos de interacción humana dirigidos al consumidor que se encuentran dentro del campo de la informática y las telecomunicaciones, en cuanto a su desarrollo tecnológico, están avanzando a un ritmo cada vez más acelerado. Los productos más comerciales son: las cámaras, pantallas planas, computadoras, tabletas digitales, reproductores de música, y teléfonos celulares.

Hoy en día tenemos teléfonos con una inmensa cantidad de aplicaciones creadas para realizar todo tipo de tareas: desde entretenimiento, reproducción de audio y video, hasta domótica, o aplicaciones para reconocimiento de imágenes u objetos en el espacio. A estos teléfonos se les conoce como *Smartphone*. También son muy comunes ya los teléfonos que incorporan programas avanzados que trabajan con GPS para darnos nuestra posición exacta en tiempo real, o la de otra persona que tenga un dispositivo con la misma funcionalidad integrada.

Dependiendo del modelo del teléfono, este puede llegar a ser compatible con aplicaciones que exijan de un hardware con mayor capacidad de procesamiento y gráficos.

Aparte de lo que incluye el aparato por sí solo, en el mercado hay accesorios que mejoran la experiencia del uso del teléfono.

1.1 Motivación

Una característica común en los Smartphone, desde el lanzamiento del *I-Phone*, es la incorporación de la pantalla táctil. Esta se ha convertido en el estándar para los *Smartphone*, ya que resulta más llamativo para el usuario por su aspecto, y también porque ofrece mayor libertad de manejo, y más posibilidades tanto para el desarrollador como para el usuario.

Sin embargo no muchos de los productos creados para estos dispositivos van dirigidos al sector de usuarios invidentes. Según los datos del Censo del 2010, el 1.16% de la población en México tiene problemas de visión¹. Esta es una cifra que supera el millón.

La mayor parte de nuestra experiencia cultural de los medios consiste en una mezcla de texto, imágenes y sonidos, pero aun así el formato visual juega el rol más importante². Asimismo, los teléfonos móviles no están diseñados tomando en cuenta a las personas invidentes, y la pantalla táctil puede dificultar aun más el uso del teléfono a aquellos usuarios que se guían a través del tacto y el sonido. Generalmente una persona invidente, o de visión reducida, tendría mayores dificultades al utilizar la pantalla táctil. Un teléfono básico con teclado numérico y que no cuenta con pantalla táctil no tiene esta desventaja, ya que el teclado permite a uno guiarse mediante el tacto.

Puede resultar más fácil para una persona invidente realizar una llamada con un celular básico que incorpore un teclado numérico, sin embargo como objetivo de esta tesis se

¹ <http://www.inegi.org.mx/>

² Cattaneo, Zaira, Vecchi, Tomaso, " Blind Vision : The Neuroscience of Visual Impairment", MIT Press, 2011.

pretende desarrollar una aplicación que el usuario pueda utilizar al 100% sin el uso de la vista.

Pienso que esta es una oportunidad para crear algo innovador y que posiblemente pueda llegar a ser de utilidad a los usuarios de los Smartphone. Con este proyecto espero crear una alternativa innovadora a las aplicaciones ya existentes.

Biblioteca UP Bonaterra

2. Marco Teórico

2.1 Introducción

Existen muchos tipos de productos para personas invidentes: desde software para computadoras y teléfonos móviles, hasta pantallas táctiles o líneas braille. Los I-Phone, y algunos dispositivos Android como los Samsung, también incluyen la funcionalidad de texto a audio dentro de la interfaz de su sistema operativo.

Los teléfonos móviles son el mayor segmento de los dispositivos móviles³. Recientemente los teléfonos más comunes son los Smartphone, y los hay de distintas marcas y precios. Con un Smartphone puede resultar más difícil para una persona invidente interactuar con la pantalla, ya que la interfaz es más compleja que los teléfonos anteriores (la cuál solía ser una secuencia de opciones dentro de un menú), la pantalla táctil es lisa y el dispositivo contiene botones o íconos desplegados en distintos puntos de la pantalla, los cuales requieren el uso de la vista, o texto a audio.

Al igual que la aplicación Georgie para Android, que es la primera interfaz para Smartphone dirigida a personas con visión reducida, el propósito de esta tesis es crear una aplicación para reproducir archivos de audio, como música mp3.

³ Mehta, Nirav "Mobile Web Development" Packt Publishing Ltd., 2008.

2.2 Tecnología existente

Actualmente existe una gran variedad de productos electrónicos dirigidos a personas invidentes. Hay desde teléfonos inteligentes con herramientas de accesibilidad y reconocimiento de voz, a software que facilita el uso del sistema operativo de una computadora. También existe una gran variedad de aplicaciones para Smartphone.

2.2.1 Aplicaciones Existentes

El I-Phone contiene un gran repertorio de aplicaciones: Desde calculadoras, pronósticos del tiempo, programas para el GPS con manejo asistido por audio, a detectores de luz y de colores. La empresa española Code Factory desarrolló una aplicación que convierte el contenido escrito de la pantalla de un teléfono en sonido. A continuación se mencionan algunos ejemplos:

- **NoLED** es una aplicación para Android que mostrará las notificaciones en la pantalla principal de dispositivo como pequeños puntos, o *LEDs*, de distintos colores, los cuales pueden ser fácilmente identificados por el usuario como notificaciones de mensajes SMS, correo electrónico, mensajes de voz, etc. Los puntos que aparecen en la pantalla tienen distintos tonos configurables que resaltan sobre un fondo negro, aparentando ser luces LED encendidas.

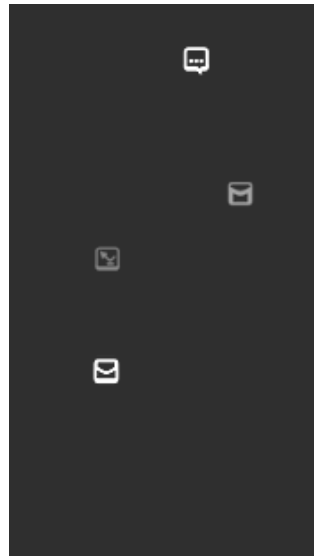


Figura 2.1: Aplicación NoLED.

Fuente: www.engadget.com

- **TalkBack:** Se trata de una aplicación, la cual es parte del servicio de Accesibilidad de Google. Esta aplicación está diseñada para asistir a las personas con visión reducida a navegar por la pantalla. La aplicación le dirá al usuario lo que ha seleccionado en la pantalla. Esta también puede leer texto y monitorear los movimientos que el usuario hace en su teléfono y leerlos en voz alta. TalkBack viene incluida en los dispositivos Android de nueva generación.
- **Classic Text to Speech Engine:** Esta aplicación desarrollada por SVOX Mobile Voices sirve para leer texto de archivos de texto como PDFs o e-books con la herramienta integrada del programa de texto a voz. Esta aplicación incluye 40 voces distintas, con diferentes acentos según el idioma elegido por el usuario. Classic Text to Speech Engine se puede encontrar en la tienda de Google, para los dispositivos Android.
- **Color ID Free:** Se trata de una aplicación de uso sencillo, disponible para dispositivos iOS. Esta aplicación empleará la cámara del dispositivo para indicar al usuario el color

de un objeto capturado. Puede ser utilizado para saber el color de la ropa, o que tan despejado está el cielo el día de hoy.

- **Light Detector:** es una aplicación dirigida a usuarios invidentes, similar a la aplicación anterior en esta lista, pero con la diferencia de que esta emite un sonido cuya frecuencia varía dependiendo de la intensidad de la luz capturada por la cámara del dispositivo. Puede ser empleado para ayudar al usuario a detectar una fuente de luz, por ejemplo, si se dejó encendida una luz en la casa, o para detectar las ventanas o salidas de un edificio.
- **Evernote:** Una aplicación que puede grabar notas de voz, fotografías y texto el cual puede ser accesado desde distintos dispositivos. Tiene una herramienta de texto a voz, que puede escanear y leer el texto capturado en una fotografía.
- **Magnify** es una aplicación para Android que aumenta el tamaño de la imagen capturada por la cámara del dispositivo. También tiene la opción de encender la linterna del teléfono para una mejor visualización de la imagen. Esta sencilla, pero útil aplicación puede ser de gran ayuda para cualquier usuario, en especial para aquellos con visión reducida, a poder visualizar mejor el texto o detalles pequeños en cualquier imagen.

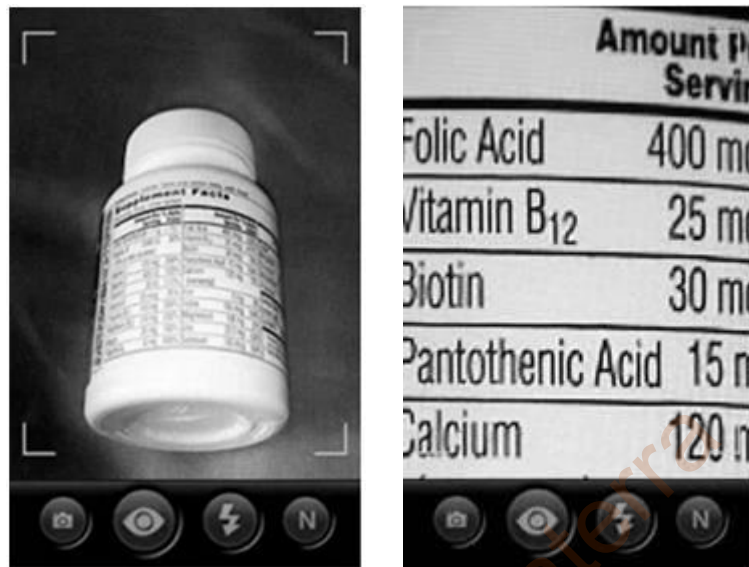


Figura 2.2: Aplicación *Magnify* para Android.

Fuente: play.google.com

- **Georgie:** La aplicación Georgie, para teléfonos Android permite a un usuario con visión reducida un manejo más fácil del Smartphone: cuenta con un menú sencillo, y botones de mayor tamaño que muestran iconos fáciles de identificar: son alargados y de tono blanco que contrasta con el fondo azul oscuro. Esta aplicación fue creada por Roger Wilson-Hinds. Una de las funciones más útiles de esta aplicación es un lector de texto que utiliza la cámara del teléfono.



Figura 2.3: Interfaz de la aplicación Georgie.

Fuente: www.engadget.com

Biblioteca UP Bonaterra

2.2.2 Productos Electrónicos

A pesar de que esta tesis trata sobre el desarrollo de una aplicación para un Smartphone, y no incluye la elaboración de ningún aparato o periférico, se incluyen los siguientes solamente como referencia.

Existen muchas ideas y diseños conceptuales de celulares e interfaces braille para ciegos, como el *Squibble Portable Braille Interface* del diseñador Andrew Mitchell, así como productos comúnmente empleados por personas invidentes que les permite realizar todo tipo de tareas, como navegar por internet o leer un periódico. Una herramienta común es el *Braille*, un tipo de máquina de escribir.



Figura 2.4: Diseño conceptual *Squibble*.

Fuente: <http://andrewmitchelldesign.co.uk/>

Los lectores de pantallas son comunes: constan de un software que lee en voz alta el texto que se muestra en la pantalla de una computadora o un teléfono. También los hay en versiones que funcionan de forma similar a un escáner, como el *ClearReader* de la empresa Optelec, utilizado para leer información impresa, como por ejemplo: un periódico o una revista. También están las líneas braille, que muestran de forma táctil la información que un lector de pantalla leería.

Asimismo, hay ampliadores digitales de texto, teclados para computadoras braille similares a las máquinas Braille, impresoras braille y reproductores mp3.

Biblioteca UP Bonaterra

2.3 Sistema Operativo del Dispositivo

El tipo de dispositivo empleado para el desarrollo de este proyecto será un Smartphone con Android 4.2.1 (Jelly Bean) integrado.

Android es un Sistema Operativo móvil basado en el kernel Linux y actualmente es desarrollado por Google. El Sistema Operativo Android contiene una interfaz basada en manipulación directa, y está diseñado principalmente para funcionar con dispositivos que incorporen una pantalla táctil, como los Smartphone o las Tabletas, entre muchos otros más especializados. El sistema operativo usa gestos o Inputs que corresponden a acciones del mundo real, como tocar, o "tap", arrastrar o pellizcar para manipular objetos en pantalla.

Se tomó la decisión de desarrollar para este sistema operativo por varios motivos, principalmente porque me interesa desarrollar con en lenguaje Java, que se emplea para escribir código nativo en Android. Otro motivo fue la versatilidad que puede ofrecer Android, al haber en el mercado una vasta cantidad de dispositivos que integran este sistema operativo, entre ellos los Smartphone LG y Samsung. Cada uno de estos dispositivos tiene distintas especificaciones técnicas, como capacidad de hardware o resolución y tamaño de la pantalla.

Debido a que Android es código abierto, no hay muchos obstáculos que puedan impedir a los fabricantes emplear esta plataforma. Pueden crear dispositivos para todos los precios, modificando el mismo sistema Android para ajustarse al poder de procesamiento de un dispositivo en **específico**. Gracias a esto, Android no está limitado a dispositivos de alta

gama, sino que además puede ser desplegado en dispositivos de bajo costo, de esta forma alcanzando un mercado más amplio⁴.

2.4 Lenguaje y herramientas de desarrollo

Android proporciona un vasto framework que te permite crear aplicaciones innovadoras y juegos para dispositivos móviles en un entorno de lenguaje Java.

Las herramientas Android SDK compilan tu código, junto con cualquier tipo de dato o archivo compatible, en un paquete Android, o archivo APK con .apk como sufijo. Un archivo APK contiene todo el contenido de una aplicación Android y es el archivo que los dispositivos que incorporan Android utilizan para instalar la aplicación⁵.

2.4.1 Componentes de una aplicación Android

Los componentes son la parte esencial para construir los bloques de una aplicación Android. Cada componente es un punto diferente por el cual el sistema puede tener acceso a tu aplicación. No todos los componentes son puntos de acceso para el usuario, y algunos dependen uno del otro, pero cada uno existe como entidad propia y juega un rol específico, y ayudan a definir el comportamiento de tu aplicación.

⁴ Zechner, Mario, y Green, Robert, "Beginning Android Games Second Edition", Apress, 2012.

⁵ <http://developer.android.com/>

Existen cuatro tipos de componentes. Cada uno sirve un propósito distinto y tiene su propio ciclo de vida que define cómo el componente es creado y destruido. Estos componentes son los siguientes:

Actividades: Representan una sola pantalla con una interfaz de usuario. Son las pantallas con las que el usuario interactúa directamente. Por ejemplo en una aplicación de redes sociales, una pantalla sería utilizada para crear una cuenta, otra para ingresar los datos de usuario, y otra para leer o publicar mensajes de otros usuarios.

Servicios: Es un componente que funciona en segundo plano, o el fondo de la aplicación. Un servicio no provee una interfaz de usuario.

Proveedores de Contenido: Dirigen un conjunto de datos compartidos, empleado para el manejo de datos de la aplicación con bases de datos o cualquier tipo de ubicación de almacenamiento de datos a la que la aplicación tenga acceso.

Broadcast Receiver: Un componente que responde a anuncios generales del sistema, como por ejemplo si la pantalla no está activa, o si el nivel de carga de la batería es bajo.

Un aspecto único y beneficioso del sistema Android es que cualquier aplicación puede arrancar un componente de otra aplicación, haciendo las aplicaciones compatibles entre sí con el fin de ahorrar código y tiempo de desarrollo.

2.4.2 Recursos de una aplicación Android

Una aplicación Android, aparte de contener los archivos con código fuente Java, requiere de recursos adicionales. Al externalizar los recursos, la tarea de mantener,

actualizar y manipularlos se vuelve más sencilla⁶. Estos recursos pueden ser archivos de audio, imágenes, o archivos XML utilizados para definir animaciones, menús, estilos, colores y el layout de la Actividad.

Estos recursos permiten modificar varios aspectos de la aplicación sin tener que modificar el código fuente, lo cual ayuda a que el proceso de desarrollo sea más eficiente, y también permite optimizar la aplicación para distintos tipos de dispositivos, ya que existe una gran cantidad de dispositivos Android en el mercado.

2.4.3 Eclipse con ADT

ADT (Android Developer Tools) es un plugin para Eclipse que provee un conjunto de herramientas que son integradas al IDE Eclipse. Ofrece acceso a muchas características que ayudan al desarrollo de aplicaciones para Android. Este fue el IDE oficial de Android, previo al IDE actual, Android Studio.

Además de las características de Eclipse, el plugin de Android ADT ofrece editores XML personalizables para crear varios archivos como los *layout* o el manifiesto de la app. También contiene el editor gráfico de layout, que permite previamente visualizar el diseño de una pantalla, y editarla sin necesidad de utilizar código XML. Esto resulta muy práctico a la hora de diseñar la interfaz de usuario.

⁶ Meier, Reto, "Professional Android 4 Application Development", John Wiley & Sons, Inc., 2012.

Debido al que el lanzamiento de Android Studio es reciente, decidí continuar utilizando Eclipse, ya que es el programa con el que he desarrollado hasta ahora y en el que se comenzó a escribir la aplicación para esta tesis. Tras concluir la tesis, dado el caso de que se continúe el desarrollo de la aplicación o su mantenimiento, pretendo migrarla a Android Studio.

Biblioteca UP Bonaterra

3. Diseño de la Aplicación

3.1 Problemática

El objetivo de este proyecto es desarrollar una aplicación Android la cuál contendrá distintos menús y funcionalidades usadas con interfaz basada en audio. La aplicación leerá por audio la opción señalada por el usuario, y se darán instrucciones adicionales dependiendo del contexto. El menú está formado por los siguientes elementos: Un layout base, y un layout emergente que contendrá los botones de navegación y los botones para usar las distintas funciones de la aplicación

Un elemento que hace esta aplicación única es el menú emergente basado en un botón inicial, que actúa de la siguiente manera: el cuadro que encapsula los botones, en vez de estar fijo en la misma posición dentro la pantalla, aparece en el punto en donde el usuario hace contacto. De esta manera el usuario no empleará la vista para encontrar el botón, cuya posición normalmente estaría predefinida en un punto fijo del área activa de la pantalla táctil.

A grandes rasgos el menú actúa de la siguiente forma:

1. La aplicación recibe el input del usuario (el usuario toca la pantalla en cualquier punto o coordenada X Y del área activa).
2. El cuadro del menú aparecerá debajo del punto inicial Y.
3. El usuario arrastrará el punto de contacto hacia abajo y pasará por las distintas opciones disponibles, y serán leídas en audio una por una. A continuación se muestra un esquema que lo ejemplifica.



Figura 3.1: Esquema de menú.
Fuente: Propia.

Todas las indicaciones e instrucciones serán recibidas por audio. De este modo el reto de este proyecto es que la interfaz sea sencilla, y que sea posible su uso sin ayuda visual.

3.2 Aplicación propuesta

La aplicación será un menú que funcionará con instrucciones por audio para que el usuario pueda reproducir archivos de música y audio libros. Las instrucciones por audio facilitaran al usuario el uso de sus múltiples herramientas, como cambiar la configuración de la aplicación, elegir un archivo mp3 de la lista de canciones y seleccionar las distintas opciones del reproductor mp3 (Play, pausa, siguiente archivo...).

En un dispositivo con botones tangibles, el usuario sabe donde se encuentra cada opción mediante el tacto. En una pantalla táctil, es necesario el uso de la vista para ubicar cada botón.

Una característica distintiva de esta aplicación, es que el contexto o cuadro emergente que contendrá las opciones o elementos del menú no se encontrará en una posición fija, sino que será creado en el punto en el que el usuario toque la pantalla.

Esta funcionalidad será clave para permitir al usuario ubicar fácilmente el encabezado del menú, que se encontrará en el punto donde toque la pantalla. Gracias a este sistema el usuario podrá navegar por el menú sin requerir de apoyo visual, simplemente mediante la ayuda de instrucciones por audio.

3.3 Análisis y requerimientos de la aplicación

A continuación se mencionan los requerimientos funcionales, requerimientos no funcionales, y los requerimientos del usuario para la aplicación.

3.3.1 Requerimientos funcionales

El usuario deberá poder navegar por la interfaz basada en audio. Deberá de ser un proceso intuitivo y con una curva de aprendizaje mínima. El usuario constantemente recibirá indicaciones para saber en qué menú de la aplicación se encuentra.

Empleando este sistema, el usuario deberá de poder seleccionar las opciones deseadas de la aplicación, como reproducir el archivo de audio correcto, o cambiar la configuración de la aplicación.

El cuadro o layout que contendrá el menú será creado debajo del punto en el que el usuario haga contacto con la pantalla. El layout emergente contendrá las opciones disponibles, dependiendo de la pantalla que este activa. Estas opciones pueden ser específicas de la pantalla, como cambiar el idioma en el menú de herramientas del app, o también pueden ser opciones para navegar (Pantalla anterior, etc.) .

El usuario se desplazará hacia abajo. Conforme vaya deslizando el dedo, pasará por las distintas opciones. Cuando el cursor esté sobre la opción que el usuario desee elegir, retirará el dedo de la pantalla y la aplicación pasará a la pantalla correspondiente.

El usuario podrá navegar hacia la pantalla anterior pulsando brevemente, o con un tap en cualquier punto de la pantalla. Presionar el botón menos de una fracción de segundo será considerado por la aplicación como una orden para retroceder.

Al pasar de una pantalla a otra, sonará un pitido que hará saber al usuario que cambió de pantalla. El tono será distinto si el usuario navega hacia adelante o si retrocede al menú anterior.

El usuario sabrá en que menú se encuentra ya que al presionar el display, será indicado por audio el menú o pantalla actual.

Si el usuario tocó la pantalla, pero no desea realizar ninguna opción, soltará la pantalla cerca del punto inicial, siempre y cuando no sea interpretado por el programa como un tap, en cuyo caso retrocederá a la pantalla anterior.

El usuario podrá desplazarse por el menú para escuchar las distintas opciones o archivos de audio correspondientes a cada botón, regresar al punto de origen (primer botón en el

menú), y dejar de hacer contacto con la pantalla para continuar en la misma actividad/menú.

Las opciones deben de estar a una distancia razonable del punto inicial de contacto, para un manejo más cómodo. Poner la opción muy retirada o demasiado cerca podría ser la causa de que el usuario cometa un error al intentar seleccionar una opción.

El programa debe ser confiable y los menús bien estructurados. Con este tipo de programa ligero en gráficos, puede resultar más difícil detectar un bug.

3.3.2 Requerimientos no funcionales

El hardware que correrá el programa deberá de ser un aparato de media gama para poder cumplir con los siguientes requerimientos:

- Eficiencia: La aplicación no contará con gráficos pesados, por lo cual tendrá buen rendimiento. Esto es importante ya que el usuario podría no percibir un retraso del programa (lag) debido a procesamiento de variables o gráficos, si este llega a ocurrir en cualquier momento, ya sea tras seleccionar una opción, o pasar a un menú distinto. Por este motivo es importante que el programa sea eficiente.
- Efectivo: El programa realizará las funciones (enlistadas más adelante) que el usuario seleccione, y que requiera el menor esfuerzo posible.
- Factores emocionales: Deberá de ser divertido, o entretenido para el usuario utilizar la aplicación. Para que la experiencia no resulte tediosa, uno de los requerimientos del hardware es que no tenga una pantalla demasiado pequeña.
- Performance: Tiempo de respuesta que hay desde que es detectado el input del usuario a que la aplicación responda. Debe de ser lo más rápido posible.

La arquitectura del sistema hará posible que se cumplan los siguientes requerimientos:

- Extensibilidad: Será posible agregar nuevas funciones al programa (por ejemplo, que tenga radio, o que pueda sincronizarse a una base de datos en la nube).
- Mantenibilidad: Corrección de defectos detectados a futuro y mejora de la eficiencia y confiabilidad del programa.
- Precio: Accesible. No se necesitará un equipo costoso para correr la aplicación, la cual estará disponible de forma gratuita.
- Privacidad: La aplicación no necesitará recolectar y almacenar información del usuario.
- Calidad: Los defectos serán corregidos según la prioridad. Se comenzará con los que afecten el manejo de la aplicación.

3.3.3 Requerimientos del usuario

El usuario podrá disponer de un nuevo tipo de interfaz para Smartphone que le permitirá realizar diversas funciones. Se guiará solamente con el tacto (la pantalla touch screen), e indicaciones por audio. Las funciones que el usuario realizará son las siguientes:

- Escuchar la lista de archivos de audio.
- Seleccionar una canción de la lista para reproducir.
- Controlar el reproductor (Pasar a la siguiente canción, pausar la canción o apagar el reproductor de audio).
- Configurar la pantalla de la lista de audio.
- Cambiar el idioma de la aplicación.
- Cerrar la aplicación.

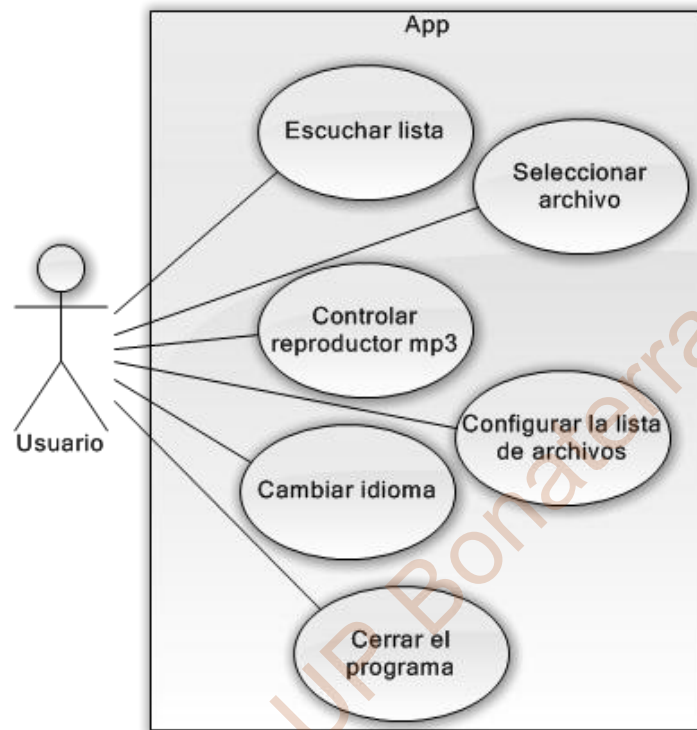


Figura 3.2: Diagrama de casos de uso.

Fuente: Propia.

Esto viene siendo una alternativa a los productos existentes, con la diferencia de que esta aplicación está diseñada para poder ser usada al 100% sin necesidad de usar la vista. Va dirigida tanto a personas con visión reducida, como a personas con ceguera.

La aplicación es sencilla y de fácil manejo. Puede ser utilizada por cualquiera, y la curva de aprendizaje es mínima.

3.4 Características del diseño de la aplicación

La aplicación contendrá las pantallas necesarias para cumplir con el objetivo propuesto, que es el de crear un reproductor de archivos de audio. Para lograr este objetivo, todas las pantallas serán accedidas con el sistema de menú emergente.

Cada pantalla o actividad en la aplicación compartirá esta misma característica.

3.4.1 Funcionamiento de la Interfaz

El tipo de interfaz de la aplicación es poco convencional. Los I-Pod Shuffle se controlan mediante audio, sin el uso de un display. Sin embargo, controlar un Smartphone de la misma manera resulta más complejo.

El elemento clave que hará posible esta interacción es el menú emergente, que aparecerá en el punto del input del usuario. De esta forma este se guiará por el tacto y sonido, y no por la vista o memoria.

El menú emergente abarcará el ancho de la pantalla. El programa tomará la coordenada Y (vertical) del input del usuario para decidir donde posicionar el menú emergente.

A continuación se muestra un ejemplo en la figura 3.2.

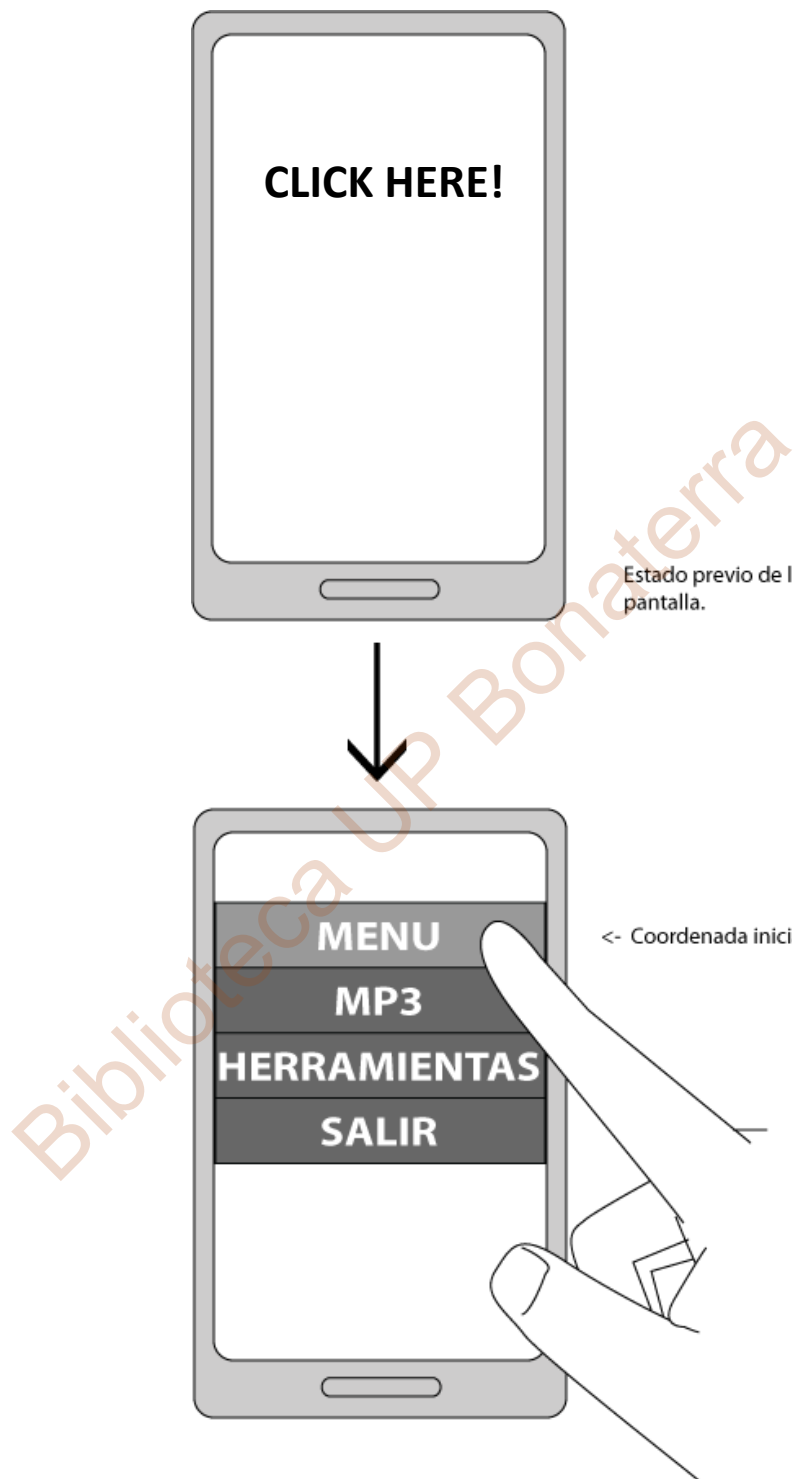


Figura 3.3: Esquema de la interfaz 2.
Fuente: Propia.

El menú permanecerá en la misma posición hasta que el usuario deje de tocar la pantalla. Dependiendo del punto que el usuario suelte la pantalla, es interpretado por el programa de la forma correspondiente para activar el evento deseado.

En el caso de que el usuario quiera retroceder en el menú, solamente tendrá que tocar rápidamente (tap) cualquier punto del área activa de la pantalla, o seleccionar la última opción del menú.

3.4.2 Acciones del layout

Presionar y arrastrar el cursor reproduce audio, con el fin de hacer saber al usuario su ubicación dentro del menú. Desplazar el cursor solamente reproduce las indicaciones por audio, no selecciona ninguna opción ni comienza una actividad distinta. Para que una opción sea seleccionada, el usuario soltará el display cuando se encuentre en el punto correspondiente a una opción válida dentro del menú.

Como ejemplo se muestran a continuación las opciones del menú principal, en donde se puede apreciar cómo el botón inicial, punto de origen del layout emergente que contiene el nombre de la actividad, se sitúa en la parte superior del menú, seguido de cada una de las opciones.



Figura 3.4: Elementos del menú.

Fuente: Propia.

Para seleccionar una de las opciones, el usuario arrastra el puntero hacia abajo y se guiará con el sonido, como se muestra a continuación en la figura 3.4:



Figura 3.5: Esquema de la interfaz 2.

Fuente: Propia.

Estas son las acciones que se realizan con el botón en el menú principal:

1. Deslizar hacia abajo y soltar: Ir a "MP3", "HERRAMIENTAS" y "SALIR".
2. Tap: Cancelar/Salir de la aplicación.

Asimismo, presionar y deslizar el botón sin soltar causa lo siguiente:

Presionar: Indica en qué menú se encuentra el usuario. Soltar en ese mismo punto para no realizar ninguna acción, mientras no sea tap (evento de presionar y soltar con una duración total de 1/4 de segundo).

Deslizar hacia abajo y mantener presionado: Señala por audio las opciones del menú.

3.5 Versiones

El proyecto no se desvió notablemente de lo que se planeaba en su etapa inicial. Las versiones previas a la actual varían en cuanto a algunos detalles menores, por ejemplo el ancho del menú contextual en el diseño de la interfaz, o la ubicación de algunas opciones y la estructura del menú.

3.5.1 Versión 1.0

Reutilizando la idea central y las herramientas desarrolladas durante la primera versión del proyecto, decidí centrarme en realizar una aplicación para reproducir archivos de audio, como música o audio libros. Esta es una de las funciones que se pretendía implementar parcialmente al programa durante la fase de diseño de la versión 1.0.

Opte por crear esta versión al ser la decisión más práctica e innovadora: actualmente existen varios reproductores de audio para dispositivos móviles, pero ninguno que indique por audio el título de la canción antes de seleccionarla, ni que incluya una serie de herramientas para el usuario que puedan ser utilizadas con la interfaz desarrollada para este proyecto.

3.6 Modelo de Menús

Las opciones del menú principal son las siguientes:

- MP3: Muestra la lista de archivos de audio hallados en el dispositivo.
- Herramientas: Accede al menú de configuración del usuario.
- Salir: Opción para salir de la aplicación. Otra opción sería el tap.

Cada una de estas pantallas podrán ser seleccionadas desde el menú principal. Se podrá regresar a este menú cancelando (presionando tap) la cantidad de veces requeridas, o tras un plazo de inactividad. A continuación se muestra un diagrama de árbol de este modelo:

Biblioteca UP Bonoloma

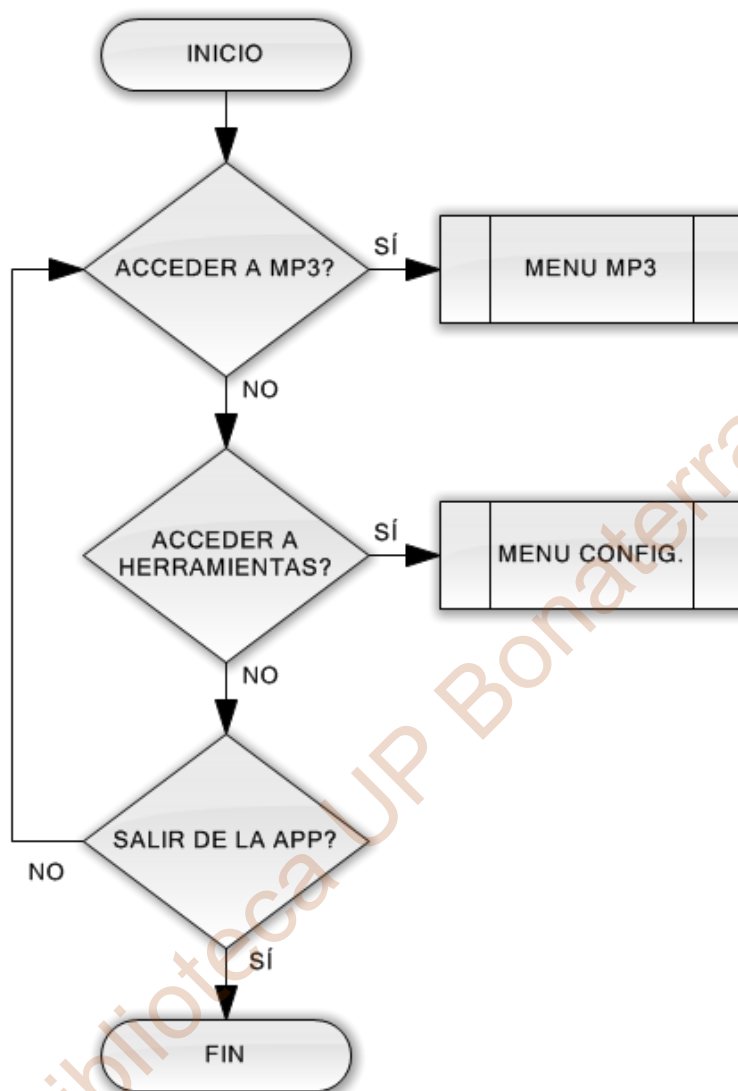


Figura 3.6: Diagrama de flujo principal.

Fuente: Propia.

4. Desarrollo de la Aplicación

Este capítulo describe los aspectos del desarrollo del programa realizado con el SDK de Android para Eclipse. Se indica principalmente el propósito y funcionamiento de cada una de las pantallas o menús del sistema, con el uso de los diagramas de flujo como apoyo para explicar el proceso de desarrollo.

4.1 Lógica y desarrollo del programa

Las aplicaciones Android típicamente son impulsadas por eventos. A diferencia de los programas de línea de comandos o scripts, las aplicaciones impulsadas por eventos comienzan y después esperan a que ocurra un evento, como un botón siendo pulsado⁷.

Cada una de las pantallas de esta aplicación tiene en común la parte de la lógica que crea el menú con la ayuda del evento `onTouchEvent()`. Varios fragmentos del código son reutilizados en cada una de las actividades.

A continuación se presenta un diagrama que contiene estos elementos compartidos en secuencia ordenada:

⁷ Phillips, Bill, y Hardy, Brian, "Android Programming: The Big Nerd Ranch Guide", Big Nerd Ranch, Inc. 2013.

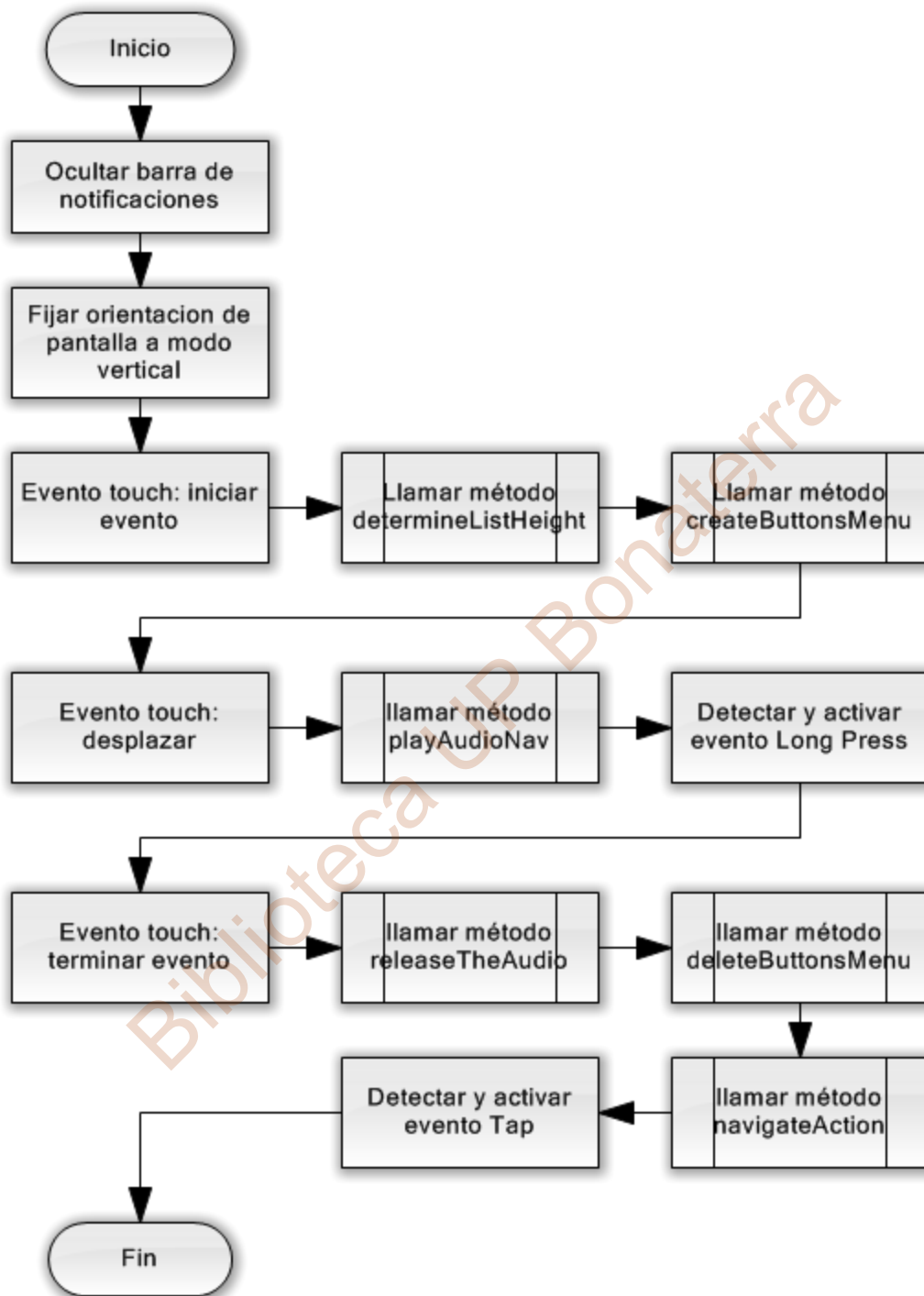


Figura 4.1: Eventos principales de las actividades.

Fuente: Propia.

Tras declarar las variables el programa excluye la barra de notificaciones, la cual se encuentra por defecto presente en las aplicaciones Android. Quitar esta barra de notificaciones reduce las probabilidades de error en esta aplicación.

El programa después fijará la orientación de la pantalla a modo vertical. Por motivos de diseño de la interfaz de la aplicación, esta funcionará solamente en este formato.

Los tres eventos principales son los siguientes:

- Iniciar evento touch: Este evento se activa en el momento en que el usuario toca la pantalla.
- Desplazar: Es el evento touch que detecta el desplazamiento. El evento va capturando las coordenadas X Y en tiempo real donde el usuario está haciendo contacto con la pantalla.
- Terminar evento touch: Se activa en el momento en que el usuario deja de tocar la pantalla.

El evento inicial manda a llamar el método "createButtonsMenu", el cual crea el cuadro donde se encuentran los botones del menú. La posición del cuadro varía dependiendo de la coordenada Y inicial donde el usuario haya hecho contacto. Estas coordenadas son capturadas por el evento que detecta el desplazamiento. Previo a este evento se manda a llamar "determineListHeight", que se encarga de la altura del layout emergente. La altura depende de la cantidad de botones en el cuadro, los cuales tienen medidas predefinidas.

El evento que detecta el desplazamiento manda a llamar el método "playAudioNav". Este método determina que instrucción por audio reproducir, dependiendo del punto Y dentro de la pantalla donde el usuario está haciendo contacto. También deja de reproducir la instrucción por audio una vez el punto Y se encuentra fuera del área designada para el botón.

Ese evento también detecta y activa el *Long Press*: una vez es activada esta rutina, lo único que hará de momento es reproducir instrucciones adicionales para el usuario.

El evento final primero mandará a llamar el método "releaseTheAudio". Esto programáticamente es necesario. El objeto de audio Android, MediaPlayer, debe de convertirse en nulo para que la instrucción pueda ser reproducida nuevamente. Tras esto se mandará a llamar "navigateAction", que se encargará de abrir la siguiente actividad correspondiente. Por ejemplo, si el usuario estaba señalando un archivo de audio en la lista del programa, la aplicación abrirá una nueva actividad que reproducirá este archivo. Al finalizar el ciclo de los tres eventos touch, también se detecta si el tap ha sido activado, en cuyo caso se cerrará la actividad actual y se pasará a la anterior.

4.1.1 Funcionamiento de las distintas pantallas

El programa al inicializar, comenzará en el menú principal de la aplicación. Desde aquí el usuario seleccionará una de las opciones siguientes:

- Mostrar archivos de audio.
- Mostrar configuración de usuario.
- Salir de la aplicación.

Previo a que el usuario seleccione una de las opciones, la aplicación cargará la configuración definida por el usuario, desde un archivo JSON. La aplicación creará el archivo de configuración JSON en el caso de que este no se encuentre (si es la primera vez que la aplicación es arrancada en el dispositivo).

Tras cargar la configuración, la pantalla del menú principal será iniciada. A continuación se muestra un diagrama del funcionamiento del menú principal.

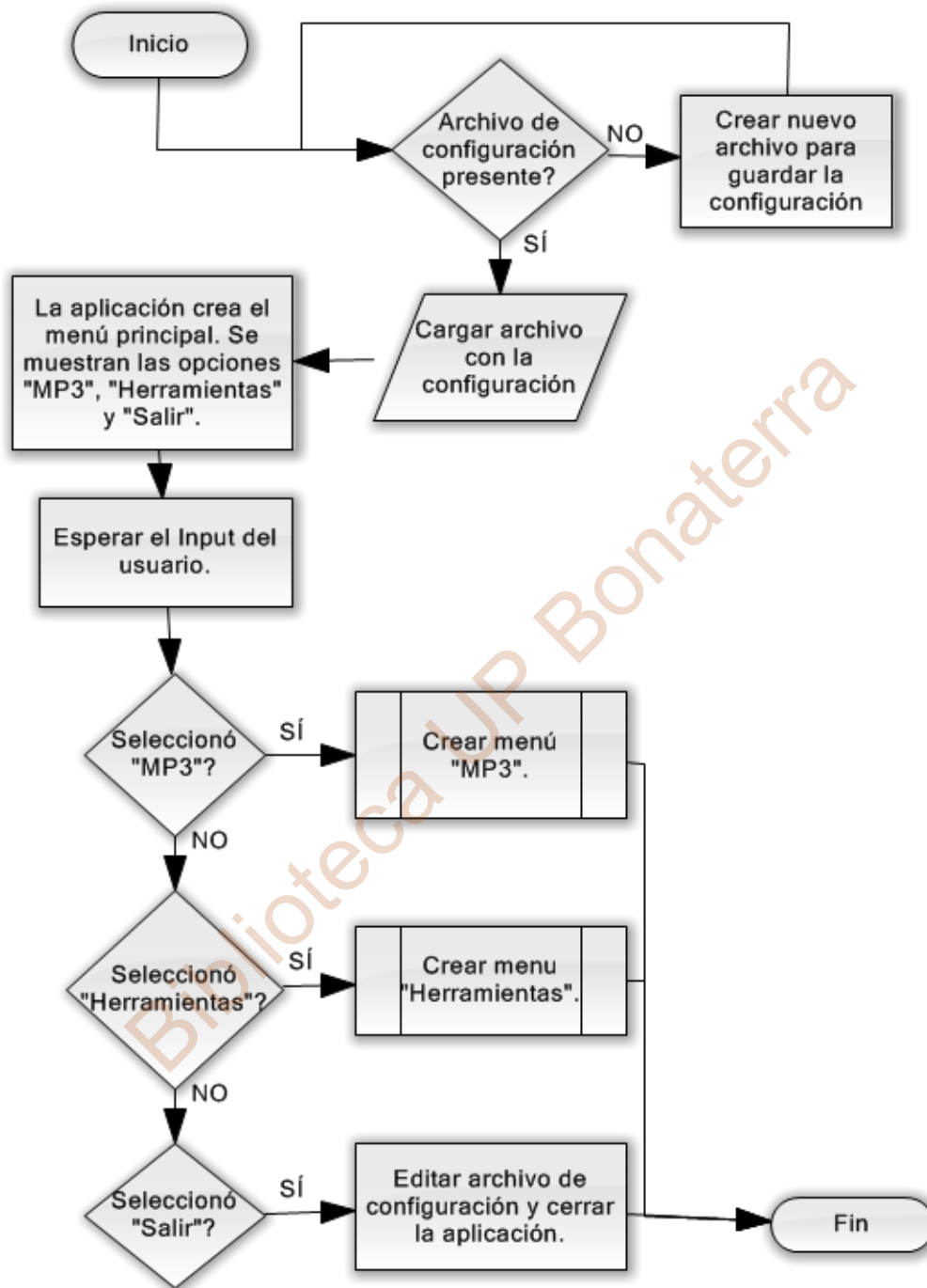


Figura 4.2: Diagrama de flujo. Menú principal.

Fuente: Propia.

La primera opción muestra la lista de archivos audio que pueden ser reproducidos, por ejemplo música o audio libros. Esta funcionalidad vendría siendo clave en este proyecto, ya que el propósito es crear un reproductor MP3 funcional. Los archivos de audio se mostrarán en una lista, la cual el usuario podrá revisar con el mismo sistema de menú: El título del archivo de audio aparecerá en cada botón, y será leído en voz alta con el método texto a voz de Android.

Biblioteca UP Bonaterra

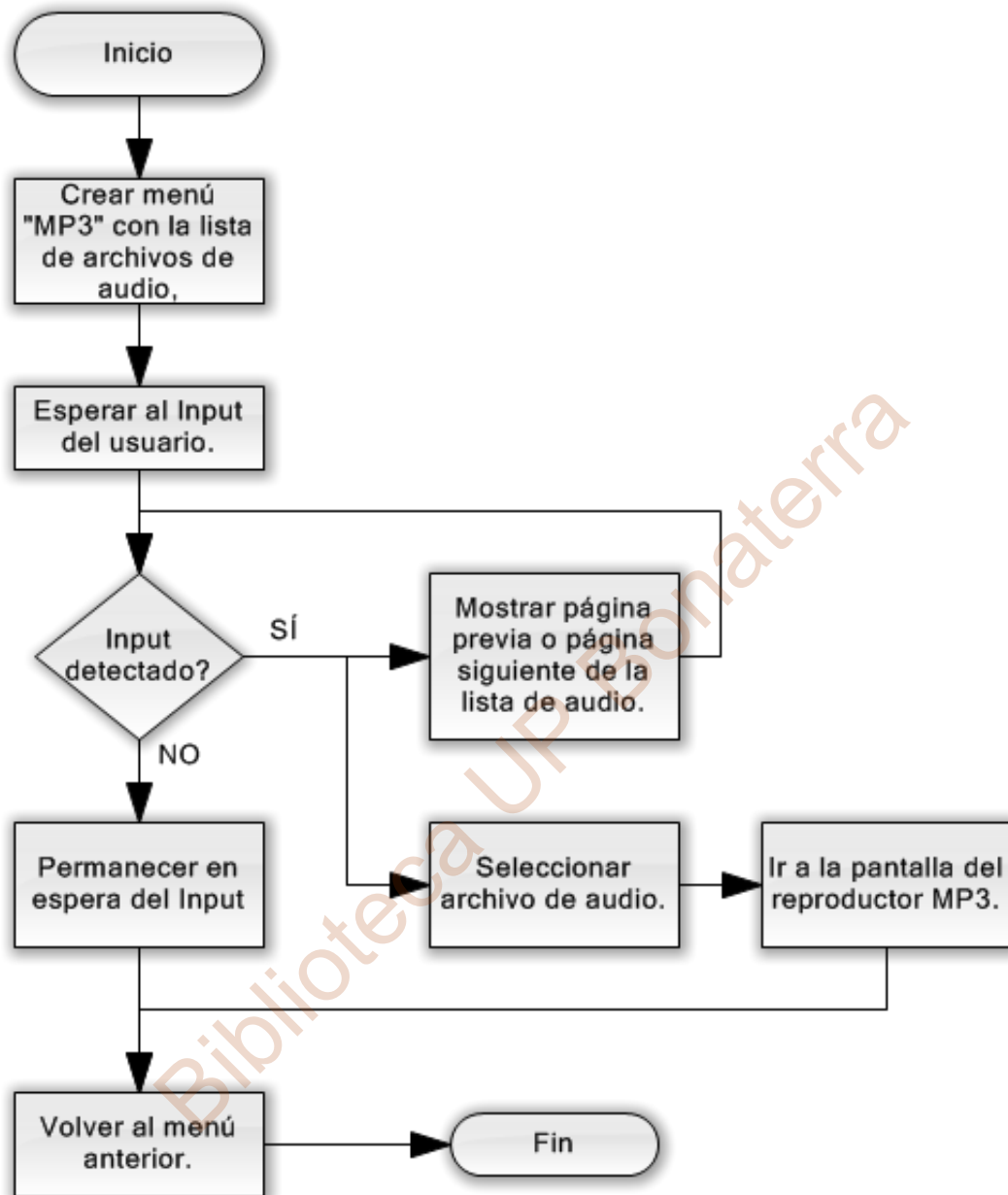


Figura 4.3: Diagrama de flujo. Lista de archivos de audio.

Fuente: Propia.

La opción que se muestra en el diagrama a continuación sirve para la configuración de usuario de la aplicación. En el proyecto han sido implementadas dos opciones, la primera

para leer el título del archivo de la lista de archivos de audio, o simplemente reproducir brevemente una muestra de la canción mientras el usuario está señalando el botón correspondiente. La segunda opción cambia el idioma de la UI de Español a Inglés.

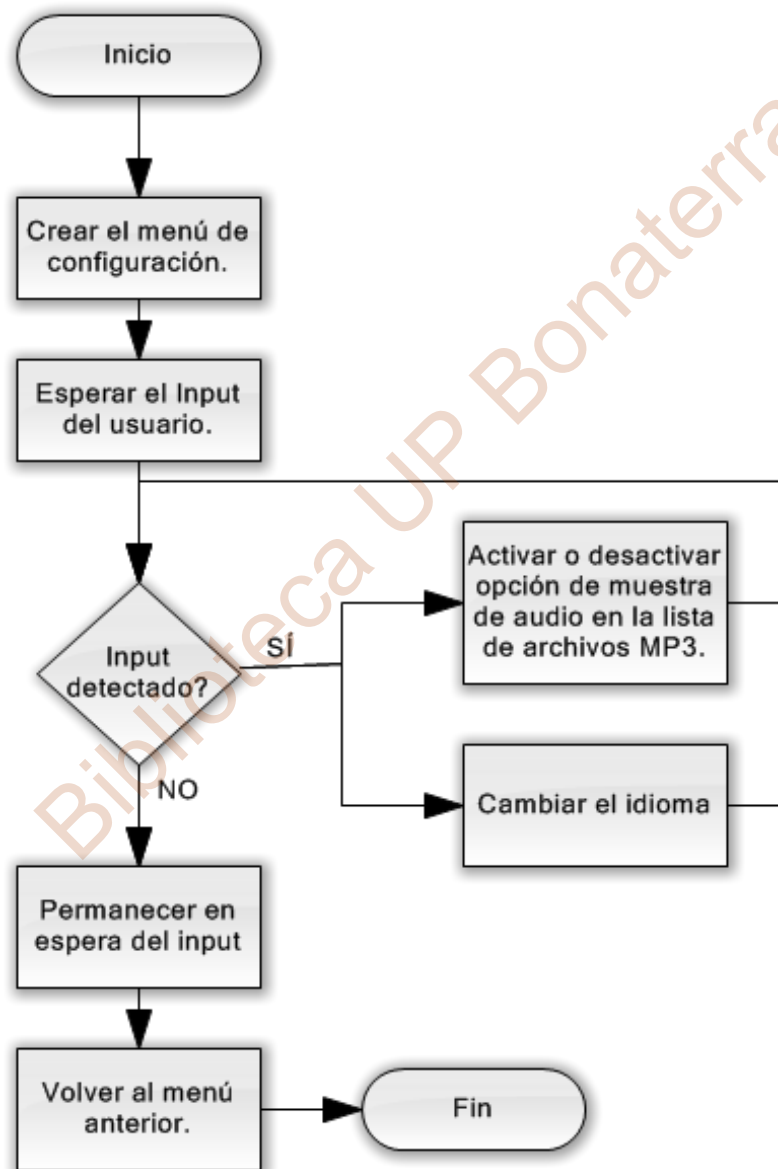


Figura 4.4: Diagrama de flujo. Actividad de configuración.

Fuente: Propia.

El siguiente diagrama muestra la actividad con el reproductor de audio. Funciona de la siguiente manera:

- El usuario toca el display mientras la pista de audio se está reproduciendo, y esta se detendrá temporalmente para que el usuario pueda escuchar los comandos por audio. Los comandos son los siguientes:
- Pausa, para detener el archivo que se está reproduciendo, pero continuar en la actividad del reproductor de audio. El mismo botón sirve como Play.
- Siguiente pista, y pista anterior.
- Stop: Esto detendrá el archivo de audio que se está reproduciendo, y volverá a la pantalla de lista de audio.

Biblioteca UP Bonaterra

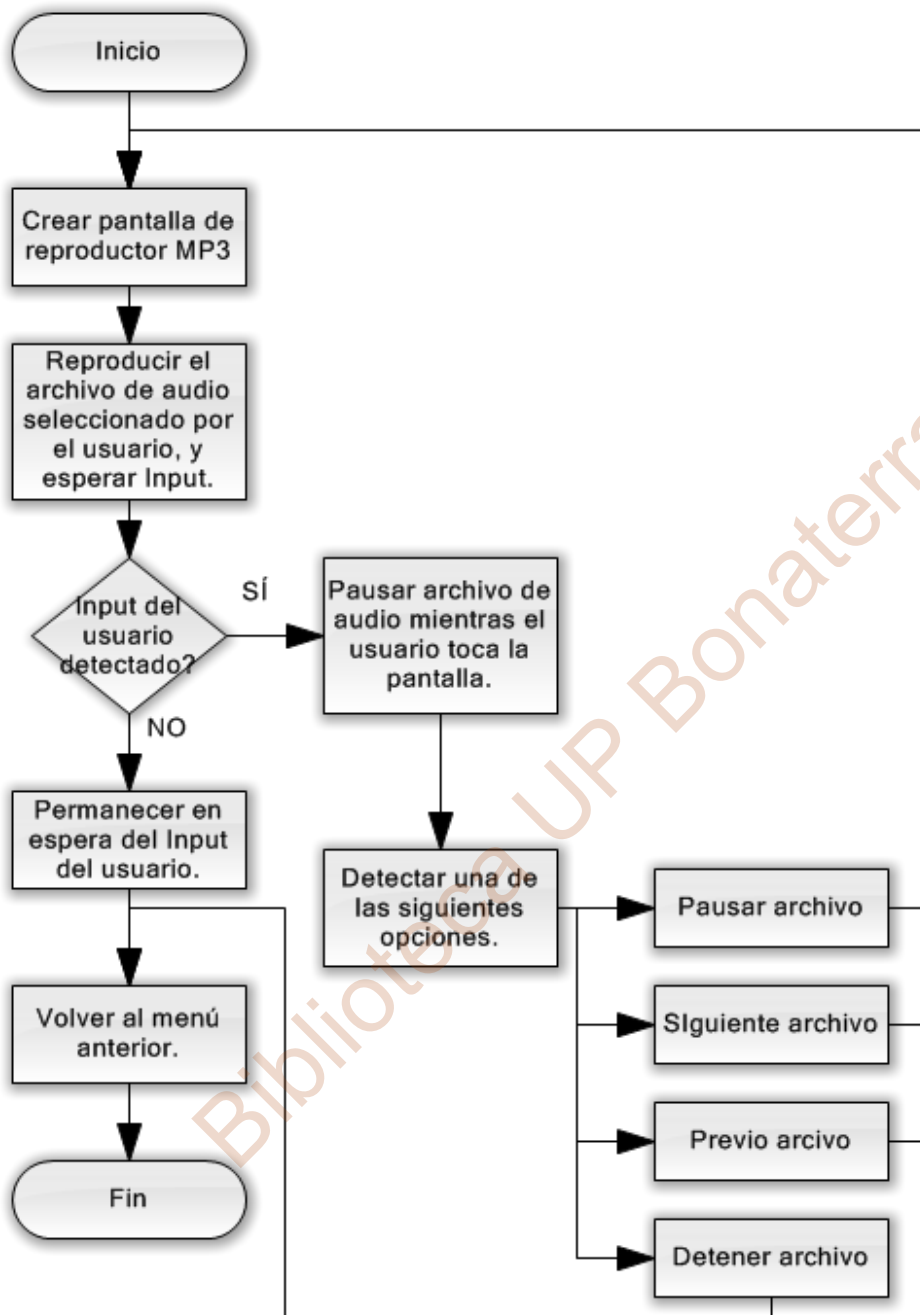


Figura 4.5: Diagrama de flujo. Reproductor de audio.
Fuente: Propia.

4.2 Experimentos y Resultados

Para lograr el funcionamiento de la aplicación se realizaron varios experimentos. Constantemente se corrió la aplicación con cada una de las clases o actividades del programa a lo largo de su etapa de desarrollo, con el propósito de hallar defectos o áreas de mejora. Esto facilitó la integración de los distintos componentes de la aplicación para dar forma al programa en su etapa final.

Cada uno de los componentes indicados son los que se mencionan a continuación:

- Actividad principal.
- Lista de Audio.
- Reproductor de audio.
- Menú de configuración.
- Menú de configuración 1 (Activar muestra de archivo de audio).
- Menú de configuración 2 (Idiomas).

Los experimentos fueron exitosos y permitieron una cantidad mínima de errores y de re trabajo a la hora de ensamblar la versión final 1.0 de la aplicación.

Inicialmente se realizó el debug del programa con el emulador integrado al SDK, sin embargo se consiguió una mayor eficiencia al migrar los experimentos a un dispositivo físico Lanix S700. Este fue el dispositivo seleccionado para correr la aplicación, ya que cuenta con una pantalla de mayor tamaño a las comunes, y logró ejecutar la app como se deseaba.

4.2.1 Resultados de la Encuesta Realizada

Se realizó una encuesta a diez participantes invidentes en la ciudad de Aguascalientes. La mitad de los encuestados asisten al Centro de Atención Múltiple (CAM) 6 de Aguascalientes.

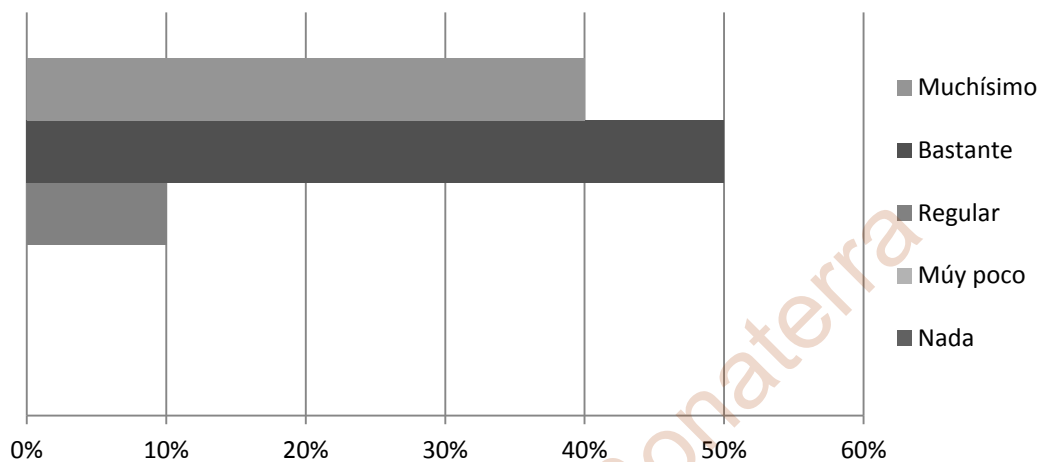
Las pruebas consistieron en demostrar a los participantes las funciones y características de la aplicación y cómo utilizarla. Cada uno de los participantes navegó a través de los menús y opciones, además de seleccionar distintas pistas de audio.

Tras la demostración de la aplicación, a cada participante se le hizo una serie de preguntas para determinar su nivel de satisfacción, además de dos preguntas adicionales (6 y 7) para ayudar a mejorar la aplicación, cuyas respuestas de mayor relevancia fueron incluidas en este documento.

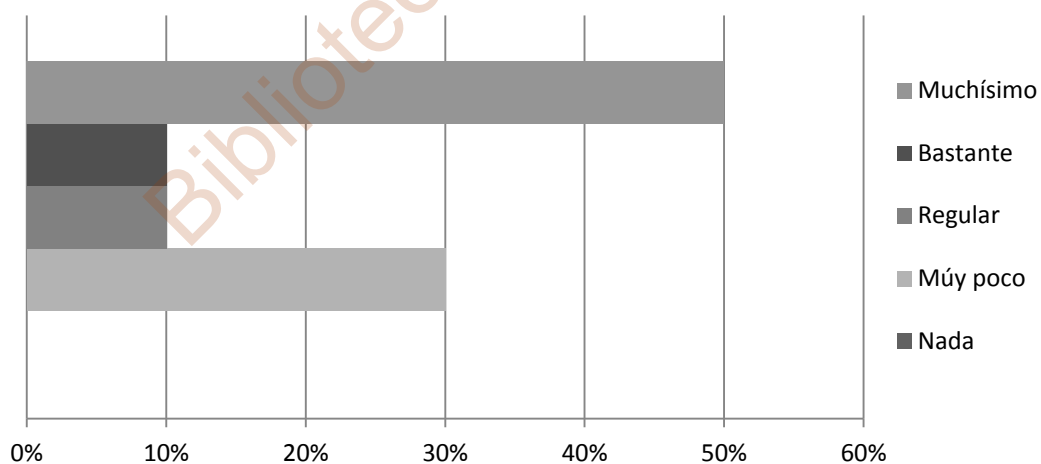
A continuación se muestran unas gráficas que incluyen el porcentaje de los resultados de la encuesta. Como se puede apreciar los resultados fueron favorables, y los participantes tuvieron en su mayoría una opinión positiva sobre la aplicación.

La mayor discrepancia en la encuesta fue hallada en la pregunta dos: la mitad de los encuestados tenía mucha experiencia utilizando aplicaciones móviles con teléfonos inteligentes, mientras que el 30% tenía poca experiencia manejando esta tecnología.

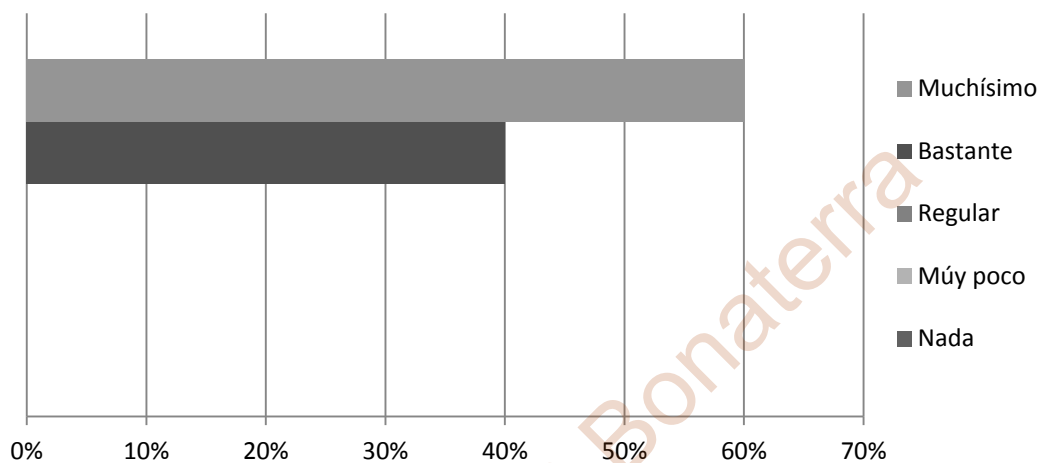
1. ¿Qué tan innovadora consideras que es esta aplicación?



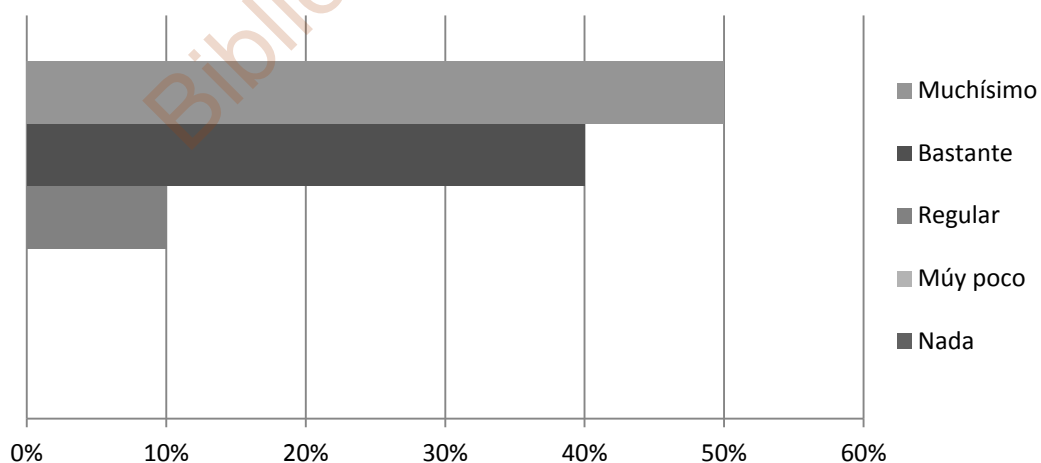
2. ¿Con que frecuencia utilizas aplicaciones similares para teléfonos Smartphone?



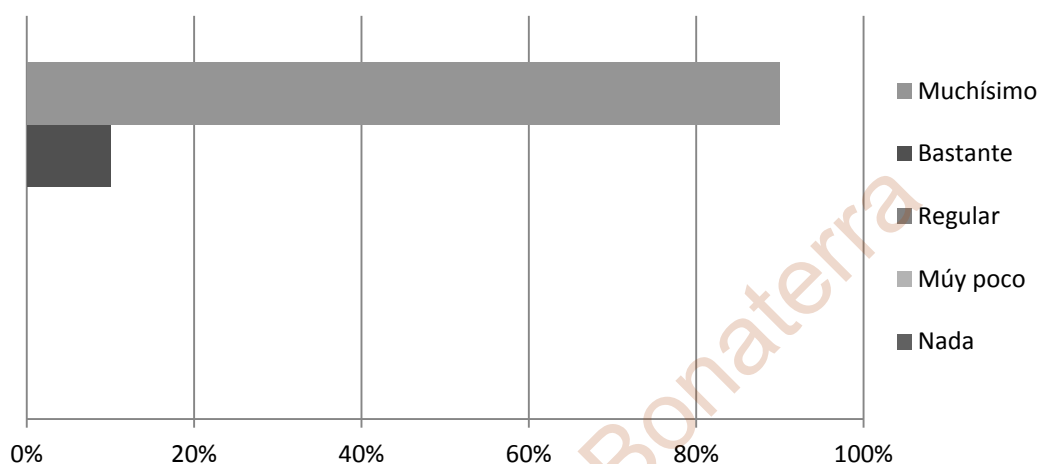
3. ¿Qué tan práctico te resultaría adquirir esta aplicación? ¿Cumple con su función?



4. ¿Si este producto estuviera en el mercado a un precio accesible, que probabilidades hay de que lo adquirirías para tu teléfono?



5. ¿Consideras que esta aplicación le podría servir a alguien más? ¿Cuánto la recomendarías?



6. En tus propias palabras ¿Que es lo que más te agrado de esta aplicación?

Participante	Edad	Respuesta
Alejandro	10	Es sencilla. No hay que hacer mucho esfuerzo.
Abelardo	15	Comodidad para cualquier usuario. Te puedes guiar con las opciones de voz, y te va indicando por medio de la voz las distintas funciones de la aplicación.
Francisco	16	La forma de desplazarse del dedo y la voz.
Iván	37	Los gestos táctiles. Que la aplicación te indique cómo salir o retroceder (las instrucciones).

7. En tus propias palabras ¿Que mejoras crees que beneficiarían a esta aplicación?

Participante	Edad	Respuesta
Alejandro	10	Mas voces. Antes de entrar al menú/aplicación, que la voz indique que hacer.
Abelardo	15	Función de radio. Función para grabar la radio, con opción de pausar la grabación y continuar (para omitir los anuncios). Resumir o acortar los comandos para mayor velocidad. Opción de acelerar la voz. Hacer la voz más humana, menos robótica.
Francisco	16	Más idiomas. Más acentos.
Iván	37	Que funcione al 100% sin necesidad de una conexión. Que especifique la opción seleccionada, por ejemplo: "Menú principal, encabezado."

Creo que la retroalimentación que obtuve durante la realización de esta encuesta será de gran ayuda a la hora de mejorar la aplicación. Algunas de estas ideas serán tomadas en cuenta e incluidas en el siguiente punto de este documento.

5. Conclusiones y trabajo futuro

Se aprovecharon las ventajas del IDE para Android al desarrollar esta aplicación. El sistema de desarrollo mediante el uso de Actividades da una gran ventaja al programador a la hora de desarrollar para dispositivos móviles.

En general, el rendimiento de la aplicación con un dispositivo de gama media sobrepasó mis expectativas. Al momento de realizar pruebas esperaba que hubiera cierto retraso al procesar la información, o pasar a una ventana nueva. Sin embargo, no se mostró ningún tipo de retraso no anticipado durante el manejo de la aplicación.

De la misma forma, el motor de texto a voz integrado en el framework funcionó mejor de lo esperado: los archivos de audio los leía de forma clara y correcta.

Una de las desventajas más aparentes del reproductor de audio fue la demora a causa de la carga de datos específicos. Por ejemplo, el motor de texto a voz utiliza por defecto el acento en inglés de E.U.A. Al dejar el idioma por defecto, no existe retraso alguno, ya que la información viene precargada. Sin embargo, cuando se desea utilizar un acento de otro idioma, en este caso el español, hay una demora considerable a la hora de crear la actividad o pantalla. Se intentaron varios métodos recomendados, como utilizar distintos threads y dejar que el acento en español para el texto a voz se cargue en un thread distinto, sin embargo esto solo causaba que el audio no respondiera por un par de segundos tras cargar la actividad.

Al final se optó por utilizar un loader, es decir, la aplicación dará instrucciones al usuario de esperar a que cargue la aplicación: el programa indica al usuario que debe de esperar a que el contenido termine de cargarse, aunque esto solo ocurre si el usuario presiona la

pantalla antes de tiempo. Este problema solo afecta la pantalla "Audio File List", ya que es la única que utiliza la herramienta texto a voz por el momento. Las otras pantallas utilizan archivos mp3 para las instrucciones. A futuro se pretende investigar a mayor detalle este problema, e intentar buscar una alternativa más viable y eficiente.

5.1 Conclusiones

Según vaya incrementando el desarrollo de aplicaciones dirigidas a personas con discapacidad, habrá mayor variedad, al igual que más oportunidades de mejora. Creo que el desarrollo de esta tesis ha contribuido en la creación de una alternativa al tipo de menús interactivos en los dispositivos de pantalla táctil.

De momento no existen aplicaciones con una interfaz que emplee un cuadro de opciones cuya posición varíe dependiendo del input del usuario, e incluya esta forma sencilla de seleccionar las distintas opciones de la lista. Se trata de una aplicación de reproducción de archivos de audio que es única en su diseño.

La versión 2.0 además aportó una mejora al proyecto en cuanto a rendimiento de la interfaz. El uso de layouts dentro de la pantalla, en vez de objetos con archivos de imágenes y eventos programados, ayudó a que no existiera ninguna demora al crear las distintas actividades y cargar sus elementos. Con excepción de la lista de archivos de audio. Esto resulta muy conveniente en dispositivos antiguos o de gama media y baja.

El propósito de esta tesis fue el crear una aplicación que no requiriera del uso de la vista para manipularla, y fue posible gracias a este sistema de layouts dinámicos.

Considero que esta tesis puede llevar a la creación de otros proyectos o apps que implementen este tipo de sistema o interfaz, lo cual pueda ser de ayuda a las personas

invidentes. Creo que la mayor aportación que esta aplicación brindaría a la sociedad es el proporcionar una herramienta novedosa que permitiría a las personas invidentes relajarse y escuchar música, documentos como audio libros, y posiblemente estaciones de radio. Es verdad que ya existen en las tiendas de Android, Apple y Microsoft numerosas aplicaciones dirigidas a invidentes tal como la que se presenta en esta tesis, sin embargo no hay un repertorio tan extenso como creo que debería haber. Al menos para dispositivos móviles. Creo que es importante que más productos de este tipo sean publicados en las tiendas móviles, ya que esto concientizaría más a las empresas y desarrolladores independientes, lo cual los motivaría a realizar productos con interfaces especiales para invidentes y débiles visuales.

Expandir el volumen y la popularidad de aplicaciones para invidentes y débiles visuales podría animar a aquellos que no se sientan cómodos utilizando estos programas, o que simplemente no les llame la atención, a integrarse en el uso de la tecnología móvil. De esta forma se acercarían más a la sociedad, en vez de ser alienados de la creciente tecnología digital que conecta a las personas.

5.2 Costos del proyecto

Se estima que completar este programa costaría aproximadamente unos 120,000 pesos. A pesar de que parte del código de la aplicación ya ha sido desarrollado, aun se encuentra en fase de prototipo. Para poder ser lanzada al mercado requiere que se implementen los cambios que serán mencionados en la sección de Trabajo Posterior. Los costos desglosados se presentan a continuación:

- **Plataforma:** Android. Posiblemente se emplearan librerías o herramientas como AndEngine, que no generará costo adicional aparte de la documentación o cursos para su manejo. Sin costo estimado.
- **Hardware:** Laptop o PC de escritorio armada de media/alta gama. Equipos móviles estándar para realizar pruebas. 25 000 pesos.
- **Desarrolladores:** 1 desarrollador. 65 000 pesos.
- **Outsourcing adicional:** Especialistas en actuación de voz, y/o diseñador gráfico. Asesores. Testers para las pruebas Beta. 30 000 pesos.
- **Tiempo de desarrollo:** De 3 a 5 meses. Incluye las fases de desarrollo y pruebas.

Es difícil decidir cómo y cuánto cobrar por la aplicación, y es un tema que requiere de una investigación muy profunda además de un estudio de mercado. De momento una gran cantidad de las aplicaciones que se encuentran en la tienda de Android son gratuitas. Esto se debe a que tienen diversas formas de cobrar para recuperar (lentamente) la inversión que representa los costos de desarrollo de la aplicación. Existen diversos métodos para cobrar a los usuarios una vez hayan descargado la aplicación gratis. Generalmente son los siguientes:

- Adquisición de cuenta o versión *Premium* de la app.
- Micro transacciones.

- Publicidad/anuncios.
- Costo por instalación (contratación de motores para publicitar la app).
- Patrocinios.

Para la creación de esta aplicación la mejor forma de hacerlo es distribuirla de forma gratuita, ya que generalmente esto multiplica la cantidad de descargas. Aún cuando la aplicación cuesta 15 pesos, por poner un ejemplo, si no es gratuita esto puede reducir considerablemente el número de usuarios.

Utilizaré una mezcla de los métodos mencionados previamente para cobrar por la aplicación y recuperar la inversión, con excepción del uso de los anuncios. Creo que en este caso existirían complicaciones si estos se añadieran: la mayor parte de las publicidades son *popups* o imágenes que el usuario al que va dirigido este programa no podría leer. Dado el caso de que la publicidad fuera por audio, esto estorbaría enormemente en el manejo de la aplicación y podría frustrar al usuario. Lo último que necesita cualquier aplicación, sobre todo una con un mercado relativamente menor, son malas puntuaciones y comentarios cuya consecuencia sería la pérdida de usuarios potenciales.

5.3 Trabajo Futuro

Como posibles mejoras al programa, aparte de resolver o reducir los problemas mencionados, también pretendo mejorar el diseño del layout en general. Un ejemplo sería agregar una opción dentro del menú de configuración para cambiar el tono de los colores, y emplear fuentes y gráficos amigables y de buen contraste para poder ser identificados con mayor facilidad.

De momento esta aplicación cuenta con numerosos bugs, como la sobre posición de instrucciones de audio en algunas pantallas. Esto hace que el audio sea ininteligible, por lo que arreglar este y otros bugs a futuro sería una de las mayores prioridades en el trabajo posterior de esta aplicación.

El reproductor de audio en su estado actual cuenta con una funcionalidad esencial. En una versión posterior de la aplicación pretendo agregar más opciones, como reproducción de canciones aleatorias, o la posibilidad de avanzar o retroceder dentro de una pista de audio, posiblemente con una barra estilo termómetro, o simplemente manteniendo presionados los botones "Anterior" y "Siguiente", aunque este último presentaría una complejidad elevada en esta aplicación en particular.

A futuro el diseño de la interfaz de audio será más complejo que en su estado actual. De momento las instrucciones de audio de la interfaz se conforman de varios archivos de audio que son compilados junto con el resto del programa como material *raw*. Más adelante las instrucciones por audio se podrían ejecutar mediante el motor texto a voz. Esto ayudará a que el programa sea más ligero. También ayudará a simplificar el traducir la aplicación a más idiomas, una vez se consiga remediar el problema de demora al no emplear el acento precargado.

Si no hay conexión de red, el texto a voz en español con acento de México no funciona, aunque se haya descargado ya el idioma seleccionado. El *TTS* en inglés no requiere de una conexión. La idea es que la app no necesite una conexión para funcionar correctamente (salvo cuando se lleguen a emplear los servicios Cloud de la aplicación). Se requiere una conexión a internet para ciertos idiomas o acentos. Éste fue un problema que pude resolver utilizando el acento de España, aunque me gustaría hallar la causa de este comportamiento. Esto podría servir para incluir una variedad de acentos además de los idiomas. Así los usuarios podrían elegir el acento de acuerdo a su región, si así lo desean.

El idioma de la app se configurará de acuerdo al idioma por defecto del dispositivo, la primera vez que se utilice tras su instalación.

Para hacer la aplicación más amigable, creo que sería buena idea mejorar voces, contratar voces en off profesionales. Esto sería dado el caso que a futuro se siga alternando el audio entre archivos *Raw* y *TTS*.

Tras realizar una encuesta a personas invidentes de Aguascalientes, creo que algunas de las ideas que aportaron podrían mejorar esta aplicación considerablemente. Por ejemplo, la posibilidad de incluir una función de radio con varias opciones prácticas y cómodas para el usuario.

Creo que ayudaría mucho al usuario a comprender el funcionamiento de la interfaz en su primer uso si se incluyera una guía o tutorial: Como el diseño de la aplicación es poco común, creo que esto sería indispensable. Si a los participantes de la encuesta no les hubiera explicado cómo utilizar la aplicación antes de probarla, habría sido un poco difícil que la pudieran manejar correctamente sin antes perder tiempo aprendiendo a prueba y error. Esto se debe a que están acostumbrados a hacer click sobre la opción que desean elegir, en vez de soltar el dedo para seleccionar dicha opción como es el caso de esta app. La guía o tutorial se activará automáticamente tras arrancar la aplicación por primera vez.

También creo que sería de gran ayuda para el usuario que se especificaran mejor cada una de las opciones: por ejemplo que se indique que el primer botón del cuadro es un encabezado. De esta forma el usuario sabrá que seleccionar esta opción no activará ningún evento. En otros casos sería conveniente hacer los comandos mas resumidos, para evitar redundancia y acelerar el paso de la interacción. En esencia, hacer instrucciones de audio más claras y específicas.

Los usuarios invidentes utilizan aplicaciones con instrucciones por audio similares a las de esta aplicación, como "NDVA". Sin embargo, la velocidad de la voz es distinta que en mi

aplicación. Las personas invidentes que manejan estas tecnologías están acostumbradas a escuchar los comandos a una velocidad mucho mayor, por lo cual creo que sería de gran ayuda poner una opción para regular la velocidad de las instrucciones.

Adicionalmente se planea explorar la posibilidad de agregar la opción de reproducir audio almacenado en la nube. Este es un mercado en crecimiento, e implementar esta característica a la aplicación le daría variedad, y el usuario podría escuchar noticias, música o audio libros con el uso de una conexión a internet, sin tener que preocuparse por transferir los datos manualmente al dispositivo desde otra máquina. Soundcloud es un servicio web que utiliza este tipo de sistema para escuchar archivos almacenados en la nube.

Otra mejora sería incorporar el uso de gestos para ciertas funciones, como slide a la derecha/izquierda para cambiar de página en la lista de canciones. Este tipo de gestos existen en otras aplicaciones: la herramienta de accesibilidad incorporada a los iPhones incluye la opción de seleccionar los distintos botones de un menú emergente o contextual con los gestos slide a la derecha para seleccionar la siguiente opción, y slide anterior para la opción previa.

Para ampliar el mercado y las posibilidades de que esta aplicación pueda llegar a más personas, una vez exista una versión estable y que esté siendo distribuida en la tienda de Google, creo que sería viable desarrollar una versión para iPhone.

6. Referencias

Meier, Reto, "Professional Android 4 Application Development", John Wiley & Sons, Inc., 2012.

Zechner, Mario, y Green, Robert, "Beginning Android Games Second Edition", Apress, 2012.

Phillips, Bill, y Hardy, Brian, "Android Programming: The Big Nerd Ranch Guide", Big Nerd Ranch, Inc. 2013.

Schroeder, Jayme, y Broyles, Brian, "AndEngine for Android Game Development Cookbook", Packt Publishing, 2013.

Mehta, Nirav "Mobile Web Development" Packt Publishing Ltd., 2008.

Ling, Richard, "The Mobile Connection : The Cell Phone's Impact on Society", Interactive Technologies, Morgan Kaufmann, Junio, 2004.

Saini, J.S. Goyal, Amit, "Empowering the Disabled", Abhisek Publications, 2009.

Webster, Alec, Roe, Joao, "Children with Visual Impairment : Social Interaction, Language, and Learning in Mainstream School", Routledge, 1997.

Cattaneo, Zaira, Vecchi, Tomaso, " Blind Vision : The Neuroscience of Visual Impairment", MIT Press, 2011.

Stuber, Gordon L., "Principles of Mobile Communication, Second Edition", Kluwer Academic Publishers, 2000.

<http://www.inegi.org.mx/>

<http://developer.android.com/>

<http://andrewmitchelldesign.co.uk/>

<http://www.engadget.com>

Biblioteca UP Bonaterra