

Universidad de las Ciencias Informáticas

Facultad 5



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

**Aplicación Android para reproducir
audio en la plataforma SmartAP
mediante bluetooth.**

Autora: Martha García Montero

Tutores: Ing. Yadira Ramírez Rodríguez

Ing. Luis Manuel Vidal Piña

La Habana, junio de 2016

Declaración de autoría

Declaro que soy la única autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas con los derechos patrimoniales de la misma, con carácter exclusivo. Autorizo a dicho centro para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del autor

Martha García Montero

Firma del tutor

Ing. Luis Manuel Vidal Piña

Firma del tutor

Ing. Yadira Ramírez Rodríguez

Dedicatoria

A mi mamá por ser la persona más importante en mi vida.

A mi hermanito Ale, eres mi niño pequeño, te quiero mucho.

A mis abuelos Alvaro y Dulce que sé que en el lugar que estén me están mirando y cuidando siempre.

Agradecimientos

A mi mamá por ser madre, amiga y compañera. Por ser el motor impulsos de todos mis planes, por apoyarme siempre en los momentos más difíciles desde mi infancia hasta ahora. Por ver luz donde yo veía oscuridad. Por soportar a una niña que no acaba de crecer. Por ser tú.

A mi padre por ser mi padre, por su personalidad, por el amor silencioso que tiene hacia sus hijos, por enseñarme cosas que no se aprenden en una escuela y por su comida.

A mi familia que no es solo la de sangre, sino a todas aquellas personas que son parte de mi vida desde el día en que nací y fueron definiendo la forma de ver lo que es: la familia, los amigos, mi carácter y mi vida. Incluyo a aquellos que ya no están y que se recuerdan a diario.

A mi tía Mary, fuiste el relevo de una persona increíble y te convertiste en lo mismo. te quiero muchísimo.

A Ade, no conozco una gallega dicharachera más maravillosa, resabiosa y cariñosa que ella, gracias por ser mi abuela.

Hay una fase que siempre llevo en mi pensamiento: “Amigos más que hermanos, porque los amigos se escogen los hermanos no”, yo adoro a mis hermanos, pero los amigos son un pilan fundamental en mi vida.

A Ivon por ser mi hermana desde antes de saber caminar.

A mis amigos de infancia que duran para siempre, aunque la distancia sea cada día mayor: Adrián, Kiri y el resto del piquete.

A mis amigos del IPI, personas muy difíciles de olvidar y de nombrarlas a todas.

A Rosy y Susana que fueron las primeras personas que me apoyaron cuando llegue a la UCI y que hoy siguen hay. Dando trabajo, pero hay.

A mi grupo 5106 Yeili, Dairon, Riqui.

A cuatro chicas muy especiales que desde que entraron en mi vida provocaron algunos cambios algunos para bien y otros no tanto: Daymi, Amanda, Liena y Gretel.

Dicen que repetir es lo peor que hay, pero viéndolo ahora si no lo hubiese hecho no tendría en mi vida a personas como Hector, Yosvani y Andrea con la cual me encanto trabajar durante mucho tiempo pues siempre te da un buen consejo, aunque no sea el que quieras hoy.

A mis chicos del 5504 en especial a mis secuaces: Henry, José Miguel, Yei y Ofelia, gracias por apoyarme en momentos complicados.

Agradecimientos

A mi tutora, por ser más que mi tutora, por alarme las orejas cuando hacía falta, gracias por ser mi profe, por escucharme, por ser mi amiga y por permitirme ser parte de tu pequeña familia.

A mi tutor, que puedo decir de ti, un compañero cariñoso y amable, un tutor insoportable, un hombre increíble, bello por dentro y por fuera. Bb TQM.

Resumen

En el mundo de hoy el desarrollo de nuevas tecnologías para los dispositivos móviles es un mercado que está en pleno auge. Dentro de ellas se encuentran las aplicaciones para el sistema operativo Android, y los microcontroladores con software Arduino, muy utilizado en el campo de la robótica.

La Universidad de las Ciencias Informáticas no queda fuera en el desarrollo de las nuevas tecnologías. Es por esto que en la facultad 5 el centro de informática industrial(CEDIN) cuenta con la línea de desarrollo sistemas embebidos en la cual se desarrolló una plataforma denominada SmartAP para la reproducción de audio, que utiliza entre sus componentes un microcontrolador Arduino, pero su uso se limita al control manual del sonido. Sus funcionalidades solo permiten encender la plataforma, leer la lista de forma cíclica y solo saltar al siguiente archivo. Tiene entre sus componentes un periférico *bluetooth* de baja energía (*BLE* por sus siglas en inglés) que permite la conexión por *bluetooth* con cualquier dispositivo móvil que lo contenga. Es por esta razón que es necesaria desarrollar una aplicación que permita esta conexión entre la plataforma y algún dispositivo compatible con el periférico, con el objetivo de controlar la reproducción de audio de la plataforma.

Este trabajo se basa en desarrollar una aplicación para sistema operativo Android 4.3+, con el propósito de solucionar las limitaciones de la misma. Utilizando diferentes tecnologías como el ambiente de desarrollo integrado Android Studio, el lenguaje de programación Java, la herramienta digital Adobe Photoshop y sustentado por la metodología ágil XP.

Palabras Clave: aplicación Android, Arduino, *bluetooth* de baja energía, plataforma SmartAP, reproducir audio.

Contenido

Declaración de autoría	1
Dedicatoria	2
Agradecimientos	3
Resumen	5
Introducción	1
Capítulo 1 Fundamentación teórica	5
Introducción	5
1.1 Sistema operativo Android	5
1.1.1 Ventajas y desventajas de Android	5
1.2 Protocolo <i>bluetooth</i>	6
1.2.1 Funcionamiento	6
1.2.2 Estándares	6
1.3 Plataforma SmartAP	7
1.4 Plataforma Arduino	8
1.4.1 Características de Arduino	9
1.5 Análisis de soluciones existentes	9
1.6 Entorno de desarrollo	11
1.6.1 Android Studio	11
1.6.2 ADT de Eclipse	11
1.6.3 Arduino Software	12
1.7 Lenguajes de programación	13
1.8 Herramienta de diseño gráfico Adobe Photoshop	14
1.9 Metodología de desarrollo	15
Consideraciones parciales	16
Capítulo 2 Planificación y Diseño	17
Introducción	17
2.1 Fase I: Planificación	17
2.1.1 Las historias de usuario	17
2.1.2 Requisitos no Funcionales del sistema	23
2.1.3 Estimación de esfuerzos.	24
2.1.4 Iteraciones	25
2.1.5 Plan de entregas	27
	VI

2.2	Fase II: Diseño	27
2.2.1	Representación arquitectónica de la solución	27
2.2.2	Patrones de Diseño GRASP	34
2.2.3	Tarjetas CRC (Clases-Responsabilidad-Colaboración)	35
	Consideraciones parciales	38
Capítulo 3 Desarrollo y Pruebas		39
	Introducción	39
3.1	Fase III: Desarrollo	39
3.1.1	Tareas de ingeniería	39
3.1.2	Estilos de codificación	41
3.2	Fase IV: Pruebas	42
3.2.1	Pruebas de aceptación	42
3.2.2	Pruebas de aceptación de la primera iteración	44
3.2.3	Pruebas de aceptación de la segunda iteración	44
3.2.4	Pruebas de aceptación de la tercera iteración	45
3.2.5	Análisis de los resultados obtenidos	45
	Consideraciones parciales	45
Conclusiones generales		47
Recomendaciones		48
Referencia Bibliográfica		49
Glosario de Términos		52
Anexos 1 Tareas de ingeniería		56
Anexo 2 Pruebas de aceptación		63

Introducción

En los últimos años los dispositivos móviles han pasado a ser una parte esencial de la vida diaria de los seres humanos. Las grandes compañías han apostado fuerte en el desarrollo de estos dispositivos y esto ha significado un aumento exponencial de sus características, técnicas y repositorios de aplicaciones (apps).

Entre los dispositivos móviles destacan los denominados teléfonos inteligentes (*smartphone* del inglés *smart*: inteligente y *phone*: teléfono). Estos ya no son solo aparatos de comunicación, sino pequeños ordenadores de bolsillo con los que se pueden ir a todas partes y en todo momento. Lo mismo pasa con las tabletas (*tablets*, término de la lengua inglesa, para definir una computadora portátil más grande que un *smartphone*), que aunque, no de manera tan arrolladora como los anteriores, han entrado en el mercado de los dispositivos inteligentes con una fuerte presencia desde el año 2010.

Uno de los sistemas operativos más usados dentro de los dispositivos móviles es Android. Propiedad de Google y basado en un núcleo de GNU/Linux, en conjunto con la variedad de apps disponibles en Google Play Store son los principales motivos de su éxito. Además, permite la creación de aplicaciones de una manera muy sencilla mediante un ambiente de desarrollo integrado (IDE del inglés *Integrated Development Environment*) basado en Java, como es Android Studio, junto al SDK Manager de Android.

También está en plena expansión el uso de microcontroladores desde el año 2012. La plataforma de hardware libre Arduino es su mayor exponente y está abierta para la elaboración de prototipos, engloba la parte de hardware (con una placa y un microcontrolador) y software (con un entorno de desarrollo propio). La utilización de estos dispositivos es una buena manera de iniciarse en el mundo de la electrónica y la programación, gracias a su sencillez. Así quedan atrás los tiempos en que programar estos tipos de dispositivos solo se podía hacer con lenguajes de bajo nivel como por ejemplo el ensamblador o similares.

La gran acogida que han recibido las plataformas de hardware libre Arduino en tan poco tiempo, ha hecho que aparezcan una gran variedad de dispositivos, y con ellos muchas posibilidades de hacer nuevos prototipos. Para la creación de los mismos existen multitud de sensores y componentes electrónicos que permiten hacer proyectos sobre cualquier ámbito, cada día aparecen nuevos componentes a utilizar. Este es un campo en plena

expansión que permitirá implementar muchas soluciones para el Internet de las Cosas (IoT) (1).

La Universidad de las Ciencias Informáticas no queda fuera en el desarrollo de las nuevas tecnologías. Es por esto que en la línea desarrollo sistemas embebidos del centro de Informática Industrial (CEDIN) perteneciente a la facultad 5 se construyó una plataforma de hardware libre basada en Arduino para el desarrollo de reproductores de audio denominada SmartAP, que permite reproducir diferentes formatos de audio como son: MP3, WMA, AAC, entre otros.

La línea sistemas embebidos se encarga del diseño de diagramas eléctricos e implementación de los *firmwares* necesarios para el desarrollo de diferentes dispositivos. Actualmente desarrolla aplicaciones de supervisión y control para los dispositivos de bajas prestaciones. Las investigaciones están centradas en las herramientas y aplicaciones embebidas de los PLC, los microcontroladores es AVR de ATMEL y la plataforma Arduino.

La plataforma SmartAP (en inglés, Smart Audio Player) contiene un *firmware* que presenta un grupo de limitantes, teniendo en cuenta que hasta el momento solo es operable a través de la interfaz ubicada en el dispositivo de forma manual. Además, carece de un dispositivo que ofrezca la información asociada al proceso de reproducción de las pistas de audio entre las que se encuentran: el nombre, el tiempo de duración y la cantidad de pistas que contiene la lista de reproducción. Los archivos de audio se almacenan en la *microSD*, conectada a la plataforma. Por otra parte, la navegación en la lista de reproducción solo es hacia adelante y de forma cíclica. Esto trae consigo la pérdida de tiempo por parte del usuario ya que la plataforma no cumple con conceptos adecuados de usabilidad. No presenta las funciones básicas que son comunes en los reproductores de audio actuales como son: ver la lista de reproducción, seleccionar de una pista en específico además de poder saltar de una pista a otra de diferentes formas (repetida, secuencial o aleatorias).

De la situación problemática antes expuesta se deriva el siguiente **problema de investigación** a resolver: ¿Cómo mejorar la reproducción de audio de la plataforma SmartAP?

La investigación tiene como **objeto de estudio** las aplicaciones para reproducir audio mediante *bluetooth*, constituyendo el **campo de acción** las aplicaciones Android para reproducir audio mediante *bluetooth* en la plataforma SmartAP.

Para darle solución al problema de investigación se propone como **objetivo general** desarrollar una aplicación Android para la reproducción de audio en la plataforma SmartAP.

Como **idea a defender** se plantea que con la aplicación Android para la reproducción de audio en la plataforma SmartAP mediante *bluetooth*, mejorará la accesibilidad del usuario a la plataforma, así como su competitividad con otros dispositivos de reproducción de audio.

Del objetivo general se desglosan las siguientes **tareas de la investigación**:

- Revisión bibliográfica para desarrollar un marco teórico de acuerdo a las tendencias actuales de las aplicaciones para la reproducción de audio de forma remota.
- Definición de los requisitos funcionales y no funcionales para el desarrollo de la aplicación.
- Diseño de la aplicación a desarrollar según los requisitos encontrados.
- Implementación de las funcionalidades del sistema propuesto.
- Realización de pruebas de aceptación a la aplicación.

La investigación estará estructurada por: introducción general, tres capítulos, conclusiones, recomendaciones, referencia bibliográfica, glosario de términos y anexos.

- **Capítulo 1: Fundamentación teórica.**

En este capítulo se hace referencia a los componentes principales para el funcionamiento de la plataforma SmartAP y en qué consiste. Se hace una comparación entre aplicaciones que existen para la reproducción de audio en diferentes tipos de dispositivos (PC, tabletas, teléfonos inteligentes). Además, se caracterizan los lenguajes de programación a utilizar para desarrollar tanto en Android(Java) como en Arduino(C++). Se describe la metodología de desarrollo de software a utilizar.

- **Capítulo 2 Planificación y Diseño**

Se realiza una valoración de las primeras dos fases del ciclo de vida de la metodología XP: planificación y diseño. Se establecen las historias de usuario, las cuales servirán para un mejor entendimiento y conocimiento del software y representan las principales necesidades del sistema. Se planifica el proceso de desarrollo, se expone el diseño del sistema y la representación arquitectónica del mismo.

- **Capítulo 3 Desarrollo y Pruebas:**

Se exponen las principales características de las dos últimas fases del ciclo de vida de la metodología XP: desarrollo y pruebas. Se procede a desarrollar las tareas de la ingeniería que responden a las historias de usuario (HU) abordadas en cada iteración. Luego mediante las pruebas de aceptación se verifica que el producto resultante cumpla con los requerimientos definidos.

Capítulo 1 Fundamentación teórica

Introducción

En este capítulo se hace referencia a los componentes principales para el funcionamiento de la plataforma SmartAP y en qué consiste. Se hace una comparación entre aplicaciones que existen para la reproducción de audio en diferentes tipos de dispositivos (PC, tabletas, teléfonos inteligentes). Además se caracterizan los lenguajes de programación a utilizar para desarrollar tanto en Android(Java) como en Arduino(C++). Se describe la metodología de desarrollo de software a utilizar.

1.1 Sistema operativo Android

Android es un sistema operativo para dispositivos móviles basado en Linux, en conjunto con aplicaciones *middleware* está enfocado para ser utilizado en teléfonos inteligentes, tabletas y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. El sistema operativo está compuesto por más 12 millones de líneas de código aproximadamente, incluyendo XML, C, Java y C++ (2).

Desde abril de 2009, las versiones de Android han sido desarrolladas bajo un nombre en clave y siguen un orden alfabético: Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice CreamSandwich, JellyBean, KitKat, Lollipop y Marshmallow. Para dar solución a las mejoras que los dispositivos requerían, tanto de hardware (pantallas más grandes, desaparecían los teclados), como de software (nuevas funcionalidades y requerimientos de los dispositivos).

A partir de la versión Android 4.3 JellyBean fue que se comenzó a dar soporte para *bluetooth 4.0 (Bluetooth Smart o BLE)*, es por esto que la aplicación a desarrollar es para esta versión o superiores.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK por sus siglas en inglés» provisto por Google el cual se puede descargar gratuitamente desde el sitio oficial de Android. Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos (3).

1.1.1 Ventajas y desventajas de Android

Entre las ventajas más notables del sistema operativo Android se encuentran (4):

Capítulo 1 Fundamentación teórica

- Da completa libertad al propietario de un terminal a instalar cualquier aplicación en el formato APK.
- Google liberó la mayoría de su código bajo la licencia apache, una licencia libre y de código abierto.
- Cuenta con una de las comunidades más grande de desarrolladores.
- Dos tercios de sus aplicaciones son gratuitas y disponibles en su tienda oficial de aplicaciones: Google Play.
- Es completamente personalizable.
- Posee sistema de multitarea inteligente.

Entre sus desventajas se pueden mencionar (4):

- El acceso a las actualizaciones de cada versión, depende del fabricante.
- La mayoría de software malicioso creados para Linux están destinados a dispositivos con este sistema operativo.

Los dispositivos con sistema operativo Android tienen entre sus variantes de establecer conexión con otros dispositivos el *bluetooth*.

1.2 Protocolo *bluetooth*

Bluetooth es un protocolo de comunicación basado en redes inalámbricas de área personal (WPAN), el cual permite la transferencia de datos entre 2 o más dispositivos por medio de un enlace de radiofrecuencia de 2.4 GHz (5).

1.2.1 Funcionamiento

Cada dispositivo deberá estar equipado con un *microchip*, que transmite y recibe en la frecuencia de 2.4 GHz y tiene una dirección única de 48 bits basado en el estándar IEEE 802.15.1 (6).

Las especificaciones principales del *bluetooth* son (6):

- Potencia del transmisor: entre 1 y 100 mW, típica de 2,5 miliwatios.
- Canales máximos: hasta 3 de voz y 7 de datos.
- Velocidad de datos: hasta 720 kbit/s.
- Rango esperado del sistema: hasta 100 metros.
- Interferencia: Bluetooth minimiza la interferencia potencial al emplear saltos rápidos en Frecuencia =1.600 veces por segundo.

1.2.2 Estándares

Capítulo 1 Fundamentación teórica

Todas las versiones de los estándares de *bluetooth* están diseñadas para permitir que el último estándar cubra todas las versiones anteriores.

Bluetooth v1.0

Fue la primera versión creada. Los dispositivos tenían problemas comunicándose entre sí y al conectarse perdían el anonimato. La versión 1.1 solucionó varios problemas de la 1.0 y se convirtió en el primer *bluetooth* funcional. Con la versión 1.2 aumentó el caudal de datos a 1 Mbit/s y se cambiaron las frecuencias de radio para eliminar interferencias.

Bluetooth v2.0

La versión 2.0 tiene mejor funcionalidad a la hora de la unión con varios dispositivos y mejora el consumo de energía. Al salir la versión 2.1 mejoro la conexión bajando el número de procesos durante el proceso de unión y redujo aún más el consumo de energía.

Bluetooth v3.0

La versión 3.0 ofrece transmisión de alta velocidad de hasta 24Mbit/s. Usa el estándar 802.15 *bluetooth* radio, pero también incluye funcionalidades 802.11 para mayores transferencias de archivos y una velocidad mejorada.

Bluetooth v4.0

La versión 4.0, que incluye la alta velocidad y protocolos del *bluetooth* de bajo consumo. Se basa en wifi y consta de protocolos *bluetooth* legados.

Bluetooth de baja energía (BLE) es un subconjunto de Bluetooth v4.0 con una pila de protocolo completamente nueva para la rápida acumulación de enlaces sencillos. Como alternativa a los protocolos estándar que se introdujeron en *bluetooth* v1.0. Está dirigido a aplicaciones de muy baja energía. Los diferentes diseños de *chips* permiten dos tipos de implementación: modo dual y modo único.

La plataforma SmartAP también posee un periférico de comunicación bluetooth de baja energía que le permite intercambiar información con dispositivos compatibles.

1.3 Plataforma SmartAP

SmartAP (en inglés, Smart Audio Player) es una plataforma de hardware basada en Arduino que permite reproducir varios formatos de audio. La plataforma hace uso de un dispositivo Arduino Micro, un módulo que permite reproducir archivos de audio en varios formatos y funciona además con un periférico BLE, para el intercambio de información por este tipo de interfaz de comunicación (7).

La plataforma SmartAP está compuesta por tres elementos fundamentales (7):

Capítulo 1 Fundamentación teórica

- Placa *Adafruit nRF8001 Bluetooth Low Energy* (BLE), para establecer comunicación vía BLE con cualquier dispositivo compatible con este tipo de interfaz (iOS, Android4.3+). BLE es una versión ligera de *bluetooth* 4.0 diseñada especialmente para disminuir el consumo de energía.
- Placa *Adafruit VS1053 MP3/AAC/Ogg/MIDI/WAV Codec*, que permite decodificar una amplia variedad de formatos de audio, entre los que se destacan: MP3, AAC, OggVorbis, WMA, MIDI, FLAC y WAV.
- Dispositivo Arduino Micro, es una tarjeta basada en un microcontrolador ATmega32u4.

Además, la plataforma SmartAP utiliza el protocolo SPI (Interfaz de Periféricos en Serie o *Serial Peripheral Interface* por sus siglas en inglés) para la comunicación entre sus módulos. SPI es un protocolo de datos en serie síncrono utilizado por los microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas. También se puede utilizar para la comunicación entre dos microcontroladores (8).

La plataforma SmartAP contiene un *firmware* que presenta limitaciones pues hasta el momento solo es operable desde la interfaz del dispositivo. Al tener una placa BLE, el concepto del mismo permite que la plataforma SmartAP actúe como un periférico al que pueden conectarse otros dispositivos.

El periférico es lo que se conoce como servidor GATT, en este sentido es importante tener en cuenta que las conexiones BLE son exclusivas, lo que significa que la plataforma puede atender solamente una conexión en cada momento. Una vez que se recibe una conexión la plataforma deja de estar disponible hasta que dicha conexión se termine (7).

1.4 Plataforma Arduino

Arduino es una plataforma de desarrollo de computación física (*physical computing*) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para la placa (9).

Para ello se utiliza el lenguaje de programación de Arduino que es una implementación de *Wiring*, una plataforma de computación física que a su vez se basa en *Processing*, un entorno de programación multimedia (10).

Arduino surge en el año 2005, en el instituto de diseño interactivo de Ivrea (Italia) como una herramienta fácil de emplear para prototipado rápido, dirigido a estudiantes sin experiencia

en electrónica y programación, de costo muy económico. Con la premisa de que fuera compatible con todos los sistemas operativos y sin barreras de entrada para los principiantes (11).

A través de los años Arduino ha sido el núcleo de miles de proyectos, que incluye desde el manejo de objetos cotidianos hasta el de instrumentos científicos complejos. Una comunidad mundial de fabricantes: estudiantes, aficionados, artistas, programadores y profesionales, han añadido una increíble cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos (12).

1.4.1 Características de Arduino

Arduino se ha utilizado en diferentes proyectos y aplicaciones. Ofrece algunas ventajas para los profesores, estudiantes y aficionados interesados (12):

- El software de Arduino es fácil de usar para los principiantes, pero lo suficientemente flexible para los usuarios avanzados.
- Todas las placas Arduino y el software son completamente de código abierto.
- Es una herramienta clave para aprender cosas nuevas.
- Asequible: las placas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores.
- Multiplataforma: El software de Arduino (IDE) se ejecuta en los sistemas operativos Windows, Macintosh OSX y Linux.
- El software de Arduino está publicado como herramienta de código abierto, disponible para la extensión por programadores experimentados.
- Los planos de las placas Arduino se publican bajo una licencia de *Creative Commons*.

1.5 Análisis de soluciones existentes

En la actualidad existen muchas aplicaciones para reproducir audio de forma remota en diferentes dispositivos *smartphone*, *tablets*, TV y/o plataformas, desde variados sistemas operativos Windows, Linux, Mac y Android. Mientras en la búsqueda nacional no se ha encontrado ninguna aplicación que este enfocada hacia este objetivo.

A continuación, se hace un resumen de las principales aplicaciones para reproducir audio de forma remota que se utilizan en la actualidad, para ver las diferencias en: sistema operativo al que están dirigidas, interfaz de comunicación y el protocolo que utilizan.

Aplicación	Desplegada en SO	Dirigida a SO	a Interfaz de comunicación	de Privativa	Protocolo
Remoto Samsung Audio	Android 4.4.4	Sistema Giga Sansumg (Android)	<i>bluetooth</i> BLE	Si	SPI
Ubuntu Remote Control	Android 4+	Ubuntu,Mint, Gentoo, entre otros.	Wifi	Si	SSH
Escritorio remoto de Chrome	Mac, Windows	Mac, Windows	Wifi	Si	SSH
AIMP	Windows, Android	Windows, Android	Wifi	Si	SSH
Remote	Apple(iOS)	iOS	Wifi	Si	DACP
Reproductor Arduino TFG-UPV	Android 4.3+	Arduino MEGA 2560	<i>bluetooth</i> BLE	Si	SPI

Tabla # 1 .Comparación de soluciones existentes. Elaboración propia.

Las aplicaciones antes mencionadas son de las más destacadas dentro de algunos de los sistemas operativos más populares para móviles como Windows Phone, iOS o Android, pero ninguna se puede utilizar para reproducir audio desde la plataforma SmartAP, ya que son privativas por lo que no se puede acceder a su código y esto no permite adaptarlas a las necesidades del sistema. La aplicación que más características comparte con la solución a la que se desea arribar es Reproductor Arduino TFG-UPV que está enfocada hacia un microcontrolador Arduino específico, solo se limita a reproducir las pistas de audio quedando por debajo de las expectativas de la aplicación que se quiere desarrollar, que debe incluir eliminar alguna pista, grabar y enviar algún archivo hacia la plataforma SmartAP.

No obstante, al consultar la tabla anterior se logra obtener una guía de aplicaciones como AIMP y Ubuntu Remote Control, encaminada a adoptar buenas prácticas en cuanto a

criterios de usabilidad del diseño de las interfaces de usuario y los íconos más utilizados por los usuarios: reproducir, detener, subir y bajar volumen, pista anterior y a la pista siguiente, seleccionar el tipo de reproducción secuencial/repetir/aleatoria, para lograr una mejor interacción con la aplicación. Se consultaron también algunas aplicaciones de reproducción de audio para el sistema operativo Android como TTPop para conocer sobre la interacción con las listas de audio y como se podrían manipular de forma más eficiente.

1.6 Entorno de desarrollo

Un entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios, así Dev C++, es un IDE para el lenguaje de programación C++; Web DevStudio, lo es en línea para el lenguaje de programación C/C++ (13).

En la actualidad se destacan 2 IDEs para desarrollar Android: Android Studio y el *plugin* ADT (del inglés *Android Developer Tools*) para Eclipse.

1.6.1 Android Studio

Android Studio es un IDE para la plataforma Android. Está disponible para descargar en Windows, Mac OS X y GNU/Linux.

El entorno también dispone de un emulador Android. Cuando la aplicación esté en fase de pruebas, se puede comprobar si el funcionamiento es el esperado. También se puede simular la aplicación en un dispositivo externo (14).

Un IDE de código abierto, con infinidad de ejemplos, tutoriales y recursos que se pueden encontrar en internet, libros o incluso en la misma página web de desarrolladores oficial de Android (3). Es muy fácil programar y diseñar una aplicación y que esta pueda funcionar en diversos *smartphones* de diversas marcas.

1.6.2 ADT de Eclipse

Es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de *plugins*. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. Es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el *plugin* JDt que viene incluido en la distribución estándar del IDE (15).

Ventajas de Android Studio frente al ADT de Eclipse (16):

Capítulo 1 Fundamentación teórica

- Android Studio acaba de ser 'nombrado' en el 2015 por Google, el IDE oficial para desarrollar aplicaciones para Android.
- Tiene su propio emulador.
- Permite sin necesidad de instalación de un *plugin* auxiliar, la construcción y gestión de proyectos basado en Maven.
- Refactorización y completado avanzado de código Android.
- Vista en tiempo real de renderizado de *layouts*.
- Nuevos módulos en proyecto.
- Editor de navegación.
- Visualización de recursos desde editor de código.

La aplicación será desarrollada en Android Studio pues al compararla con ATD Eclipse es superior a él en cuanto al desarrollo de proyectos para Android. Además de que Eclipse necesita muchos recursos en un ordenador: memoria RAM, disco duro, microprocesador; es por esto que se necesitaría una máquina con altas prestaciones para poder programar en este IDE.

1.6.3 Arduino Software

Para desarrollar Arduino existe el IDE Arduino Software, es de código abierto, esto permite que sea fácil de escribir código y subirlo al tablero. Se ejecuta en Windows, Mac OS X y Linux. El entorno está escrito en Java y basado en *Processing*.

Contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos (17).

Este entorno se utilizará para interactuar y realizar las modificaciones necesarias con el código de la plataforma SmartAP para una eficiente comunicación con la aplicación. Para esto se utilizaron diferentes lenguajes de programación dependiendo del IDE empleado.

1.7 Lenguajes de programación

Un lenguaje de programación permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador, para que éste pueda comunicarse con los dispositivos de hardware y software existentes.

Existen muchos lenguajes de programación de alto nivel hoy en día como Java, C++, C# entre otros. Cada uno de ellos con sus características propias, es por esto que se realiza una búsqueda para ver cuál es el más adecuado para desarrollar la aplicación.

Como se observa en la imagen 1, el lenguaje más utilizado en la actualidad por los desarrolladores para programar en las diferentes plataformas es Java, el lenguaje nativo de Android, que se impone con un 58%, seguido muy de lejos por HTML/CSS/JavaScript con un 13%, las herramientas visuales de programación forman un 10% del total, C/C++ se queda atrás con un 7% y C# se resigna con el 4% restante. El 8% que queda sin asignar va hacia la categoría de otros (18).

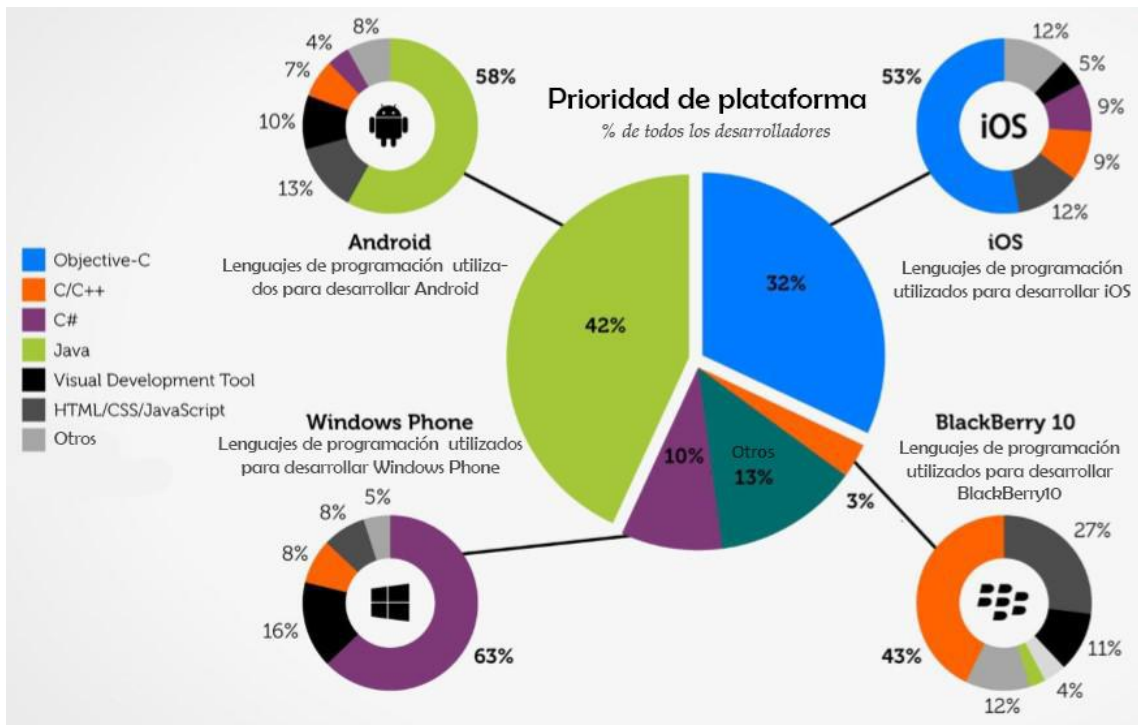


Imagen 1. Lenguajes más utilizados por los desarrolladores (18).

La filosofía de Java es WORA, acrónimo en inglés para “escribe una vez, ejecuta donde sea”. Tiene influencias de Pascal y C++. Java es un lenguaje de programación orientado a objetos, multiplataforma, cuyos desarrollos pueden ejecutarse en páginas web, dispositivos móviles y computadoras (19).

Capítulo 1 Fundamentación teórica

Es por ello que se seleccionó para desarrollar la aplicación Android que le da respuesta al problema planteado a Java como el lenguaje de programación.

Además, para programar las modificaciones necesarias en la plataforma SmartAP se utiliza C++ el cual es el lenguaje estándar de los microcontroladores con software Arduino, aunque es posible programarlo en otros lenguajes.

Es una adaptación de C++ proveniente de avr-libc que provee de una biblioteca de C de alta calidad para usar con los microcontroladores AVR de Atmel (20). Es un lenguaje de medio nivel. Trata con objetos básicos como caracteres, números, también con bits y direcciones de memoria, posee una gran portabilidad y se utiliza para la programación de sistemas: construcción de intérpretes, compiladores, editores de texto, entre otros (21).

Además de la programación de la aplicación se debe realizar el diseño de sus interfaces para esto se utiliza la herramienta de diseño gráfico Adobe Photoshop.

1.8 Herramienta de diseño gráfico Adobe Photoshop

Las herramientas digitales de diseño gráfico son aplicaciones que permiten al usuario crear y editar imágenes de gráficos rasterizados (o imágenes de píxeles) de forma interactiva y almacenarlas en la computadora en un formato de archivo gráfico, como JPEG, PNG, GIF y TIFF.

Adobe Photoshop es el programa de diseño gráfico más vendido de todo el mundo, en más de 100 idiomas. En su plataforma ofrece diversas herramientas muy específicas para poder editar cada elemento de una imagen digital. Este programa forma parte del paquete de Adobe denominado Adobe Creative Suite. Este software se utiliza principalmente para realizar retoques fotográficos, aunque también es muy utilizado para la realización de diseños web, y para la elaboración de cualquier tipo de diseño gráfico, su nombre en español significa literalmente "taller de fotos" (22).

Su uso en el desarrollo de la aplicación abarca todo el diseño de los iconos o botones que se emplearon en las diferentes interfaces de usuario que presenta el sistema. Entre los diseños se encuentran las imágenes de los botones: reproducir/pausar, detener, reproducir pista anterior, reproducir pista siguiente, bajar volumen, subir volumen, conectado/desconectado, secuencial/repetir/aleatorio y el grabar.

1.9 Metodología de desarrollo

Como metodología se denominan a la serie de métodos y técnicas de rigor científico que se aplican sistemáticamente durante un proceso de investigación para alcanzar un resultado teóricamente válido. En este sentido, la metodología funciona como el soporte conceptual que rige la manera en que aplicamos los procedimientos en una investigación (23).

Las metodologías tradicionales son muy efectivas y necesarias para proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Para este proyecto donde se tiene un tiempo limitado es más útil emplear una metodología ágil pues, aunque reduce el costo del cambio en todas las etapas del ciclo de vida del sistema, combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo para garantizar la calidad del software desarrollado y hace que este supere las expectativas del cliente.

Dentro de las metodologías ágiles las que más se destacan son Scrum y programación extrema (XP por sus siglas en inglés Extreme Programming) (24). Las dos comparten características similares como (25):

- Tamaño de procesos pequeños.
- Se consiguen productos usables con mayor rapidez.
- Se atienden las necesidades del usuario con mayor exactitud.
- Se obtiene código más simple y más fácil de entender, reduciendo el número de errores.
- Curva de aprendizaje rápida.

Sin embargo, Scrum está centrada en la gestión de proyectos y XP por su parte se centra en la programación y obtención de un producto final, además genera documentación del software desarrollado.

Luego de hacer el análisis, se selecciona XP como metodología de desarrollo para la presente investigación debido a que se quiere tener una documentación más detallada que se ajuste al proyecto en función.

La Programación Extrema está dividida en 4 fases principales (26):

Capítulo 1 Fundamentación teórica

- Planificación.
- Diseño.
- Desarrollo.
- Pruebas.

El proyecto a realizar es pequeño y necesariamente cambiante en su etapa de desarrollo, a la hora de la selección de los principales requisitos y esta metodología ofrece una estrategia de comunicación fluida entre el cliente y el desarrollador, además de la facilidad para enfrentar los cambios necesarios y continuos.

Consideraciones parciales

Después del análisis del estado del arte se puede llegar a las siguientes consideraciones parciales:

- Se desarrollará una aplicación Android para dispositivos con la versión 4.3+.
- La aplicación a estar centrada en la conexión y reproducción de audio de forma remota en la plataforma SmartAP.
- Las herramientas a utilizar para el desarrollo de la solución serán: el IDE Android Studio y Arduino Software, los lenguajes Java y C++, Adobe Photoshop, siguiendo las pautas de la metodología de desarrollo XP.

Capítulo 2 Planificación y Diseño

Introducción

En este capítulo se realizó una valoración de las primeras dos fases del ciclo de vida de la metodología XP: planificación y diseño. Se establecen las historias de usuario, las cuales servirán para un mejor entendimiento y conocimiento del software. Estas historias de usuario son diseñadas por el proyecto Arex del centro CEDIN (en lo adelante cliente) representando en estas las principales necesidades del sistema. Además, se planifica el proceso de desarrollo, se expone el diseño del sistema y la arquitectura empleada.

2.1 Fase I: Planificación

En esta fase el cliente plantea a grandes rasgos las historias de usuario y establece la prioridad de cada una de ellas, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente (25).

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias (27). Es la fase en la que se define el alcance general del proyecto y el resultado de esta es un Plan de entregas (28).

2.1.1 Las historias de usuario

Las historias de usuario (HU) es la técnica utilizada por XP para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Relación de las historias de usuario

Las HU se representan mediante tablas las cuales contienen las siguientes secciones:

- **Número:** Las siglas de HU más un número consecutivo, este permite la historia de usuario.
- **Nombre:** Nombre que identifica la HU.

Capítulo 2 Planificación y Diseño

- **Prioridad en negocio:** Esta característica es dada por el cliente con los valores: alta, media o baja en dependencia de la importancia en que desean ser implementadas.
- **Iteración asignada:** Número de la iteración en la cual se desarrollará la HU.
- **Puntos estimados:** Tiempo estimado en semanas que se le asignará.
- **Descripción:** Breve descripción del proceso que define la HU.
- **Observaciones:** Alguna acotación importante de señalar acerca de la HU.

Historia de Usuario	
Número: HU_1	Nombre: Conectar la aplicación Android con la plataforma SmartAP.
Programador responsable: Martha García Montero	Prioridad en negocio: Alta
Puntos estimados: 2	Iteración asignada: 1
Descripción: El usuario tendrá la opción de conectar la aplicación Android con la plataforma SmartAP. Para esto primero se verifica si está activo el <i>bluetooth</i> , sino es así solicita el permiso para su activación. Seleccionar dentro de las conexiones que aparecen en la interfaz cuál es la que le corresponde a la plataforma.	
Observaciones: La aplicación reconoce todos los dispositivos que tengan encendido el <i>bluetooth</i> a su alcance hasta 50m.	

Tabla # 2 .HU_1 Conectar la aplicación Android con la plataforma SmartAP.

Historia de Usuario	
Número: HU_2	Nombre: Desconectar la aplicación Android con la plataforma SmartAP.
Programador responsable: Martha García Montero	Prioridad en negocio: Alta
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario tendrá la opción de desconectar la aplicación Android con la plataforma SmartAP. Para esto selecciona el botón “Salir” del menú.	
Observaciones:	

Tabla # 3 .HU_2 Desconectar la aplicación Android con la plataforma SmartAP.

Historia de Usuario

Capítulo 2 Planificación y Diseño

Número: HU_3	Nombre: Importar pistas.	
Programador responsable: Martha García Montero		Prioridad en negocio: Alta
Puntos estimados: 1.5		Iteración asignada: 1
Descripción: El usuario tendrá la opción importar las pistas de audio de la plataforma. Para esto selecciona el botón “Importar pistas” del menú. Se visualizan las pistas en la interfaz principal.		
Observaciones:		

Tabla # 4 .HU_3 Importar pistas.

Historia de Usuario		
Número: HU_4	Nombre: Reproducir pistas.	
Programador responsable: Martha García Montero		Prioridad en negocio: Media
Puntos estimados: 0.5		Iteración asignada: 2
Descripción: El usuario tendrá la opción de reproducir una pista de audio de la plataforma. Para esto selecciona una pista entre las pistas que han sido importadas y el botón “Reproducir”.		
Observaciones:		

Tabla # 5 .HU_4 Reproducir pista.

Historia de Usuario		
Número: HU_5	Nombre: Pausar pista.	
Programador responsable: Martha García Montero		Prioridad en negocio: Media
Puntos estimados: 0.5		Iteración asignada: 2
Descripción: El usuario tendrá la opción de pausar una pista de audio de la plataforma. Para esto selecciona el botón “Pausar” y la pista se detendrá en el mismo lugar.		
Observaciones:		

Tabla # 6 .HU_5 Pausar pista.

Historia de Usuario		
---------------------	--	--

Capítulo 2 Planificación y Diseño

Número: HU_6	Nombre: Detener pista.	
Programador responsable: Martha García Montero	Prioridad en negocio: Media	
Puntos estimados: 0.5	Iteración asignada: 2	
Descripción: El usuario tendrá la opción de detener una pista de audio de la plataforma. Para esto selecciona el botón “Detener” y la pista se reiniciará sin reproducirse.		
Observaciones:		

Tabla # 7 .HU_6 Detener pista.

Historia de Usuario		
Número: HU_7	Nombre: Reproducir pista anterior.	
Programador responsable: Martha García Montero	Prioridad en negocio: Media	
Puntos estimados: 0.5	Iteración asignada: 2	
Descripción: El usuario tendrá la opción de reproducir pista de audio anterior. Para esto selecciona el botón “Anterior” y retrocede una pista.		
Observaciones:		

Tabla # 8 .HU_7 Reproducir pista anterior.

Historia de Usuario		
Número: HU_8	Nombre: Reproducir pista siguiente.	
Programador responsable: Martha García Montero	Prioridad en negocio: Media	
Puntos estimados: 0.5	Iteración asignada: 2	
Descripción: El usuario tendrá la opción de reproducir pista de audio siguiente. Para esto selecciona el botón “Siguiente” y adelanta una pista.		
Observaciones:		

Tabla # 9 .HU_8 Reproducir pista siguiente.

Historia de Usuario		
Número: HU_9	Nombre: Bajar volumen.	

Capítulo 2 Planificación y Diseño

Programador responsable: Martha García Montero	Prioridad en negocio: Media
Puntos estimados: 0.5	Iteración asignada: 2
Descripción: El usuario tendrá la opción de bajar el volumen del audio de la plataforma. Para esto selecciona el botón “Bajar volumen”.	
Observaciones:	

Tabla # 10 .HU_9 Bajar volumen.

Historia de Usuario	
Número: HU_10	Nombre: Subir volumen.
Programador responsable: Martha García Montero	Prioridad en negocio: Media
Puntos estimados: 0.5	Iteración asignada: 2
Descripción: El usuario tendrá la opción de subir el volumen del audio de la plataforma. Para esto selecciona el botón “Subir volumen”.	
Observaciones:	

Tabla # 11 .HU_10 Subir volumen.

Historia de Usuario	
Número: HU_11	Nombre: Reproducción secuencial, repetida o aleatoria.
Programador responsable: Martha García Montero	Prioridad en negocio: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El usuario tendrá la opción de elegir qué tipo de reproducción desea escuchar: secuencial, repetida o aleatoria. Para esto selecciona el botón “Secuencial/Repetir/Aleatorio” y este cambiará según el tipo de reproducción.	
Observaciones: Secuencial es cuando las pistas se reproducen una a continuación de otra según están en la lista de audio. Repetir es cuando la pista en ejecución se puede volver a reproducir. Aleatorio es cuando las pistas saltan de forma aleatoria para su reproducción.	

Tabla # 12 .HU_11 Reproducción secuencial, repetida o aleatoria.

Historia de Usuario

Capítulo 2 Planificación y Diseño

Número: HU_12	Nombre: Eliminar pistas.	
Programador responsable: Martha García Montero		Prioridad en negocio: Baja
Puntos estimados: 0.5		Iteración asignada: 3
Descripción: El usuario tendrá la opción de eliminar una pista de audio de la plataforma. Para esto selecciona una pista entre las pistas que han sido importadas y el botón “Borrar” del menú.		
Observaciones:		

Tabla # 13 .HU_12 Eliminar pista.

Historia de Usuario		
Número: HU_13	Nombre: Enviar pista a la plataforma SmartAP.	
Programador responsable: Martha García Montero		Prioridad en negocio: Baja
Puntos estimados: 1		Iteración asignada: 3
Descripción: El usuario tendrá la opción de enviar pista de audio que ya esté guardada en el dispositivo hacia la plataforma. Para esto selecciona el botón “Enviar archivo” del menú. Marca la ruta del archivo que desea enviar y el botón “Aceptar”.		
Observaciones:		

Tabla # 14 .HU_13 Enviar pista a la plataforma SmartAP.

Historia de Usuario		
Número: HU_14	Nombre: Grabar pista a la plataforma SmartAP.	
Programador responsable: Martha García Montero		Prioridad en negocio: Baja
Puntos estimados: 2		Iteración asignada: 3
Descripción: El usuario tendrá la opción de grabar pista de audio hacia la plataforma. Para esto selecciona el botón “Grabar” del menú. Se puede grabar al elegir la secuencia de opciones “Grabar audio”, “Pausar audio” y por último “Enviar archivo”.		
Observaciones: Antes de enviar el archivo también existe la posibilidad de escuchar la grabación al escoger el botón “Repetir”.		

Tabla # 15 .HU_14 Enviar pista a la plataforma SmartAP.

2.1.2 Requisitos no Funcionales del sistema

Los requisitos no funcionales son propiedades o cualidades que el producto debe cumplir. Estas propiedades son las características que hacen al producto atractivo, usable, fiable, rápido y confiable.

Requisitos de usabilidad

RNF 1. La aplicación tiene como usuario final a los especialistas del proyecto AREX de la línea de desarrollo Sistemas Embebidos que interactúan con la plataforma SmartAP.

Restricciones de diseño e implementación

RNF 2. El producto de software final debe diseñarse sobre un patrón arquitectónico cliente-servidor.

RNF 3. La aplicación se debe desarrollar con el IDE Android Studio empleando el lenguaje Java.

RNF 4. Para el diseño de los botones de las diferentes interfaces de usuario se debe utilizar la herramienta digital Adobe Photoshop

RNF 5. Cuando se desconecte la plataforma de la aplicación, los botones se deshabilitan automáticamente.

Interfaz

RNF 6. La aplicación debe tener un diseño fácil de manipular para un usuario inexperto, donde no sea necesario mucho entrenamiento para ser utilizado.

RNF 7. Los botones utilizados en la aplicación deben tener un nombre o descripción entendible para el usuario.

RNF 8. Los colores que predominarán en las interfaces de la aplicación serán grises, negro y blanco, mientras que para los botones se emplearía el color azul.

RNF 9. Se deben organizar los iconos de tal manera que los utilizados para controlar la reproducción deben colocarse en la parte superior de la aplicación debajo del nombre.

RNF 10. La aplicación muestra en la parte superior el nombre de la misma y el menú de opciones.

Requisitos de hardware

RNF 11. El dispositivo donde será montada la aplicación tiene que tener como mínimo 256MB de RAM.

Requisitos de software

RNF 12. El dispositivo donde será montada la aplicación tendrá como mínimo sistema operativo Android 4.3.

2.1.3 Estimación de esfuerzos.

Para lograr un desarrollo eficiente y satisfactorio, se realizó una estimación de esfuerzos para cada una de las historias de usuario identificadas en el proceso de planificación, llegando a los resultados que se muestran a continuación.

Historias de usuario	Puntos de Estimación (semanas)
Conectar la aplicación Android con la plataforma SmartAP.	2
Desconectar la aplicación Android con la plataforma SmartAP.	1
Importar pistas.	1.5
Reproducir pistas.	0.5
Pausar pista.	0.5
Detener pista.	0.5
Reproducir pista anterior.	0.5
Reproducir pista siguiente.	0.5
Bajar volumen.	0.5
Subir volumen.	0.5
Reproducción secuencial, repetida o aleatoria.	0.5
Eliminar pistas.	0.5

Enviar pista a la plataforma SmartAP.	1
Grabar pista a la plataforma SmartAP.	2
Total	12

Tabla # 16 .Estimación de Esfuerzos.

2.1.4 Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entregas está compuesto por iteraciones de no más de tres semanas. Sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción (25).

Después de haber definido las HU y estimado el esfuerzo propuesto para la realización de cada una de ellas, se tomó la decisión de dividir el sistema en 3 iteraciones, las cuales se detallan a continuación:

Iteración # 1:

La primera iteración tiene como objetivo la implementación de las HU #1, HU #2 y HU #3 con prioridad para el cliente de Alta respectivamente. Se dispone de 4.5 semanas para implementar todas las tareas. Se obtiene como resultado una primera versión del sistema propuesto la cual será mostrada al cliente, para luego pasar a la siguiente iteración. Además de definir la arquitectura de todo el sistema.

Iteración # 2:

En esta iteración se darán cumplimiento a las HU #4, HU #5, HU #6, HU #7, HU #8, HU #9 y HU #10 las cuales hacen alusión al funcionamiento de la aplicación de forma general, con prioridad media respectivamente. Se cuenta con 3.5 semanas para llevar a cabo la implementación de esta iteración. Al finalizar se obtendrá una segunda versión del sistema propuesto y se le hará llegar al cliente la iteración anterior junto con la presente para la aprobación o cambios pertinentes con el cliente.

Iteración # 3:

Capítulo 2 Planificación y Diseño

En esta iteración se darán cumplimiento a las HU #11, HU #12, HU #13 y HU #14 las cuales hacen alusión a funcionalidades que necesita el cliente que estén implícitas en la aplicación, con prioridad baja respectivamente. Se cuenta con 4 semanas para llevar a cabo la implementación de esta iteración. Al finalizar se obtendrá una tercera versión del sistema propuesto y se le hará llegar al cliente la iteración anterior junto con la presente para la aprobación o cambios pertinentes con el cliente.

Plan de duración de las iteraciones.

Para lograr una mayor organización del trabajo se crea un plan de duración de las iteraciones; el mismo tiene como objetivo mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de ellas.

No iteración	Historias de usuario	Duración Total de iteración
Iteración # 1	Conectar la aplicación Android con la plataforma SmartAP. <hr/> Desconectar la aplicación Android con la plataforma SmartAP. <hr/> Importar pistas.	4.5 semanas
Iteración # 2	Reproducir pistas. <hr/> Pausar pista. <hr/> Detener pista. <hr/> Reproducir pista anterior. <hr/> Reproducir pista siguiente. <hr/> Bajar volumen. <hr/> Subir volumen.	3.5 semanas
Iteración # 3	Reproducción secuencial, repetida o aleatoria.	4 semanas

Eliminar pistas.
Enviar pista a la plataforma SmartAP.
Grabar pista a la plataforma SmartAP.

Tabla # 17 .Plan de duración de Iteraciones.

2.1.5 Plan de entregas

A continuación, se muestra el plan de entregas desarrollado para dar solución al problema planteado. Para desarrollar el mismo se tuvieron en cuenta los puntos de estimación para obtener un resultado final.

No Iteración	Duración Total de Iteración	Fecha Inicio	Fecha Final	Versión del sistema
Iteración # 1	4.5 semanas	4/01/2016	5/02/2016	1.0
Iteración # 2	3.5 semanas	8/02/2016	4/03/2016	2.0
Iteración # 3	4 semanas	7/03/2016	4/04/2016	3.0

Tabla # 18 .Plan de entregas

2.2 Fase II: Diseño

La metodología XP sugiere que hay que conseguir diseños simples y claros, procurando hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. (28)

2.2.1 Representación arquitectónica de la solución

La representación arquitectónica de la solución planteada confiere una alta prioridad a los elementos físicos de ambos dispositivos, tanto de la plataforma SmartAP como el dispositivo con sistema operativo Android, así como a el protocolo SmartAP creado para la comunicación entre ellos.

Dispositivo con sistema operativo Android

Por un lado, el dispositivo con Android consta de un módulo de *bluetooth*, el cual constituye el medio físico que soporta la comunicación e interacción con los demás dispositivos, dicho

Capítulo 2 Planificación y Diseño

módulo es controlado por un manejador (*driver*) nativo de la plataforma Android denominado: *BluetoothGattCallback*, el cual provee funcionalidades como las que se muestran en la tabla 19:

Funcionalidad	Descripción
onConnectionStateChange	<p>Se ejecuta automáticamente cuando el estado de la conexión <i>bluetooth</i> cambia, ya sea por conexión o desconexión. Provee el parámetro “estado de conexión” (<i>int status</i>) y “nuevo estado” (<i>int newState</i>), los cuales pueden tomar diferentes valores:</p> <ul style="list-style-type: none">0: desconectado (<i>STATE_DISCONNECTED</i>)1: conectando (<i>STATE_CONNECTING</i>)2: conectado (<i>STATE_CONNECTED</i>)3: desconectando (<i>STATE_DISCONNECTING</i>)
onServicesDiscovered	<p>Se emplea cuando la lista de servicios remotos, características o parámetros del dispositivo remoto han sido actualizadas o nuevos servicios han sido descubiertos. Provee el parámetro “estado de conexión” (<i>int status</i>), el cual toma dos valores:</p> <ul style="list-style-type: none">0: dispositivo remoto escaneado exitosamente (<i>GATT_SUCCESS</i>)257: falla en el escaneo del dispositivo remoto (<i>GATT_FAILURE</i>)
onCharacteristicChanged	<p>Función lanzada como resultado de una notificación del cambio de una característica en la conexión remota. Provee el parámetro “característica <i>bluetooth</i> cambiada” (<i>BluetoothGattCharacteristic characteristic</i>), el</p>

	cual posee información de la característica que ha sufrido cambios en la conexión. Para la solución propuesta brinda grandes oportunidades, para la implementación del protocolo del envío y recepción de comandos.
onCharacteristicRead	Es lanzada cuando el <i>bluetooth</i> del dispositivo remoto lee características del servicio <i>bluetooth</i> brindado por el Android.
onCharacteristicWrite	Función lanzada cuando el <i>bluetooth</i> del dispositivo remoto intenta escribir o enviar características a través del servicio <i>bluetooth</i> brindado por el Android.

Tabla # 19 .Funcionalidades del manejador nativo del sistema operativo Android.

Protocolo SmartAP

Para la comunicación efectiva entre los módulos de *bluetooth* es necesaria la implementación de un protocolo de entendimiento entre ambos denominado SmartAP, en el cual mediante un procedimiento de llamadas y retorno permita intercambiar información asociada con el proceso de reproducción, grabación y envío de pistas. Para el desarrollo de este protocolo se definió un *buffer* de datos de al menos una posición, donde viajará en la posición 0 el comando asociado con la petición, en las posiciones siguientes la información que es necesario transmitir, o parámetros asociados. Como se muestra en la imagen 2.

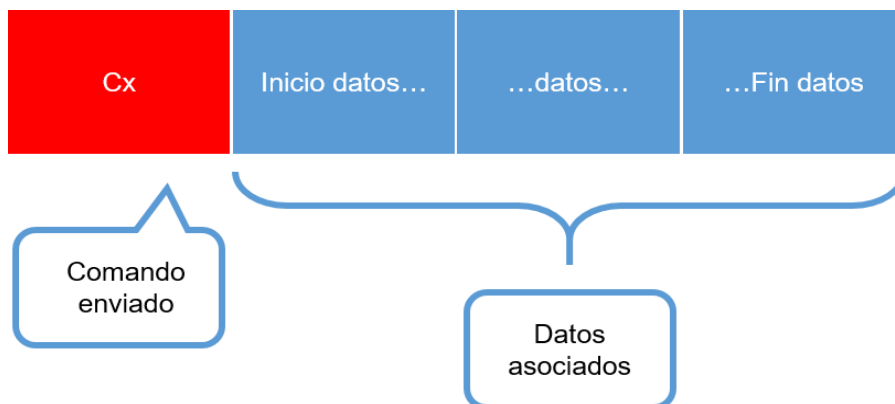


Imagen 2. Formato de transmisión de datos. Elaboración propia.

Capítulo 2 Planificación y Diseño

El protocolo comprende los comandos que se muestran a continuación con sus funcionalidades asociadas, así como la explicación de la naturaleza de los datos que serán agregados al *buffer* en cada caso:

Comando	Valor	Función asociada	Datos en buffer	Retorno
COMMAND_START_PLAYING	0X1	Comando para iniciar la reproducción de una pista de audio específica.	A partir de la posición 1 del <i>buffer</i> se envían los caracteres correspondientes al nombre de la pista a reproducir en formato "8.3".	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.
COMMAND_STOP_PLAYING	0X2	Comando para parar la reproducción que se está realizando.	No incluye datos en el <i>buffer</i> .	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.
COMMAND_PAUSE_PLAYING	0X3	Comando para poner en pausa la reproducción que se está realizando.	No incluye datos en el <i>buffer</i> .	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.
COMMAND_GET_AUDIO_FILE	0X4	Comando para obtener la lista de reproducción asociada a las pistas de audio almacenada en la memoria de la	No incluye datos en el <i>buffer</i> .	Retorna un <i>buffer</i> por cada una de las pistas. En cada <i>buffer</i> se envía el nombre y la extensión de la pista en formato "8.3".

Capítulo 2 Planificación y Diseño

		plataforma SmartAP.		
COMMAND_DELETE_AUDIO_FILE	0X5	Comando para borrar una pista de la memoria de la plataforma SmartAP.	A partir de la posición 1 del <i>buffer</i> se envían los caracteres correspondientes al nombre de la pista que se desea borrar en formato "8.3".	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.
COMMAND_SEND_AUDIO_FILE	0X6	Comando para enviar un archivo determinado desde el dispositivo Android, y grabarlo en la memoria de la plataforma SmartAP.	En la posición 1 del <i>buffer</i> se envía un comando de apoyo al proceso de envío, el cual puede tener 3 estados: 1: inicio de envío de archivo. 2: continuación de envío de archivo. 3: fin de envío de archivo. Los datos binarios asociados al archivo se almacenarán a partir de la segunda posición del <i>buffer</i> con un máximo de	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.

Capítulo 2 Planificación y Diseño

			20 posiciones, o sea que el tamaño máximo de un envío de datos será de 22 posiciones.	
COMMAND_VOLUME_CHANGE	0x7	Comando para modificar el volumen de la reproducción de las pistas: subir o bajar en el caso necesario.	La posición 1 del <i>buffer</i> contendrá solo dos estados: 1: subir volumen un decibel. 2: bajar volumen un decibel.	Verdadero o falso, dependiendo del resultado de la ejecución de la orden.
COMMAND_SEND_NEXT_SONG	0x8	Comando para cambiar a la siguiente canción en la lista de reproducción.	No incluye datos en el <i>buffer</i> .	Retorna un <i>buffer</i> de dos posiciones: en la posición 0 el comando 0x1 (inicio de reproducción), en la posición 1 el índice de la pista que se comienza a reproducir luego de cambiar de pista.
COMMAND_SEND_PREV_SONG	0x9	Comando para cambiar a la canción anterior en la lista de reproducción.	No incluye datos en el <i>buffer</i> .	Retorna un <i>buffer</i> de 2 posiciones: en la posición 0 el comando 0x1 (inicio de reproducción), en la posición 1 el índice de la pista que se comienza a

				reproducir luego de cambiar de pista.
COMMAND_CHANGE_PLAY_MODE	0x10	Comando para cambiar el modo de reproducción de las pistas.	No incluye datos en el <i>buffer</i> .	No incluye datos en el <i>buffer</i> .

Tabla # 20 .Funcionalidades del protocolo. Elaboración propia.

Plataforma SmartAP

La plataforma SmartAP cuenta entre sus componentes con un módulo *bluetooth*. Este módulo es el que se encarga de la comunicación física, para ello el Arduino Micro que funciona como núcleo de la plataforma se auxilia de un manejador para la interacción con este módulo físico: Adafruit_BLE_UART, el cual presenta funcionalidades para la interconexión con otros dispositivos compatibles y sistema operativo Android. De igual forma el Arduino Micro usa el manejador Adafruit_VS1053 para el control de la tarjeta decodificadora de audio que tiene integrado una microSD donde se alojan los datos (normalmente archivos de audio).

La biblioteca Adafruit_BLE_UART para el control del módulo *bluetooth* del Arduino consta de funcionalidades que son usadas para la implementación del protocolo de comunicación en el *firmware*. Para ello se auxilia de las funcionalidades descritas en la tabla siguiente:

Funcionalidad	Descripción
setDeviceName	Función para establecer el nombre indicativo de la plataforma, el cual se debe pasar como único parámetro, este nombre será el que verán los otros dispositivos cuando intenten conectarse a los dispositivos circundantes que tengan activado el <i>bluetooth</i> . Para el caso en la plataforma se implementa como sigue: <code>uart.setDeviceName("SmartAP");</code>
write	Función para enviar un <i>buffer</i> de datos al otro dispositivo de la conexión. Se usa enviando dos

	parámetros: el primer parámetro sería el <i>buffer</i> en sí, el segundo es la longitud de este <i>buffer</i> .
begin	Funcionalidad de inicio del módulo <i>bluetooth</i> , al invocarla automáticamente queda configurado y activado el <i>bluetooth</i> .

Tabla # 21 . Funcionalidades de la biblioteca *Adafruit_BLE_UART*. Elaboración propia.

Para un mejor entendimiento de la representación arquitectónica de la solución del problema planteado se muestra el siguiente diagrama:

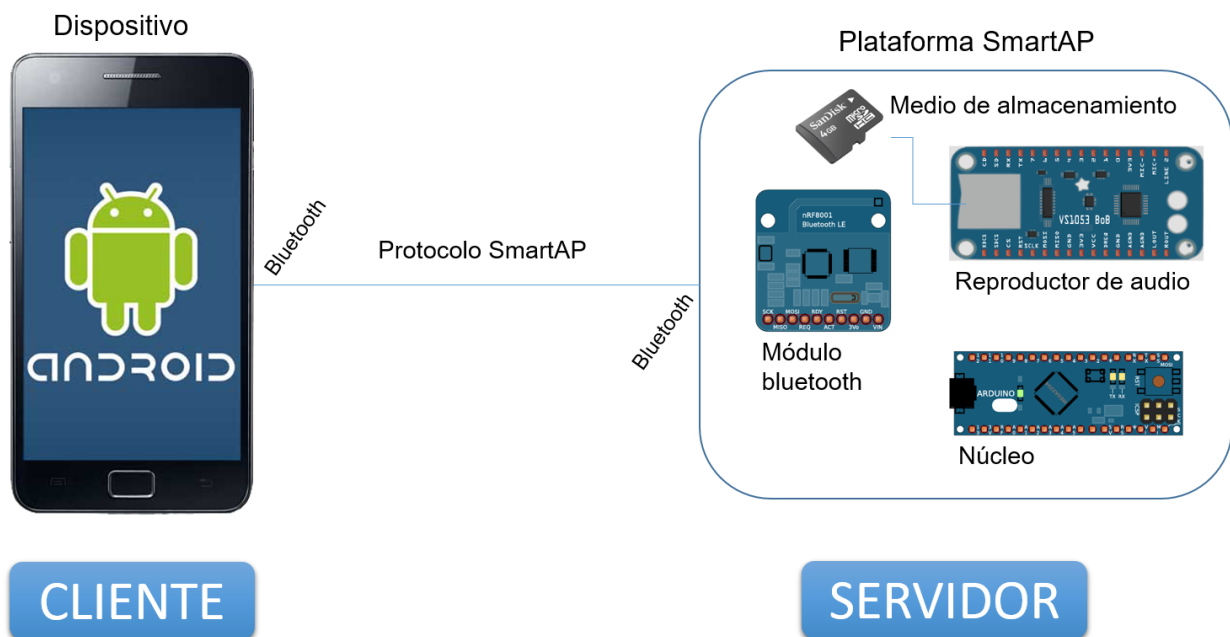


Imagen 3. Representación arquitectónica de la solución. Elaboración propia.

2.2.2 Patrones de Diseño GRASP

GRASP son los patrones generales de software para la asignación de responsabilidades a objetos. Describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades y constituyen la base del cómo se diseñará el sistema (29). En la implementación de la aplicación se utilizaron los siguientes:

- **Experto**

Este patrón se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad (30).

En la solución la clase `BleUart` es la encargada de todo el procesamiento relacionado con la interacción entre el módulo `bluetooth` de la plataforma y el dispositivo Android.

- **Creador**

Este patrón es el responsable de asignarle a una clase la responsabilidad de crear una instancia de otra (30). En la solución la clase `MainActivity` es la que contiene al menos una instancia de la clase `BleUart`, `OGGActivity` y `SimpleFileDialog`.

- **Alta cohesión**

La cohesión es la medida de la fuerza que une a las responsabilidades de una clase. Una clase con alta cohesión mejora la claridad y la facilidad de su uso, su mantenimiento se simplifica y es fácil de reutilizar. A menudo genera un bajo acoplamiento (30). En la solución este patrón fue utilizado en el diseño de la aplicación de manera general, siguiendo la premisa de que cada clase debe contener operaciones que resuelvan necesidades a fines con ellas.

- **Bajo acoplamiento**

Consiste en tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases (30). Este patrón fue utilizado para el diseño de las clases utilizadas en el desarrollo de la aplicación, por ejemplo, en las clases `SimpleFileDialog`, `DeviceScanActivity` y `BleUart`.

2.2.3 Tarjetas CRC (Clases-Responsabilidad-Colaboración)

Las tarjetas clase-responsabilidad-colaboración (CRC) proponen una forma de trabajo, preferentemente grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y cómo colaboran con otros para realizar tareas, las cuáles registran el nombre de las clases, sus responsabilidades y las otras clases con la que colaboran (31).

Para la confección de cada una de las tarjetas se utilizó la tabla cuyo modelo cuenta con los siguientes campos:

- **Clase:** Nombre de la clase a la cual pertenece la tarjeta.
- **Responsabilidad:** Describe cuales son las funcionalidades que deben ser implementadas por la clase.

Capítulo 2 Planificación y Diseño

- **Colaboración:** Enumera las diferentes clases con las cuales tiene relación la clase a la cual pertenece la tarjeta CRC. Se representa la dirección de la clase dentro del código del módulo de configuración.

Tarjeta CRC	
Clase: BleUart	
Responsabilidad:	Colaboración:
sendChangePlayModeCommand isConnected registerCallback connectToDevice unregisterCallback disconnect sendAudioFileCommand sendStopCommand sendPauseCommand sendVolumeCommand sendNextSongCommand sendPrewSongCommand sendPlayCommand sendDeleteFileCommand sendGetAudioFileCommand	

Tabla # 22 . Tarjeta CRC clase BleUart.

Tarjeta CRC	
Clase: DeviceScanActivity	
Responsabilidad:	Colaboración:
onListItemClick onPause onResume	MainActivity

Tabla # 23 . Tarjeta CRC clase DeviceScanActivity.

Tarjeta CRC	
Clase: MainActivity	
Responsabilidad:	Colaboración:
onCreate tipoReproduccion onResume onDestroy sendFile sendFileClick stopPlayingClick pausePlayingClick volumeUpClick volumeDownClick nextSongClick prevSongClick startPlayingClick recordClick onActivityResult deleteAudioFileClick getPlayListClick onCreateOptionsMenu onOptionsItemSelected onConnected onConnectFailed onDisconnected onReceive onCommandGetAudioFileReceived	OGGActivity BleUart SimpleFileDialog

Tabla # 24 . Tarjeta CRC clase MainActivity.

Tarjeta CRC	
Clase: SimpleFileDialog	

Responsabilidad:	Colaboración:
SimpleFileDialog chooseFile_or_Dir createSubDir getDirectories createDirectoryChooserDialog updateDirectory createListAdapter	

Tabla # 25 . Tarjeta CRC clase SimpleFileDialog.

Tarjeta CRC	
Clase: OGGActivity	
Responsabilidad:	Colaboración:
onCreate onDestroy startRecording stopRecording reviewClick senFileClick	

Tabla # 26 . Tarjeta CRC clase OGGActivity.

Consideraciones parciales

- En la fase de planificación se describieron las 14 historias de usuario del sistema y los requisitos no funcionales, las cuales permitieron tener las bases para la implantación de la aplicación Android para reproducir audio de forma remota en la plataforma SmartAP mediante *bluetooth*.
- Se determinó que la duración del desarrollo de la aplicación Android como solución propuesta es de 12 semanas, para lo cual contará con 3 iteraciones, al finalizar cada una se tendrá una entrega funcional de dicha aplicación.
- En la fase de diseño se determinó una representación arquitectónica del sistema utilizado en el desarrollo de la aplicación Android.

Capítulo 3 Desarrollo y Pruebas

Introducción

En el presente capítulo se exponen las principales características de las dos últimas fases del ciclo de vida de la metodología XP: desarrollo y pruebas. Se procede a desarrollar las tareas de la ingeniería que responden a las historias de usuario abordadas en cada iteración, luego mediante las pruebas de aceptación se verifica que el producto resultante cumpla con los requerimientos definidos.

3.1 Fase III: Desarrollo

Dentro de la fase de iteraciones se encuentra la implementación o el desarrollo del código el cual se sustenta en buenas prácticas planteadas por el ciclo de vida de XP para esta etapa entre ellas se encuentran: disponibilidad del cliente, uso de estándares, programación dirigida por las pruebas (“*test-driven programming*”), programación en pares, integraciones permanentes y ritmo sostenido.

3.1.1 Tareas de ingeniería

Asociado a cada iteración se encuentra la planificación de las tareas de ingeniería o programación, cada HU se transforma en estas tareas que son desarrolladas por programadores, dentro del equipo de desarrollo, aplicando la práctica de la programación en parejas. Para cada iteración se realizó la distribución de tareas en correspondencia con las HU que se desarrollaron.

Historia de usuario	Tareas de ingeniería
Conectar la aplicación Android con la plataforma SmartAP.	<ol style="list-style-type: none">1. Desarrollar una función para conectar el dispositivo con la plataforma SmartAP.2. Desarrollar una interfaz visual para la aplicación.
Desconectar la aplicación Android con la plataforma SmartAP.	<ol style="list-style-type: none">1. Desarrollar una función para desconectar el dispositivo con la plataforma.

Capítulo 3 Desarrollo y Pruebas

Importar pistas.	<ol style="list-style-type: none">1. Desarrollar una función para importar las pistas desde la plataforma.2. Desarrollar una función para mostrar las pistas importadas
Reproducir pistas.	<ol style="list-style-type: none">1. Desarrollar una función para reproducir las pistas.
Pausar pista.	<ol style="list-style-type: none">1. Desarrollar una función para pausar las pistas.
Detener pista.	<ol style="list-style-type: none">1. Desarrollar una función para detener las pistas.
Reproducir pista anterior.	<ol style="list-style-type: none">1. Desarrollar una función para reproducir la pista anterior.
Reproducir pista siguiente.	<ol style="list-style-type: none">1. Desarrollar una función para reproducir la pista siguiente.
Bajar volumen.	<ol style="list-style-type: none">1. Desarrollar una función para bajar el volumen de la plataforma.
Subir volumen.	<ol style="list-style-type: none">1. Desarrollar una función para subir el volumen de la plataforma.
Reproducción secuencial o aleatoria.	<ol style="list-style-type: none">1. Desarrollar una función secuencial-aleatoria para cambiar modo de reproducción.
Eliminar pistas.	<ol style="list-style-type: none">1. Desarrollar una función para eliminar pista.
Enviar pista a la plataforma SmartAP.	<ol style="list-style-type: none">1. Desarrollar una función para seleccionar ruta.

	2. Desarrollar una función para enviar archivo.
Grabar pista a la plataforma SmartAP.	1. Desarrollar una función para grabar audio. 2. Reutilizar la función para enviar archivo.

Tabla # 27 . *Tareas de ingeniería*

En el anexo 1 se muestran las tareas de ingeniería realizadas.

3.1.2 Estilos de codificación

Los estilos de codificación o programación definen la estructura y apariencia física del código, lo que facilita su comprensión, mantenimiento y lectura. Los estilos de codificación y documentación explicados en este capítulo son aplicados en el CEDIN, esto justifica el empleo de los mismos en la aplicación propuesta.

Nomenclatura general

- El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras se utilizará el estilo de escritura *lowerCamelCase*, que dicta que para un nombre compuesto por varias palabras comenzará con minúscula, pero todas las palabras internas que lo componen comienzan con mayúscula.

Paquetes

- Por defecto todos los paquetes se escribirán en minúsculas y sin utilizar caracteres especiales.
- Se tendrá, así mismo, otro nivel extra dentro del paquete definido como el nombre del proyecto o de la capa.

Identación

- En el contenido siempre se identificará con tabulaciones, nunca utilizando espacios en blanco.

Clases

- El nombre de las clases comenzará con mayúsculas y cada salto de palabras debe iniciar con mayúsculas.

Métodos

- Los métodos deberán ser verbos (en infinitivo), en mayúsculas y minúsculas con la primera letra del nombre en minúsculas, y con la primera letra de cada palabra interna en mayúsculas.
- No se permiten caracteres especiales.

Nombre de variables

- El nombre de las variables debe empezar con letra minúsculas y de existir un salto de palabra comenzaría con mayúscula.

3.2 Fase IV: Pruebas

La metodología XP propone la aplicación de las pruebas de aceptación utilizadas para evaluar si se obtuvo las funcionalidades deseadas por parte del cliente.

3.2.1 Pruebas de aceptación

Debido a las características del trabajo a realizar, se determina que las pruebas de aceptación son más importantes que las pruebas unitarias ya que la aplicación a desarrollar presenta como objetivo principal el trabajo con interfaces de usuarios y se obtendrá un mejor resultado en función de la aceptación del cliente al realizar este tipo de pruebas, además marcarán el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema y adaptándose a los cambios que el sistema sufra.

Las pruebas de aceptación son creadas en base a las HU, en cada ciclo de la iteración del desarrollo. Son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos (28).

Como criterio de aprobación de cada iteración se tomó que el 100% de los casos de prueba sean exitosos para pasar de iteración. El objetivo de estas pruebas no es tener un conjunto

Capítulo 3 Desarrollo y Pruebas

de casos escritos que cubran el 100% del código, sino poder realizarle pruebas al sistema desde el punto de vista del usuario.

Para la realización de cada una de las pruebas de aceptación se siguieron una serie de pasos que se muestran a continuación:

- Identificar todas las acciones en la HU.
- Para cada acción escribir al menos una prueba.
- Para algunos datos, reemplazar las entradas que hacen que la acción ocurra y llenar en la casilla resultados esperados los resultados obtenidos.
- Para otros datos, reemplazar las entradas que hacen que la acción falle, y registrar los resultados.

Para representar las pruebas de aceptación se definieron los siguientes elementos:

- **Código:** Representa al caso de prueba, incluye el número de HU, de la prueba y si posee diferentes escenarios.
- **HU:** Número de la HU a la cual pertenece.
- **Nombre:** Junto al código conforma el identificador del caso de prueba.
- **Descripción:** Acción que debe realizar el sistema.
- **Condiciones de ejecución:** Describe las características y elementos que debe contener el sistema para realizar el caso de prueba.
- **Entrada/Pasos de ejecución:** Incluye las entradas necesarias para realizar el sistema y los pasos para realizar el caso de prueba.
- **Resultados esperados:** Descripción de la respuesta del sistema ante el caso de prueba.
- **Evaluación de la prueba:** Clasificación de la prueba en satisfactoria o insatisfactoria.

Código: HU1_P1	HU: 1
Nombre: Conectar la aplicación Android con la plataforma SmartAP.	
Descripción: Prueba para la funcionalidad Conectar la aplicación Android con la plataforma SmartAP.	
Condiciones de ejecución: La plataforma y el <i>bluetooth</i> deben estar activos.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se abre la aplicación. 2. Se escanea para buscar todos los dispositivos activos. 3. Se selecciona el que corresponde a la plataforma. 	
Resultado esperado: La aplicación abre correctamente y se conecta de forma efectiva y rápida con la plataforma.	
Evaluación de la prueba: Satisfactoria	

Tabla # 28 .HU1_P1 Conectar la aplicación Android con la plataforma SmartAP.

En el anexo 2 Pruebas de aceptación se muestran los demás casos de prueba desarrollados en las diferentes iteraciones.

3.2.2 Pruebas de aceptación de la primera iteración

En la primera iteración se realizaron pruebas a los HU 1, HU 2, HU 3 donde se comprobó el nivel de aceptación en los usuarios con cada Historia de Usuario. Se detectaron 2 no conformidades en la primera etapa de prueba, en la segunda se encontró solamente 1, mientras que en la tercera no se encontraron errores para un 100% de resultados satisfactorios cumpliendo así con el criterio de aceptación por lo que se decidió pasar a la segunda iteración de pruebas.

3.2.3 Pruebas de aceptación de la segunda iteración

En la segunda iteración se realizaron pruebas a los HU 4, HU 5, HU 6, HU 7, HU 8, HU 9 y HU 10. Se detectaron 4 no conformidades en la primera etapa de prueba, en la segunda se encontraron solamente 3, mientras que en la tercera no se encontraron errores para un

100% de resultados satisfactorios cumpliendo así con el criterio de aceptación por lo que se decidió pasar a la tercera iteración de pruebas.

3.2.4 Pruebas de aceptación de la tercera iteración

En la tercera iteración se realizaron pruebas a los HU 11, HU 12, HU 13 y HU 14. Se detectaron 2 no conformidades en la primera etapa de prueba, en la segunda se encontró solamente 1, mientras que en la tercera no se encontraron errores para un 100% de resultados satisfactorios cumpliendo así con el criterio de aceptación.

3.2.5 Análisis de los resultados obtenidos

Las pruebas de aceptación se realizaron a las funcionalidades de la aplicación, teniendo como punto de referencia: las vistas de las interfaces, interacción entre usuario y aplicación y el cumplimiento de los requisitos del sistema, para comprobar el nivel de aceptación en cada una de las iteraciones donde se encontraron 8 no conformidades en todo el ciclo de pruebas los cuales fueron resueltos con éxito, lo que arrojó los siguientes resultados:

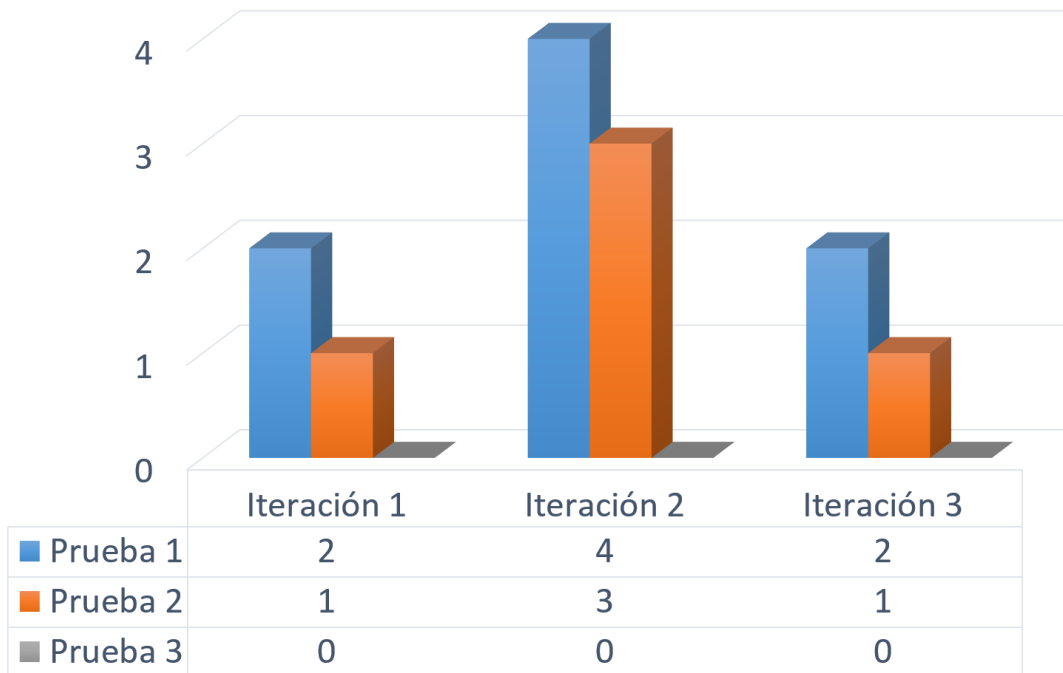


Imagen 4. Resultados de las pruebas de aceptación. Elaboración propia.

Consideraciones parciales

- En la fase de desarrollo se definió la codificación utilizada para las clases, métodos y estructuras de control definidas para la aplicación sistema operativo Android.

Capítulo 3 Desarrollo y Pruebas

- La realización de las pruebas de aceptación permitió la obtención de una aplicación para sistema operativo Android aceptada por el cliente.

Conclusiones generales

Luego de realizado el trabajo se concluye lo siguiente:

- Se implementó el protocolo SmartAP que permitió la comunicación entre la aplicación Android y la plataforma SmartAP.
- Se obtuvo una aplicación Android para reproducir audio en la plataforma SmartAP mediante *bluetooth* que garantiza la conexión entre los dispositivos.

Recomendaciones

Independientemente que se ha cumplido con el objetivo de la investigación, se sugiere ampliar el contenido para su futuro perfeccionamiento, logrando un mayor alcance e impacto mediante las acciones siguientes:

- Crear listas de reproducción.
- Visualizar el tiempo de duración de una pista.

Referencia Bibliográfica

1. Barberá, Javier Colomer. *Aplicación Android para reproducción de audio en plataforma Arduino*. 2015.
2. Venturi, Basterra - Berteza - Borello - Castillo. *Android OS Documentation, Release 0.1*. 2016.
3. <https://developer.android.com/>. [En línea] <https://developer.android.com/>.
4. <http://www.staffcreativa.pe>. [En línea] <http://www.staffcreativa.pe/blog/android-ventajas-desventajas/>.
5. Omar Andrés Rocha Jiménez, Nicolás Steven Martínez Coronado. *Desarrollo de un chat para dispositivos móviles android basado en el protocolo de comunicación bluetooth*. 2013.
6. Castellano, Alvaro Richi. *Bluetooth. Introducción a su Funcionamiento*. 2012.
7. Pozo, Dr.C Antonio Cedeño. *SmartAP, plataforma de hardware para el desarrollo de reproductores de audio*. 2015.
8. Arduino. [En línea] <https://www.arduino.cc/en/Reference/SPI>.
9. Medialab. Básicos-Arduino. [En línea] 2012. <http://medialab.usal.es/facartec/files/2012/10/B%C3%A1sicos-Arduino.pdf>.
10. Aprende Robotica. [En línea] <http://www.aprenderobotica.com/group/eslaprimera vez/page/principiantes-arduino>.
11. Artero, Óscar Torrente. *Arduino : curso práctico de formación*. 2013.
12. <https://www.arduino.cc/>. [En línea] <https://www.arduino.cc/en/Guide/Introduction>.
13. Programacion desarrollo. [En línea] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.
14. Hernández Muñoz, Victor. *Desarrollo de una aplicación móvil en sistema Android para el control remoto de dispositivos mediante la tecnología Bluetooth 4.0*. 2015.
15. Genbeta:dev. [En línea] 10 de Enero de 2014. <http://www.genbetadev.com/herramientas/eclipse-ide>.
16. Azofra, Luis Javier Elorza. *Descubriendo el Chino*. 2015.
17. <https://www.arduino.cc/>. [En línea] <https://www.arduino.cc/en/Guide/Environment>.
18. El androide libre. [En línea] 2014. <http://www.elandroidelibre.com/2014/07/el-70-de-los-desarrolladores-prefiere-android-a-pesar-de-ser-menos-rentable.html>.

19. geekpunto. [En línea] Octubre de 2015. <http://geekpunto.com/lenguajes-de-programacion-mas-usados/>.
20. Aprendiendo Arduino. [En línea] 25 de Marzo de 2015. <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estructura-de-un-programa/>.
21. Aprendiendo Arduino. [En línea] Marzo de 2015. <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-cl/>.
22. Carvalho, Fabiola B., Goncles, Marcelo y Tanomaru-Filho, Mário. *Evaluation of chronic periapical lesions by digital subtraction radiography by using Adobe Photoshop CS: a technical report. Journal of endodontics.* 2007.
23. Significados. [En línea] <http://www.significados.com/metodologia/>.
24. Ana Celmira Gualteros Gualteros, Diana Paola Orjuela Escobar. *Estudio de metodologías ágiles para proyectos de software en corto tiempo.* Bogotá : s.n., 2013.
25. López, Yolanda Borjas. *Metodología Ágil de Desarrollo de Software-XP.* 2015.
26. Escribano, Gerardo Fernández. *Introducción a Extreme Programming.*
27. Letelier, Patricio. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* 2006.
28. Joskowicz, Ing. José. *Reglas y Prácticas en eXtreme Programming.* 2008.
29. Erich Gamma, R.H., Ralph Johnson, John Vlissides. *Patrones de Diseño. Elementos de software orientado a objetos reutilizables.* 2003.
30. Larman, Craig. *UML y patrones. Introducción al análisis y diseño orientado a objetos.* Pearson : s.n., 2003.
31. Vallespir, Ing. Diego. *CRC y un Taller.* Montevideo : s.n., 2002.
32. Ron Jeffries, Ann Anderson, Chet Hendrickson. *Extreme Programming Installed.* 2001.
33. Ruíz, Carlos Andrés Peña. *Informe final de práctica profesional implementación y administración de servicios web en la Cámara de Comercio de Dosquebradas.* 2015.
34. CCM. [En línea] <http://es.ccm.net/contents/148-entorno-cliente-servidor>.
35. José H. Canós, Patricio Letelier y Ma Carmen Penadés. *Métodologías Ágiles en el Desarrollo de Software.* s.l. : DSIC -Universidad Politécnica de Valencia.
36. Molina, Laura. *Desarrollo de aplicaciones para ambientes distribuidos.* 2013.

Referencia Bibliográfica

37. Gimson, Lic. Loraine. *Metodologías ágiles y desarrollo basado en conocimiento*. La Plata : s.n., 2012.
38. Roberth G. Figueroa, Camilo J. Solís. *Metodologías Tradicionales vs. Metodologías Ágiles*. s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

Glosario de Términos

API (Application Programming Interface): Término inglés que significa Interfaz de Programación de Aplicaciones, es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

APK (Android Application Package): Es una extensión o paquete para las aplicaciones del sistema operativo Android.

APP: Término inglés que significa aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros activos informáticos.

ATMEL: Atmel Corp. es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores

AVR: Los AVR son una familia de microcontroladores RISC del fabricante estadounidense Atmel.

BLE (Bluetooth Low Energy): Término inglés que significa *bluetooth* de baja energía, es una versión ligera de *bluetooth* 4.0 diseñada especialmente para disminuir el consumo de energía.

Buffer: Memoria de almacenamiento temporal de información que permite transferir los datos entre unidades funcionales con características de transferencia diferentes.

DACP (Digital Audio Control Protocol): Término inglés que significa Protocolo de Control de Audio Digital fue presentado recientemente por Apple, y está integrado en todas las versiones recientes de iTunes TM.

Firmware: Es un bloque de instrucciones que establece la lógica de bajo nivel que controla los circuitos electrónicos de un dispositivo. Está fuertemente integrado con la electrónica del dispositivo siendo el software que interactúa directamente con el hardware: es el encargado de controlarlo para ejecutar correctamente las instrucciones externas.

Framework: En términos generales, es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

GRASP (General Responsibility Assignment Software Patterns): Término inglés que significa Patrones de Software para la Asignación General de Responsabilidad, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones.

IDE (Integrated Development Environment): Término inglés que significa Ambiente de Desarrollo Integrado o entorno de desarrollo interactivo, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Identación: Es un anglicismo (de la palabra inglesa indentation) de uso común en informática; no es un término reconocido por la Real Academia Española. Este término significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y distinguirlo mejor del texto adyacente; en el ámbito de la imprenta, este concepto siempre se ha denominado sangrado o sangría.

IDII (Interaction Design Institute Ivrea): Término inglés que significa Instituto de Diseño Interactivo, fue un programa de diseño de postgrado en el campo de diseño de interacción que opera en la ciudad de Ivrea, en el norte de Italia.

IoT (Internet of Things): Término inglés que significa Internet de las cosas, es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet.

LED (Light-Emitting Diode): Término inglés que significa diodo emisor de luz, es un componente optoelectrónico pasivo y, más concretamente, un diodo que emite luz.

Máquina virtual Dalvik: Es la máquina virtual que utiliza la plataforma para dispositivos móviles Android. La Máquina Virtual Dalvik (DVM) permite ejecutar aplicaciones programadas en Java.

Maven: es una herramienta de software para la gestión y construcción de proyectos Java, similar a Apache ANT, pero su modelo es más simple ya que está basado en XML.

Microcontrolador: Es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos.

microSD: Las tarjetas microSD o transflash corresponden al formato de tarjeta de memoria flash más pequeña que la miniSD. su uso se ciñe a aplicaciones donde el tamaño es crítico, como los teléfonos móviles, sistemas GPS o tarjetas flash para videoconsolas portátiles

Middleware: Lógica de intercambio de información entre aplicaciones ("interlogical"), es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos.

OHA (Open Handset Alliance): Es una alianza comercial de 84 compañías que se dedica a desarrollar estándares abiertos para dispositivos móviles. Algunos de sus miembros son Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia y Wind River Systems.

Periférico: Es la denominación genérica para designar al aparato o dispositivo auxiliar e independiente conectado a la unidad central de procesamiento de la computadora.

P2P (red peer-to-peer): Término inglés que significa red entre iguales o red entre pares, es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí.

PLC (Programmable Logic Controller): Término inglés que significa Controlador Lógico Programable, es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.

Plugin: Complemento, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se conoce como plug-in, add-on, conector o extensión.

RISC (Reduced Instruction Set Computer): Término inglés que significa Computador con Conjunto de Instrucciones Reducidas, es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores, con El objetivo de posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.

SDK (Software Development Kit): Término inglés que significa Kit de Desarrollo de Software, generalmente un conjunto de herramientas de desarrollo de software que le

permite al programador o desarrollador de software crear aplicaciones para un sistema concreto.

SPI (Serial Peripheral Interface): Término inglés que significa Interfaz de Periféricos en Serie.

SSH (SecureShell): Término inglés que significa Servidor de Shell Seguro, permite que un usuario acceda de forma remota a un sistema Linux, pero las comunicaciones entre el cliente y servidor viajan cifradas desde el primer momento.

TTPop: es uno de los reproductores de música más exitosos y populares para Android, Windows y iOS. Posee una alta calidad de decodificación de mp3, mp4/m4a, wma, entre otros formatos.

XML (eXtensible Markup Language): Término inglés que significa Lenguaje de Marcas Extensible"), es un lenguaje utilizado para almacenar datos en forma legible.

Anexos 1 Tareas de ingeniería

A continuación, se evidencian las tareas ingeniería o programación en las que fueron desglosadas las HU mencionadas anteriormente, para un mejor funcionamiento de la aplicación. Para la confección de cada una de las tareas se utilizó la tabla que cuenta con los siguientes campos:

- **No. de tarea:** Numeración continua que identifica a la tarea.
- **No. de HU:** Número de la HU a la cual pertenece.
- **Nombre de la tarea:** Identificación literal de la tarea.
- **Tipo de tarea:** Tipo de tarea, dígame diseño, desarrollo, prueba.
- **Puntos estimados:** Representación en porciento de la cantidad de tiempo estimada de una semana, que se utilizara para su realización.
- **Fecha inicio:** Fecha estimada de inicio de realización.
- **Fecha fin:** Fecha estimada de fin de realización.
- **Descripción:** Se describe en qué consiste la tarea y qué elementos deben cumplirse para declarar la tarea terminada.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 1
Nombre de la tarea: Desarrollar de una función para conectar el dispositivo con la plataforma SmartAP.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 4/01/2016	Fecha fin: 11/01/2016
Descripción: Se crea una función que permita la conexión entre el dispositivo y la plataforma SmartAp.	

Tabla # 29 .Tarea de ingeniería: Desarrollar de una función para conectar el dispositivo con la plataforma SmartAP.

Tarea de ingeniería	
No. de tarea: 2	No. de UH: 1

Anexos 1 Tareas de ingeniería

Nombre de la tarea: Desarrollar una interfaz visual para la aplicación.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 11/01/2016	Fecha fin: 18/01/2016
Descripción: Se crea la interfaz visual que va a mostrar la aplicación.	

Tabla # 30 .Tarea de ingeniería: Desarrollar una interfaz visual para la aplicación.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 2
Nombre de la tarea: Desarrollar de una función para desconectar el dispositivo con la plataforma.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 18/01/2016	Fecha fin: 25/01/2016
Descripción: Se crea una función que permita desconectar el dispositivo y la plataforma SmartAP.	

Tabla # 31 .Tarea de ingeniería: Desarrollar de una función para desconectar el dispositivo con la plataforma SmartAP.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 3
Nombre de la tarea: Desarrollar una función para importar las pistas desde la plataforma.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 25/01/2016	Fecha fin: 1/02/2016
Descripción: Se crea una función que permita importar las pistas desde la plataforma.	

Tabla # 32 .Tarea de ingeniería: Desarrollar una función para importar las pistas desde la plataforma.

Tarea de ingeniería	
No. de tarea: 2	No. de UH: 3

Anexos 1 Tareas de ingeniería

Nombre de la tarea: Desarrollar una función para mostrar las pistas importadas.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 1/02/2016	Fecha fin: 5/02/2016
Descripción: Se crea una función que después que las pistas sean importadas se muestren en la interfaz de la aplicación.	

Tabla # 33 .Tarea de ingeniería: Desarrollar una función para mostrar las pistas importadas.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 4
Nombre de la tarea: Desarrollar una función para reproducir las pistas.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 8/02/2016	Fecha fin: 12/02/2016
Descripción: Se crea una función que permita seleccionar y reproducir la pista de audio.	

Tabla # 34 .Tarea de ingeniería: Desarrollar una función para reproducir las pistas.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 5
Nombre de la tarea: Desarrollar una función para pausar las pistas.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 12/02/2016	Fecha fin: 15/02/2016
Descripción: Se crea una función que permita pausar la pista.	

Tabla # 35 .Tarea de ingeniería: Desarrollar una función para pausar las pistas.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 6
Nombre de la tarea: Desarrollar una función para detener las pistas.	

Anexos 1 Tareas de ingeniería

Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 15/02/2016	Fecha fin: 19/02/2016
Descripción: Se crea una función que permita detener la pista.	

Tabla # 36 . *Tarea de ingeniería: Desarrollar una función para detener las pistas.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 7
Nombre de la tarea: Desarrollar una función para reproducir la pista anterior.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 19/02/2016	Fecha fin: 22/02/2016
Descripción: Se crea una función que permita reproducir la pista anterior.	

Tabla # 37 . *Tarea de ingeniería: Desarrollar una función para reproducir la pista anterior.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 8
Nombre de la tarea: Desarrollar una función para reproducir la pista siguiente.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 22/02/2016	Fecha fin: 26/02/2016
Descripción: Se crea una función que permita reproducir la pista siguiente.	

Tabla # 38 . *Tarea de ingeniería: Desarrollar una función para reproducir la pista siguiente.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 9
Nombre de la tarea: Desarrollar una función para bajar el volumen de la plataforma.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 26/0/2016	Fecha fin: 1/03/2016

Anexos 1 Tareas de ingeniería

Descripción: Se crea una función que permita bajar el volumen de la plataforma.

Tabla # 39 . *Tarea de ingeniería: Desarrollar una función para bajar el volumen de la plataforma.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 10
Nombre de la tarea: Desarrollar una función para subir el volumen de la plataforma.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 1/03/2016	Fecha fin: 4/03/2016
Descripción: Se crea una función que subir el volumen de la plataforma.	

Tabla # 40 . *Tarea de ingeniería: Desarrollar una función para subir el volumen de la plataforma.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 11
Nombre de la tarea: Desarrollar una función secuencial-repetir-aleatoria para cambiar modo de reproducción.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 7/03/2016	Fecha fin: 10/03/2016
Descripción: Se crea una función que permita intercambia entre un reproducción de forma: secuencial, repetida o aleatoria.	

Tabla # 41 . *Tarea de ingeniería: Desarrollar una función secuencial-repetir-aleatoria para cambiar modo de reproducción.*

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 12
Nombre de la tarea: Desarrollar una función para eliminar pista.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 10/03/2016	Fecha fin: 14/03/2016
Descripción: Se crea una función que permita eliminar una pista.	

Anexos 1 Tareas de ingeniería

Tabla # 42 .Tarea de ingeniería: Desarrollar una función para eliminar pista.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 13
Nombre de la tarea: Desarrollar una función para seleccionar ruta.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 14/03/2016	Fecha fin: 17/03/2016
Descripción: Se crea una función que permita seleccionar la ruta de origen de una archivo de audio.	

Tabla # 43 .Tarea de ingeniería: Desarrollar una función para seleccionar ruta.

Tarea de ingeniería	
No. de tarea: 2	No. de UH: 13
Nombre de la tarea: Desarrollar una función para enviar archivo.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 17/03/2016	Fecha fin: 21/03/2016
Descripción: Se crea una función que una vez seleccionada la ruta de una pista pueda ser enviada hacia la plataforma.	

Tabla # 44 .Tarea de ingeniería: Desarrollar una función para enviar archivo.

Tarea de ingeniería	
No. de tarea: 1	No. de UH: 14
Nombre de la tarea: Desarrollar una función para grabar audio.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 21/03/2016	Fecha fin: 28/03/2016
Descripción: Se crea una función que permita grabar alguna pista de audio desde la aplicación	

Anexos 1 Tareas de ingeniería

Tabla # 45 .Tarea de ingeniería: Desarrollar una función para grabar audio.

Tarea de ingeniería	
No. de tarea: 2	No. de UH: 14
Nombre de la tarea: Reutilizar la función para enviar archivo.	
Tipo de tarea: Desarrollo	Responsable: Martha García Montero
Fecha inicio: 28/03/2016	Fecha fin: 4/04/2016
Descripción: Reutilizar la función que de enviar archivo.	

Tabla # 46 .Tarea de ingeniería: Modificar la función para enviar archivo.

Anexo 2 Pruebas de aceptación

A continuación, mediante tablas se evidencian las pruebas de aceptación realizadas a la aplicación Android.

Pruebas de aceptación	
Código: HU2_P1	HU: 2
Nombre: Desconectar la aplicación Android con la plataforma SmartAP.	
Descripción: Prueba para la funcionalidad Desconectar la aplicación Android con la plataforma SmartAP.	
Condiciones de ejecución: La aplicación tiene que estar encendida y conectada a la plataforma.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Establecer la conexión con la plataforma. 2. Seccionar la opción "Conexión". 	
Resultado esperado: Se desconectará la aplicación de la plataforma.	
Evaluación de la prueba: Satisfactoria	

Tabla # 47 .HU2_P1 Desconectar la aplicación Android con la plataforma SmartAP.

Pruebas de aceptación	
Código: HU3_P1	HU: 3
Nombre: Importar pistas.	
Descripción: Prueba para la funcionalidad Importar pistas.	
Condiciones de ejecución: La aplicación tiene que estar encendida y conectada a la plataforma.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar al menú de opciones. 	

Anexo 2 Pruebas de aceptación

<ol style="list-style-type: none"> 2. Seleccionar la opción “Obtener lista”. 3. Se muestran todas las pistas en la interfaz principal.
Resultado esperado: Se deben mostrar todas las pistas en la aplicación.
Evaluación de la prueba: Satisfactoria

Tabla # 48 .HU3_P1 Importar pistas.

Pruebas de aceptación	
Código: HU4_P1	HU: 4
Nombre: Reproducir pistas.	
Descripción: Prueba para la funcionalidad Reproducir pistas.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar una pista desde la aplicación. 2. Seleccionar la opción “Reproducir”. 	
Resultado esperado: Comienza a reproducirse la pista desde la plataforma.	
Evaluación de la prueba: Satisfactoria	

Tabla # 49 .HU4_P1 Reproducir pistas.

Pruebas de aceptación	
Código: HU5_P1	HU: 5
Nombre: Pausar pista.	
Descripción: Prueba para la funcionalidad Pausar pista.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas y encontrarse en reproducción.	
Entradas/Pasos de ejecución:	

Anexo 2 Pruebas de aceptación

1. Seleccionar la opción "Pausar".
Resultado esperado: Se pausa la pista en el lugar donde se reproduce.
Evaluación de la prueba: Satisfactoria

Tabla # 50 .HU5_P1 Pausar pista.

Pruebas de aceptación	
Código: HU6_P1	HU: 6
Nombre: Detener pista.	
Descripción: Prueba para la funcionalidad Detener pista.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas y encontrarse en reproducción.	
Entradas/Pasos de ejecución:	
1. Seleccionar la opción "Detener".	
Resultado esperado: Se detiene la pista en el lugar donde se reproduce.	
Evaluación de la prueba: Satisfactoria	

Tabla # 51 .HU6_P1 Detener pista.

Pruebas de aceptación	
Código: HU7_P1	HU: 7
Nombre: Reproducir pista anterior.	
Descripción: Prueba para la funcionalidad Reproducir pista anterior.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas y encontrarse en reproducción.	
Entradas/Pasos de ejecución:	
1. Seleccionar la opción "Anterior".	

Anexo 2 Pruebas de aceptación

Resultado esperado: Poder reproducir pista anterior.
Evaluación de la prueba: Satisfactoria

Tabla # 52 .HU7_P1 Reproducir pista anterior.

Pruebas de aceptación	
Código: HU8_P1	HU: 8
Nombre: Reproducir pista siguiente.	
Descripción: Prueba para la funcionalidad Reproducir pista siguiente.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas y encontrarse en reproducción.	
Entradas/Pasos de ejecución:	
1. Seleccionar la opción "Siguiente".	
Resultado esperado: Poder reproducir pista siguiente.	
Evaluación de la prueba: Satisfactoria	

Tabla # 53 .HU8_P1 Reproducir pista siguiente.

Pruebas de aceptación	
Código: HU9_P1	HU: 9
Nombre: Bajar volumen.	
Descripción: Prueba para la funcionalidad Bajar el volumen.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas.	
Entradas/Pasos de ejecución:	
1. Seleccionar la opción "Bajar volumen".	
Resultado esperado: El volumen de la plataforma disminuye.	
Evaluación de la prueba: Satisfactoria	

Anexo 2 Pruebas de aceptación

Tabla # 54 .HU9_P1 Bajar volumen.

Pruebas de aceptación	
Código: HU10_P1	HU: 10
Nombre: Subir volumen.	
Descripción: Prueba para la funcionalidad Subir el volumen.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas.	
Entradas/Pasos de ejecución: 1. Seleccionar la opción "Subir volumen".	
Resultado esperado: El volumen de la plataforma aumenta.	
Evaluación de la prueba: Satisfactoria	

Tabla # 55 .HU10_P1 Subir volumen.

Pruebas de aceptación	
Código: HU11_P1	HU: 11
Nombre: Reproducción secuencial, repetida o aleatoria.	
Descripción: Prueba para la funcionalidad Reproducción secuencial, repetida o aleatoria.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas.	
Entradas/Pasos de ejecución: 1. Seleccionar la opción "Secuencial/Repetir/Aleatorio".	
Resultado esperado: La forma de reproducción cambia según la opción en la que se encuentra: secuencial, repetida o aleatorio.	
Evaluación de la prueba: Satisfactoria	

Tabla # 56 .HU11_P1 Reproducción secuencial, repetida o aleatoria.

Pruebas de aceptación	
-----------------------	--

Anexo 2 Pruebas de aceptación

Código: HU12_P1	HU: 12
Nombre: Eliminar pistas.	
Descripción: Prueba para la funcionalidad Eliminar pistas.	
Condiciones de ejecución: Las pistas tienen que haber sido importadas.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar una pista. 2. Seleccionar la opción "Borrar" del menú. 3. Seleccionar la opción "Obtener listas" del menú 	
Resultado esperado: La archivo eliminado no saldrán nuevamente en la lista de pistas.	
Resultado obtenido:	
Evaluación de la prueba: Satisfactoria	

Tabla # 57 .HU12_P1 Eliminar pistas.

Pruebas de aceptación	
Código: HU13_P1	HU: 13
Nombre: Enviar pista a la plataforma SmartAP.	
Descripción: Prueba para la funcionalidad Enviar pista a la plataforma SmartAP.	
Condiciones de ejecución: La aplicación tiene que estar encendida y conectada a la plataforma.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar la opción "Enviar archivo" del menú. 2. Seleccionar la ruta del archivo a enviar hacia la plataforma. 3. Seleccionar la opción "Aceptar". 	

Anexo 2 Pruebas de aceptación

Resultado esperado: Archivo enviado sin error a la plataforma.
Resultado obtenido:
Evaluación de la prueba: Satisfactoria

Tabla # 58 .HU13_P1 Enviar pista a la plataforma SmartAP.

Pruebas de aceptación	
Código: HU14_P1	HU: 14
Nombre: Grabar pista a la plataforma SmartAP.	
Descripción: Prueba para la funcionalidad Grabar pista a la plataforma SmartAP.	
Condiciones de ejecución: La aplicación tiene que estar encendida y conectada a la plataforma.	
Entradas/Pasos de ejecución: <ol style="list-style-type: none">1. Seleccionar la opción "Grabar" del menú.2. Seleccionar la opción "Grabar audio".3. Seleccionar la opción "Pausar audio".4. Seleccionar la opción "Enviar archivo".	
Resultado esperado: Archivo grabado y enviado sin error a la plataforma.	
Evaluación de la prueba: Satisfactoria	

Tabla # 59 .HU14_P1 Grabar pista a la plataforma SmartAP.