PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS Y COMPUTACIÓN

DISERTACIÓN PEVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

"DISEÑO Y DESARROLLO DE UNA APLICACIÓN MÓVIL APLICANDO LA METODOLOGÍA SCRUM, QUE PERMITA EL RECONOCIMIENTO DE CANTOS DE RANAS. CASO DE ESTUDIO: ANUROS DE LOS ANDES DEL ECUADOR. "

AUTORES:
- **EDISON MAURICIO GUDIÑO CAÑAR**
- **ANDRÉS SEBASTIÁN PAEZ MORA**

DIRECTOR: ING. DAMIÁN NICOLALDE

QUITO, 2018

DEDICATORIA

Dedico este trabajo a mi familia y a mi novia.

A mi tío, quien siempre ha estado prestando su apoyo y enseñanzas.

A mi abuela, quien me demostró con su vida la importancia del trabajo duro y perseverancia.

A mis padres, quienes me apoyaron en todas las decisiones que he tomado.

A mi novia, quien me ha apoyado y a mi lado a lo largo de estos años.

A mis amigos, quienes han formado parte de mi vida desde la infancia.

Andrés Sebastián Páez Mora

AGRADECIMIENTO

A todas las personas que conocí a lo largo de este camino, a mis compañeros con quienes compartimos muchos días buenos y malos, a los maestros quienes siempre estuvieron prestos a solventar a cualquier duda.

Al Ing. Damián Nicolalde, director del proyecto de investigación, por la colaboración brindada durante todo el desarrollo del proyecto.

Al Ing. Andrés Estrella, Miembro del grupo de investigación, por asesorarnos en una parte muy importante del proyecto como es la parte teórica y técnica.

Andrés Sebastián Páez Mora

DEDICATORIA

A mi mamá

Por creer en mí desde el principio y por haberme aconsejado cuando lo necesitaba, lo cual me permitió seguir adelante en este difícil camino. Además, por haber hecho todo lo que estuvo a su alcance para que yo pudiese cumplir con este importante logro en mi formación profesional.

A mi abuelita

Que desde su partida se convirtió en una fuente de motivación para afrontar y seguir adelante en todos los obstáculos que se presentaron en mi vida.

A mi sobrinito y ahijado Benjamín

Por siempre ser una inspiración para esforzarme más cada día y especialmente por llenar de amor y alegría mi vida desde el momento que me convertí en tío.

Edison Mauricio Gudiño Cañar

AGRADECIMIENTO

A toda mi familia, hermanos, primos, tíos, madrinas, padrinos que me apoyaron y estuvieron conmigo brindándome su apoyo, además por compartir durante este tiempo conmigo buenos y malos momentos.

A todos mis amigos que conocí en el transcurso de la carrera que en algún momento me apoyaron cuando más lo necesitaba.

Al Ing. Damián Nicolalde, director del proyecto de investigación, por la colaboración brindada durante todo el desarrollo del proyecto.

Al Ing. Andrés Estrella, Miembro del grupo de investigación, por asesorarnos en una parte muy importante del proyecto como es la parte teórica y técnica.

Gracias a todas las personas que contribuyeron directa e indirectamente con el desarrollo de este proyecto.

Edison Mauricio Gudiño Cañar

RESUMEN

Este trabajo de disertación de grado forma parte del proyecto de investigación denominado “Diseño de un sistema de estimación de indicadores de biodiversidad en base a un algoritmo de análisis automático de audio digital”, financiado por la Pontificia Universidad Católica del Ecuador. El enfoque particular de este trabajo consiste en el desarrollo de un aplicativo móvil para el monitoreo e identificación de anuros en el Ecuador. Esto facilitara el trabajo de los investigadores que realizan actualmente este proceso de una forma manual.

El proceso de desarrollo de software tiene como marco de trabajo la metodología ágil SCRUM, se decidió emplear esta metodología después de realizar un estudio del arte empleando la metodología denominada “revisión sistémica de la literatura”, esto nos brindó un espectro de conocimiento importante alrededor de este tipo de aplicativos y las metodologías de desarrollo existentes. Actualmente no existe un aplicativo similar en el mercado, por lo que esta aplicación es de carácter innovador.

En segundo lugar, después de realizar la investigación del estado del arte realizamos la recopilación de requerimientos de acuerdo a la metodología SCRUM, realizamos un breve diseño del sistema aplicando diagramas UML, que nos permitió nivelar los conocimientos acerca del sistema con todo el equipo. Definimos roles necesarios para la metodología y realizamos la planificación en el Sprint 0, posteriormente iteramos a lo largo del proceso de acuerdo a las especificaciones de la metodología.

El algoritmo empleado para el reconocimiento fue propuesto como parte del trabajo de investigación, donde se aplican conceptos como: procesamiento de señales digitales de audio y aprendizaje supervisado, teniendo en cuenta que estos temas tienen un protagonismo principal actualmente en la palestra científica.

Finalmente, presentamos los resultados del proceso de desarrollo y el aplicativo construido de acuerdo a lo esperado. El aplicativo realiza el reconocimiento de especies de anuros en el Ecuador, analizando un archivo de audio que puede ser cargado o grabado en el campo, devuelve un gráfico donde se muestra los cantos pertenecientes a las diferentes especies en los diferentes intervalos de tiempo. Para concluir expusimos algunas recomendaciones que pueden ser tomadas en cuenta al momento de escalar este aplicativo

Contenido

ILUSTRACIONES Y TABLAS

CAPÍTULO 1: INTRODUCCIÓN Y FUNDAMENTOS TEÓRICOS

Este capítulo tiene como objetivo principal el contextualizar la importancia del monitoreo e identificación de especies de anuros en el Ecuador y los principales componentes tanto teóricos como prácticos que se aplicaron al momento de desarrollar este trabajo de disertación. Describiremos los antecedentes de investigaciones similares a la realizada durante este trabajo y las metodologías para desarrollar proyectos de esta naturaleza, que implican el desarrollo de software.

1.1 INTRODUCCIÓN

En el campo de la biología, el monitoreo de ejemplares forma parte trascendental del estudio de especies. Las ranas en nuestro país se encuentran alrededor de todo el territorio nacional y en diferentes hábitats, el cambio climático ha reducido dramáticamente este hábitat por lo que el cuantificar que especies siguen luchando por la supervivencia es trascendental para el futuro las mismas. (Toro, Giraldo Gomez, & Salazar Jimenez, 2006)

A lo largo de los años este monitoreo se lo ha venido realizando manualmente después de escuchar largas horas de grabaciones donde mediante la escucha de un especialista puede determinar qué tipo de especie es.(Nicolalde, D., & Pareja, 2016)

1.2 ANTECEDENTES

Muchos de los procesos relacionados con la recolección de cantos siempre se los ha realizado manualmente. El proceso de recolección de datos referentes a las ranas en el Ecuador tiene que ser mediante grabaciones por lo general estas se encuentran en un rango de 6 a 12 horas continuas, las cuales deben ser escuchadas y mediante la comparación con sonidos previamente extraídos de los cantos de ranas se puede lograr identificar diferentes especies por sus cantos.(Nicolalde, D., & Pareja, 2016)
Hoy en día gracias a la globalización de la tecnología se ha logrado que la mayoría de las personas puedan tener acceso a una gran variedad de dispositivos tecnológicos especialmente a dispositivos móviles, que se han vuelto tan necesarios en el día a día de la gran mayoría de personas. (Nicolalde, D., & Pareja, 2016)

La tecnología en estos procesos cumple un papel importante ya que con el uso de esta se puede lograr una automatización completa y unificación de los procesos de recolección, selección y etiquetado de cantos de especies de ranas. Esta automatización se puede conseguir gracias diversas herramientas tales como grabadoras de alta calidad , software para visualización y análisis de audio, paquetes de software matemático, lenguajes de programación, algoritmos computacionales de reconocimiento, etc.(Caycedo-Rosales, Ruiz-Muñoz, & Orozco-Alzate, 2013)

1.3 JUSTIFICACIÓN

Actualmente, la Pontificia Universidad Católica del Ecuador se encuentra desarrollando un proyecto de investigación denominado “Diseño de un sistema de estimación de indicadores de biodiversidad en base a un algoritmo de análisis automático de audio digital”, los usuarios de sistemas de reconocimiento de biodiversidad en su mayoría biólogos, entre ellos ha surgido la necesidad de poder realizar análisis audio en el campo mediante un aplicativo para dispositivos móviles.
Se plantea el diseño y desarrollo de la aplicación móvil puesto que dentro del proyecto de investigación mencionado anteriormente también se desarrollará una aplicación web la cual está a cargo de otros integrantes del equipo de investigación , pero la gran desventaja de la aplicación web es que no se puede realizar las pruebas en campo debido a que se necesita un PC o Laptop y además poseer acceso a internet , por lo que desarrollar la aplicación para dispositivos móviles es una gran solución frente a las desventajas antes mencionadas.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar una aplicación móvil aplicando la metodología SCRUM, que permita el reconocimiento de cantos de ranas en base a un algoritmo de análisis de audio.

1.4.2 OBJETIVO ESPECIFICOS

- Desarrollar una aplicación móvil para el proyecto de investigación “Diseño de un sistema de estimación de indicadores de biodiversidad en base a un algoritmo de análisis automático de audio digital”.
- Aplicar guías para el diseño de interfaz de usuario en aplicaciones móviles.
- Implementar el algoritmo de reconocimiento de audio digital para la detección de canto de ranas.
- Aplicar la metodología SCRUM para el desarrollo de una aplicación móvil.

1.5 FUNDAMENTOS TEORICOS

1.5.1 APLICACIONES MOVILES

Con el paso del tiempo la tecnología móvil ha ido evolucionando, para ser exactos la última década ha sido una época dorada para dicha tecnología puesto que ha ido integrando poco a poco servicios y prestaciones que agregan mucho valor para los usuarios (Gasca Mantilla, Camargo Ariza, & Medina Delgado, 2014) consiguiendo dispositivos móviles muy poderosos y capaces de desarrollar tareas complejas.

La inclusión de servicios y prestaciones en los equipos móviles está directamente relacionada con la evolución del hardware presente en estos dispositivos, cada vez su tamaño y costo se reducía mientras que su capacidad aumentaba. La necesidad de un desarrollo mayor de software fue evidente y las empresas se enfocaron tanto en el desarrollo del hardware y software permitiéndonos en la actualidad utilizar los llamados “teléfonos inteligentes”.

La naturaleza propia de los dispositivos móviles como la mayoría de tecnología existente es analógica (Gasca Mantilla et al., 2014). Poco a poco fue transformando su tecnología a digital. Algunos de los servicios más importantes que estos dispositivos proveen son conexión a internet, compatibilidad con sensores, GPS, USB entre otros(Gasca Mantilla et al., 2014). El software tiene como principal misión el aprovechar este hardware disponible.

Se han desarrollado varios tipos de dispositivos móviles como; Computadoras portables cuyo principal objetivo es tener similares prestaciones a las de una computadora personal de escritorio, Tablet PC probablemente estos dispositivos son los más nuevos en el mercado tienen prestaciones similares a las de una computadoras portable pero la gran mayoría no cuenta con un teclado físico por lo que su tamaño disminuye, Asistente Personal Digital (PDA)

son dispositivos pequeños que aceptan escritura manual y son relativamente pequeños, Celulares son dispositivos de bajo costo que cuentan solo con prestaciones como mensajes de texto, llamadas de voz y mensajes multimedia, Teléfonos inteligentes (Smartphones) estos son dispositivos que combinan las funcionalidades de computadoras portables, celulares y PDA's(Georgiev, Georgieva, & Smrikarov, 2004)Cuentan con conexión a internet, WIFI, GPS. Cuentan con cámaras integradas, micrófonos y parlantes además de procesadores que podrían ejecutar tareas medianamente complejas no cuentan con teclado y son fáciles de transportar(Georgiev et al., 2004) .

Ya que el objetivo del proyecto de investigación en el que este trabajo de disertación se basa hablaremos estrictamente de los Teléfonos Inteligentes de ahora en adelante smartphones. Según Gartner:

"Un teléfono inteligente es un dispositivo de comunicaciones móviles que usa un sistema operativo abierto identificable. Un sistema operativo abierto es compatible con aplicaciones de terceros escritas por una notable comunidad de desarrolladores. Las aplicaciones de terceros se pueden instalar y eliminar, y se pueden crear para el sistema operativo del dispositivo y las interfaces de programación de aplicaciones (API). Alternativamente, los desarrolladores deben poder acceder a las API a través de una capa discreta como Java. El sistema operativo debe admitir un entorno multitarea e interfaz de usuario que pueda manejar múltiples aplicaciones simultáneamente. Por ejemplo, puede mostrar un correo electrónico mientras reproduce música." (Gartner, 2018)

Con esta definición está claro lo que es un dispositivo móvil y podemos extraer que una aplicación móvil es desarrollada por terceros, estas pueden ser instaladas y eliminadas por el usuario y orientadas hacia un sistema operativo. Pero esta definición no es lo suficiente mente clara, al ser este un proyecto de desarrollo la definición que usaremos será: Una aplicación móvil es una colección de pantallas que consumen los recursos de un dispositivo móvil para procesar los datos del usuario, puede ser instalada o desinstalada en cualquier momento y está orientada hacia un sistema operativo móvil.

Estas aplicaciones pueden ser muy variadas de acuerdo a su contenido como:

- Arte y diseño
- Autos y vehículos
- Belleza
- Bibliotecas y demostración
- Comida
- Compras

- Comunicación
- Deportes
- Educación
- Entretenimiento
- Estilo de vida Juegos
- Finanzas
- Entre otros.

(Google Inc., 2017)

1.5.2 METODOLOGÍAS DE SOFTWARE

En la actualidad sería una errata el ignorar el relieve que el software ha alcanzado en las últimas décadas, actualmente hablamos de que el software es la base de: naciones, entidades económicas y proceso trascendentales para que el mundo funcione como lo conocemos, todo esto gracias a sus ventajas operativas sobre métodos empleados anteriormente. Como principales ventajas podemos encontrar la escalabilidad, seguridad, globalización, integración y muchos más, que el software brinda a ciertos procesos determinados en comparación con métodos como la escritura manual, utilizados en antaño.

Las ventajas del software nos dan un lineamiento claro de porque lo necesitamos, pero cuál es la definición software. El término 'Ingeniería de software' es acuñado en 1968 durante una conferencia denominada NATO (Schneider, Ph, & Shipp, 2010) , a esta conferencia asistieron desarrolladores de todo el mundo y fue convocada ya que, en ese momento de la historia, la construcción de programas atravesaba una época complicada, posteriormente denominada 'Crisis del software'(Schneider et al., 2010). Esta conferencia buscaba encontrar solución al problema de no tener un marco de trabajo genérico para construir software a diferentes escalas, en este momento el hardware empezó a mejorar y los requerimientos de los programas a aumentar, esto dificulto totalmente la labor de su construcción. Desde el principio de los tiempos los programas eran usados por personas que deseaban agilizar ciertos cálculos en específico o científicos que buscaban simular situaciones en el computador, por lo que el empezar a generar aplicaciones a gran escala y con diferentes funcionalidades era impensable en esos momentos. Es importante tomar en cuenta que sumado a esto la naturaleza del software intangible y abstracto complicaba mucho más esta tarea.

Como solución a esta denominada ‘crisis del software’ aparece un término principal “metodología desarrollo”(Schneider et al., 2010). Una metodología de desarrollo de un sistema se refiere al marco de trabajo que es empleado para estructurar, planificar y controlar el proceso constructivo de un sistema de información(Schneider et al., 2010). Con esta definición surgen algunas dudas como por ejemplo ¿Todos los sistemas requieren una metodología para su construcción? o ¿Porque es importante aplicar una metodología en el desarrollo de sistemas?

La principal razón para aplicar una metodología a el proceso de desarrollo de un sistema es que nos permite tener un marco de trabajo con calidad en la construcción de sistemas, por principios de calidad si no tenemos un proceso de construcción de calidad nuestro producto no será de calidad.

A lo largo de los años con la evolución de los sistemas informáticos las metodologías para su construcción han ido evolucionando y se han desarrollado varias, cada una con sus fortalezas y debilidades. Esto nos da un indicio de que no existe una solución universal para el desarrollo de software, en cada uno de los casos depende mucho del contexto del proyecto para poder tomar la decisión de aplicar cierta metodología(Awad, 2005).

Durante este trabajo de disertación veremos breves rasgos de cada algunas de las metodologías para desarrollar software que han marcado su aporte, así podremos tener un panorama claro de la situación actual de las metodologías y de porque elegimos la metodología propuesta para realizar este proyecto.

La historia empieza en 1970 donde Winston Royce propone una metodología denominada Waterfall en español cascada. Esta metodología es un modelo lineal con sus fases bien definidas, una fase se define como el conjunto de actividades a realizar y debe entregar un producto al final de la fase. La primera fase es donde se recaba lo que se necesita construir, en la siguiente fase diseñaremos como lo vamos a construir, en la tercera fase construiremos el producto en base al diseño, en la siguiente fase probaremos nuestro sistema y por último lo implementaremos en producción (Awad, 2005).

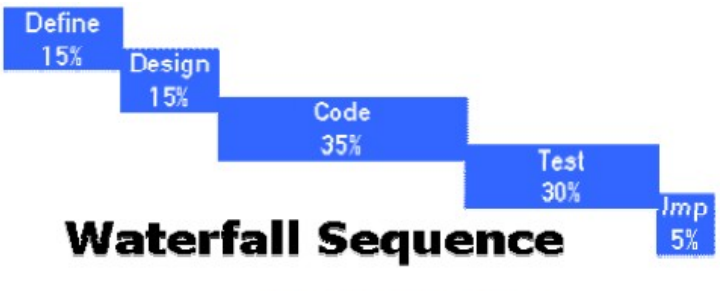


Ilustración 1 Awad(2015) Waterfall Sequence.[Figura]. Recuperado de: A Comparison between Agile and Traditional Software Development Methodologies

Después de pasar por metodologías como la denominada “codifica y corrige” de naturaleza informal y simple aparece la metodología conocida como metodología en cascada donde se definen los pasos a realizar, pero aplicable a muy pocos casos y equipos por su característica rígida. Se intentan proponer mejoras a esta metodología, posteriormente aparecen las metodologías iterativas. (Castro Gil, 2015)

El proceso unificado del desarrollo de software se basa en un desarrollo iterativo cuyo resultado por iteración es un producto que el usuario puede probar, lo que quiere decir completamente funcional (Castro Gil, 2015). A continuación, mostraremos un gráfico donde se pueden identificar las

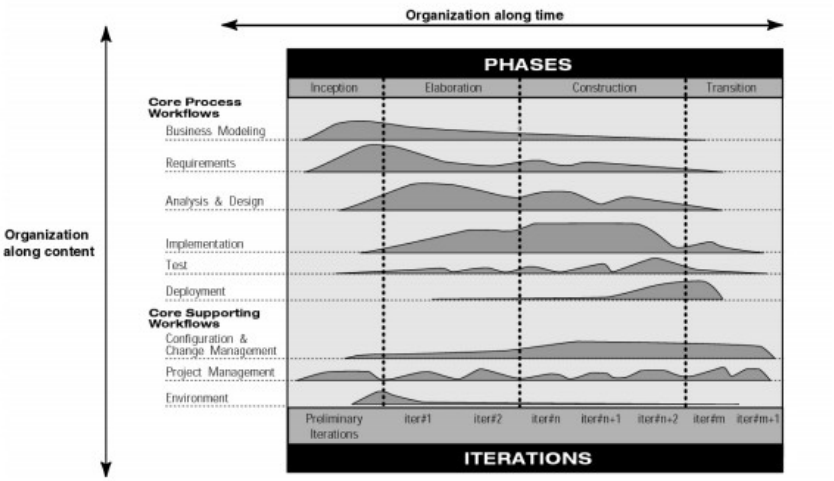


Ilustración 2 (Serrano-Cinca, Fuertes-Callén, & Mar-Molinero, 2005). The Iterative Model graph,[Figura]. Recuperado de: Measuring DEA efficiency in Internet companies

actividades a realizar durante este proceso de desarrollo de software:

Otra metodología utilizada es el modelo en espiral propuesto por Barry Boehm, esta metodología combina fases y elementos tanto del diseño como del prototipado para construir un producto de software (Awad, 2005). Tiene como base para su construcción la experiencia de la aplicación y parametrización del modelo en cascada. Existen cuatro fases importantes para en esta metodología: Definición de objetivos donde se identifican lo que se espera del proyecto, Evaluación y reducción de riesgos donde se evalúan posibles riesgos y con base en investigación se los reduce al mínimo posible, desarrollo y validación donde se construye el producto y se selecciona un modelo apropiado para la siguiente fase, Planeación es revisado el estado del proyecto y se planea las actividades para la siguiente fase(Awad, 2005).

Acabamos de describir las metodologías rígidas que han sido utilizadas en el desarrollo de software. Estas metodologías son denominadas rígidas porque imponen un proceso que hay que seguir con disciplina, por lo general estas metodologías tienen asociadas características como; alcance predictivo, documentación extensa y entendible, orientada hacia un proceso estricto y orientada a herramientas compatibles(Awad, 2005). Con el paso del tiempo y el avance tecnológico, la construcción de software añadió una variable denominada incertidumbre (Galiano, 2016), las metodologías rígidas no tenían una manera adecuada de enfrentar con esta variable por lo que una serie de alternativas salieron a la luz con el nombre de metodologías agiles.

Según la Real Academia de la Lengua la palabra ágil tiene 3 significados: "Que se mueve con soltura y rapidez,

Dicho de un movimiento: Hábil y rápido,

Que actúa o se desarrolla con rapidez o prontitud." (Española, 2017), ninguna de estas definiciones habla sobre la incertidumbre ni mucho menos, entonces porque llevan este nombre estas metodologías cuyo principal objetivo son mitigar esta variable del proceso de desarrollo. Para explicarlo primero definamos incertidumbre con un enfoque en los proyectos TIC, incertidumbre es la falta de certeza en cuanto al alcance, interlocutores, presupuesto y fechas que envuelven al proyecto (Galiano, 2016). Entonces la agilidad referente a los proyectos de TIC tiene que ser una habilidad propia de la metodología capaz de lidiar con cambios al alcance, presupuesto, fechas y ser lo suficientemente fácil para que nuevos desarrolladores la apliquen.

A continuación, describiremos algunas de las metodologías que han ganado popularidad para el desarrollo de software en los últimos años; "Extreme programming" es una metodología que cuenta con ciclos de desarrollo cortos, a la par de que se va aumentando la planeación con una retroalimentación constante.

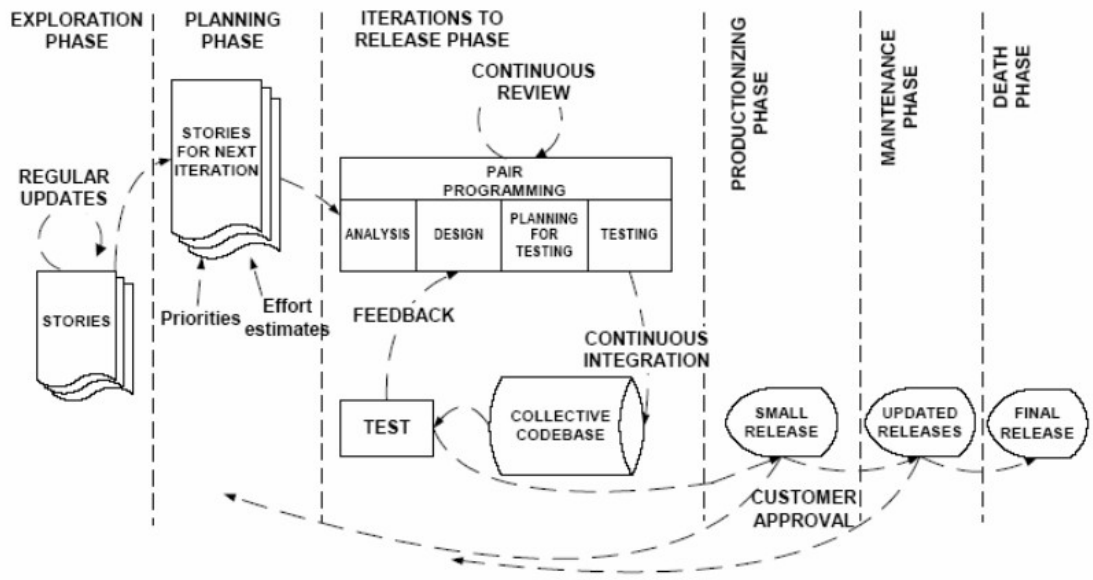


Ilustración 3 (Awad, 2005) Ciclo de vida XP. [Recuperado de] A Comparison between Agile and Traditional Software Development Methodologies

En el grafico se detalla el ciclo de vida y el proceso de desarrollo que se basa en la idea de pequeñas iteraciones y constantes actualizaciones de los requerimientos. Otra metodología empleada en la actualidad es SCRUM esta es una metodología iterativa que divide su ciclo de vida en sprints donde la comunicación con el cliente final y entre el equipo es primordial durante todo el proceso (Galiano, 2016).

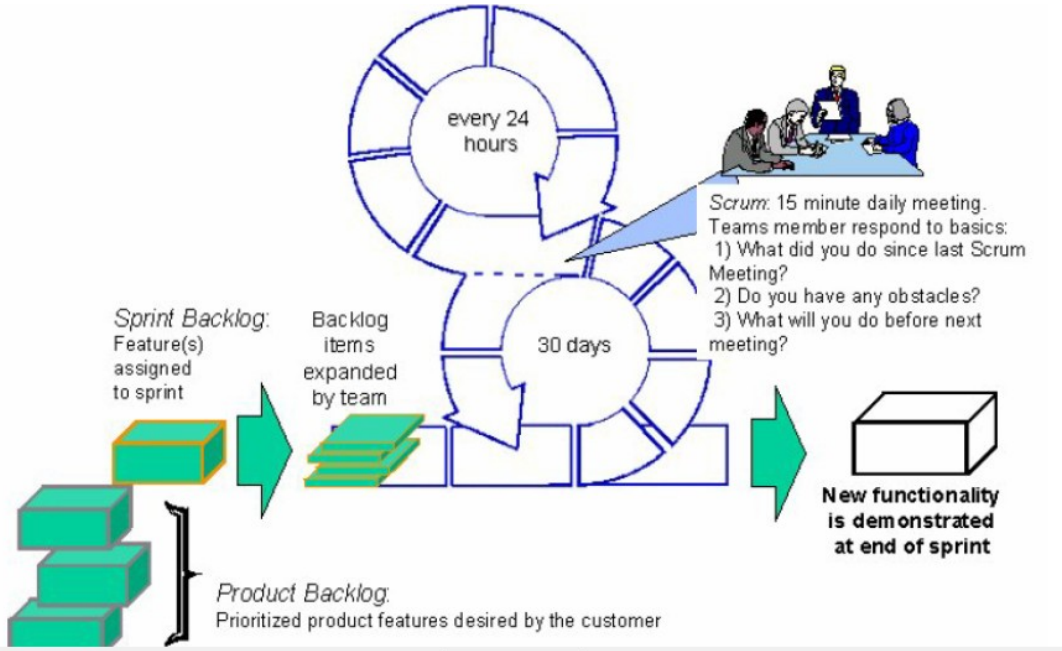


Ilustración 4 (Awad, 2005) Proceso Scrum. [Recuperado de] A Comparison between Agile and Traditional Software Development Methodologies.

La comunicación es sumamente importante para la implementación de esta metodología por lo que el tener reuniones tanto entre equipo y con el cliente brinda el desarrollo adecuado. Esta metodología es útil para proyectos con incertidumbre alta.

1.5.3 ENTORNOS PARA EL DESARROLLO MÓVIL

Existen diversos entornos que nos permiten desarrollar aplicaciones móviles para la plataforma Android a continuación, se detallarán algunos de los entornos de desarrollo para aplicaciones Android más conocidos y usados.

Java para desarrollo Android.

Con Java es posible desarrollar aplicaciones para Android en sistemas Windows, Linux y MAC OS X. Para programar usando Java, se debe tener instalado JDK (Java Development Kit) de Oracle la versión 1.7 o 1.8 son las más recomendables si se quiere desarrollar para la última versión de Android hasta la fecha (Android Oreo).
Existe un IDE (Integrated Development Environment) oficial para Android que se llama Android Studio y se lo puede obtener de la página oficial de Android Studio:

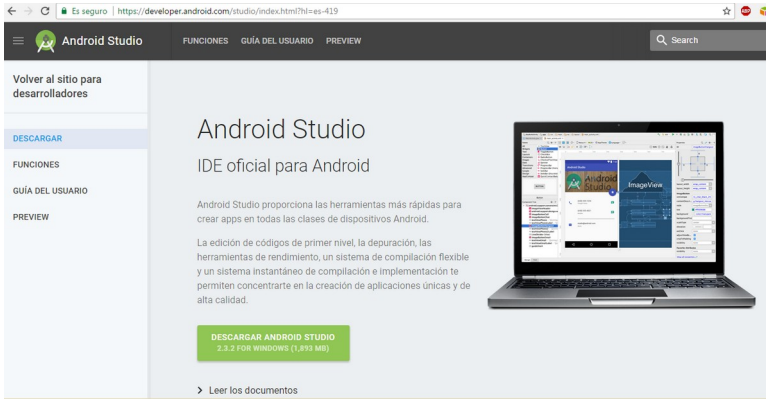


Ilustración 5 Google(2017).Página Oficial de descarga del IDE Android Studio. [Figura]. Recuperado de https://developer.android.com/studio.

Android Studio ofrece las siguientes funciones que permiten aumentar la productividad en el proceso durante la compilación de aplicaciones para Android:

- Compilación de los proyectos mediante Gradle.
- Un emulador rápido con varias funciones.
- Instant Run para aplicar cambios mientras la app se ejecuta.
- Herramientas para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones em los diferentes dispositivos android, etc.
- Permite agregar código C++ y es compatible con NDK.
- Soporte incorporado para Google Cloud Platform. (Google Inc., 2017)

Xamarin, C#:

Xamarin permite crear aplicaciones nativas de Android usando los mismos controles UI que usamos en Java, con la flexibilidad y elegancia de un lenguaje moderno que es C#, el poder de .NET y nos proporciona dos IDE's Xamarin Studio y Visual Studio. (Xamarin, 2017)

La descarga y el proceso de instalación para los diferentes sistemas operativos se encuentran detallados en su página oficial:

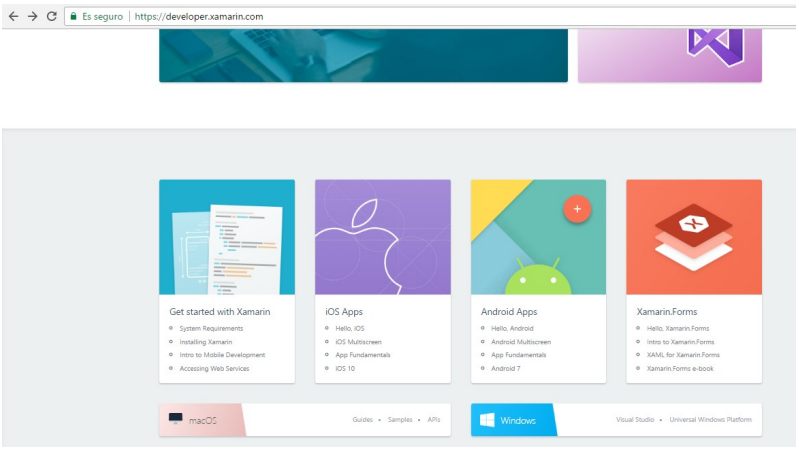


Ilustración 6 Xamarin(2017).Página Oficial de Xamarin. [Figura]. Recuperado de https://developer.xamarin.com

Algunas características de Xamarin se detallan a continuación:

- Las aplicaciones creadas mediante el entorno Xamarin contienen componentes de interfaz estándar y nativos de Android.
- Poseen acceso a API nativo ya que sus aplicaciones tienen acceso a todo el espectro de funcionalidad que expone el dispositivo, incluyendo el modo multiventana de Android y ARKit.
- Brinda un rendimiento nativo debido a que se aprovecha la aceleración de hardware específica de la plataforma y se compilan para tener un rendimiento nativo.
  (Microsoft, 2018)

Apache Córdova:

Es una plataforma híbrida para el desarrollo de aplicaciones usando tecnologías web como HTML5,Javascript y CSS, es de código abierto se basa en estándares W3C Mobile y posee APIs JavaScript para acceder a las características de los dispositivos. (Rodríguez, M. ,2013).

Además, posee soporte para iOS, Android, Blackberry, Symbian, Palm, Windows Phone.

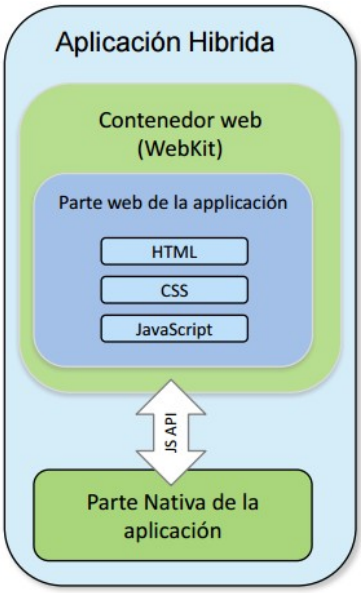


Ilustración 7 Rodríguez, M. (s.f). Esquema aplicación Híbrida. Recuperado de http://www.proyecto-cbc.org.pe/admin/recursos/publicaciones/4-Definicion-de-una-arquitectura-para-aplicaciones-moviles.pdf

Este tipo de aplicaciones híbridas contempla:

- Lo mejor del mundo web y nativo
- Estándares web abiertos
- Corre dentro de un contenedor web (Webkit) al ser encapsulada
- Acceso completo a funciones nativas con JavaScript.
- El código es el mismo en todas las plataformas.

(Rodríguez, s.f)

Python para desarrollo de aplicaciones en Android (Framework Kivy):

Kivy es un framework de Python para el desarrollo rápido de aplicaciones que hacen uso de interfaces de usuario para aplicaciones multi-touch. Es multiplataforma se ejecutan en Linux, Windows, OS X, iOS y Android. Puede soportar el mismo código escrito en todas las plataformas soportadas. (Kivy, 2017)

Es 100% libre y está bajo licencia MIT y se lo puede descargar de su página oficial:

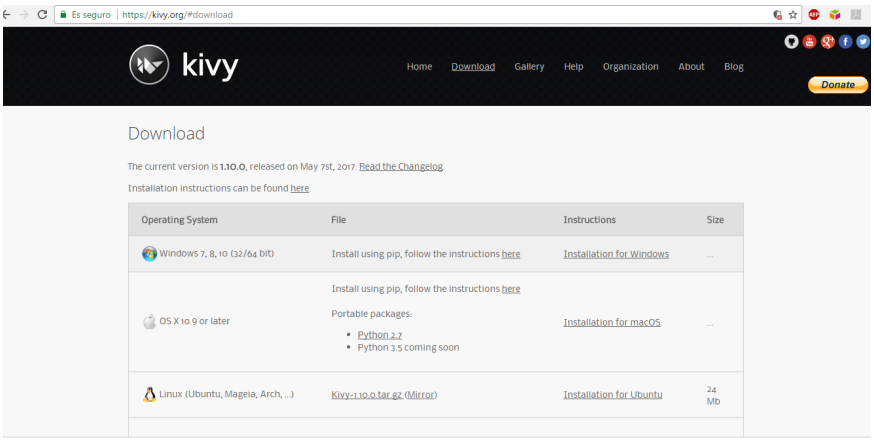


Ilustración 8 Kivy(2017). Página Oficial Framework Kivy. [Figura]. Recuperado de https://kivy.org

CAPITULO 2 ESTADO DE LA CUESTION

2.1 FORMULACIÓN DEL PROBLEMA

En los últimos años, la biodiversidad del planeta ha sido un aspecto fundamental en el ámbito de la investigación, con el fin de priorizar la conservación y reducir el riesgo de extinción de algunas especies. (Cisneros Barreiro & Silva Saltos, 2018)

En el Ecuador existen alrededor de 591 especies de ranas, por lo que, a nuestro país se le puede considerar como un lugar donde la anfibio fauna es muy numerosa (Anfibios del Ecuador, 2017). Actualmente, en la página web Anfibios del Ecuador, se tiene colecciones biológicas de tipo multimedia que incluyen aproximadamente 13600 fotos y 670 grabaciones de cantos. (Cisneros Barreiro & Silva Saltos, 2018).

El problema que se ha encontrado en base a experiencias comentadas en la facultad de Biología de la PUCE es el siguiente: escuchar y analizar los cantos grabados de forma manual, para identificar las especies, consume mucho tiempo y esfuerzo. Por lo que, la formulación de la pregunta de dicho problema es: ¿se reducirán el tiempo y el esfuerzo para analizar cantos de ranas grabados con el aplicativo móvil a desarrollar? (Cisneros Barreiro & Silva Saltos, 2018).

2.2 BÚSQUEDA O SELECCIÓN DE LAS FUENTES DE ESTUDIO

El criterio para realizar la búsqueda de fuentes de estudio fue basado en un proceso iterativo e incremental, es decir, la búsqueda se la realiza a partir de una fuente encontrada, luego se encuentran más fuentes hasta que se tenga un grupo de fuentes de estudio aceptable y definitivo. Las fuentes de estudio fueron obtenidas del Internet: de la fuente de investigación llamada SpringerLink y de páginas oficiales de cada una de las herramientas. (Cisneros Barreiro & Silva Saltos, 2018).

El criterio de selección de dichas fuentes fue basado en el análisis del resumen de cada fuente encontrada, es decir, se revisó brevemente cada artículo, publicación o enlace web, con el fin de determinar si la información encontrada aportaba al desarrollo del problema planteado anteriormente y, si la información mostraba herramientas o sistemas relacionados con nuestro proyecto. En el buscador, en este caso, el más utilizado es Google, se buscó con frases claves, por ejemplo: “reconocimiento de especies de animales”, “Shazam para animales”, ya que, con dichas frases, se desplegaban varias publicaciones científicas y sistemas/aplicaciones móviles relacionadas al tema. (Cisneros Barreiro & Silva Saltos, 2018)

Además, se utilizó fuentes formales e informales para la búsqueda de información. Entre las fuentes formales se pueden nombrar a publicaciones o artículos científicos y entre las fuentes informales se pueden señalar a noticias en páginas de internet confiables y sitios web oficiales, que posteriormente se indica detalladamente. (Cisneros Barreiro & Silva Saltos, 2018)

2.3 EXTRACCIÓN Y CODIFICACIÓN DE LOS ESTUDIOS SELECCIONADOS

Una vez que se seleccionaron las fuentes de estudio bajo los criterios establecidos, se procede a definir variables moderadoras, las mismas que pueden ser de tres tipos: las sustantivas, las metodológicas y las extrínsecas. (Cisneros Barreiro & Silva Saltos, 2018)

• Como variables sustantivas se codificaron:

- Tipo de aplicación (escritorio – web – móvil): Se refiere a si el sistema o aplicación debe estar instalada en un ordenador para funcionar y si se la puede utilizar en una computadora o en un teléfono celular. (Cisneros Barreiro & Silva Saltos, 2018)
- Disponibilidad de la aplicación (online – offline): Se refiere a si el sistema o aplicación funciona bajo una conexión de internet o si se es independiente de ello. (Cisneros Barreiro & Silva Saltos, 2018)

- Costo de la aplicación (gratuita – no gratuita): Se refiere a si el sistema o aplicación tiene un costo o es gratis. (Cisneros Barreiro & Silva Saltos, 2018)

• Como variables metodológicas se codificaron:

- Tamaño del audio que se analiza: Se refiere al tiempo de duración de la grabación que se quiere analizar en el sistema o aplicación. (Cisneros Barreiro & Silva Saltos, 2018)
- Tipo de audio que se analiza: Se refiere a la extensión del archivo de audio a analizar. Puede ser: .WAV, .mp3, .mid, .aac, entre otros. (Cisneros Barreiro & Silva Saltos, 2018)
- Tiempo de duración del análisis: Se refiere al tiempo que le toma al sistema o aplicación realizar el análisis de la grabación. (Cisneros Barreiro & Silva Saltos, 2018)
- Forma en que se presentan resultados (escrita – gráfica – escrita/gráfica): Se refiere a cómo el sistema o aplicación presenta por pantalla los resultados para que el usuario los visualice. (Cisneros Barreiro & Silva Saltos, 2018)
- Evaluación de la aplicación (si tiene – no tiene): Se refiere a si el sistema o aplicación se ha desarrollado y se han hecho pruebas en él, o solo ha sido descrito, pero aún no se ha desarrollado. (Cisneros Barreiro & Silva Saltos, 2018)

• Como variables extrínsecas se codificaron:

- Estatus de publicación de la fuente de estudio (publicado – no publicado): Se refiere a si el sistema o aplicación fue publicado o mencionado en alguna revista indexada. (Cisneros Barreiro & Silva Saltos, 2018)
- Idioma disponible: Se refiere al idioma que se presenta en el sistema o aplicación. (Cisneros Barreiro & Silva Saltos, 2018)

Una vez definidas las variables moderadoras, se procede a extraer la información relevante para la revisión sistemática, basándose en las variables anteriormente descritas. Dicha información se presenta a continuación, con su respectiva referencia bibliográfica: (Cisneros Barreiro & Silva Saltos, 2018)

Lista de herramientas (sistemas/aplicaciones) analizadas:

1. ChirpOMatic (App móvil encontrada en el Internet)
2. Warblr (App móvil encontrada en el Internet)
3. Bird Song Id (App móvil encontrada en el Internet)

4. SongSleuth (App móvil encontrada en el Internet)
5. Cicada Hunt (App móvil encontrada en el Internet)

1. ChirpOMatic

ChirpOMatic es una aplicación desarrollada por la empresa iSpiny Software, cuya sede se encuentra ubicada en Cardiff, Reino Unido. Esta empresa tiene como principal enfoque el desarrollo de aplicaciones relacionadas con la fauna que no afecte su estilo de vida. (Cisneros Barreiro & Silva Saltos, 2018)
La aplicación cuenta además de las siguientes funcionalidades:

- Uso de la app sin internet.
- Subir y compartir las grabaciones del usuario.
- Interfaz bastante amigable.
- Bird Safe mode.

La Dr. Hilary Lind miembro de iSpiny, explica que el reproducir el sonido de las grabaciones causan efectos en las aves, por lo que esta aplicación soluciona este problema añadiendo la funcionalidad Bird Safe Mode en donde para escuchar las grabaciones el usuario tiene que escucharlas a manera de llamada telefónica, así no se molesta a las aves en su territorio o sus actividades. (Cisneros Barreiro & Silva Saltos, 2018).
Actualmente la aplicación se encuentra desarrollada orientada a dispositivos Apple como son el iPad, iPhone y Apple Watch, no se encontró ninguna referencia de que se encuentre disponible para otras plataformas ni de parte del fabricante la intención de desarrollarla. El idioma de la aplicación es inglés y es importante desatacar que está disponible para ciertas regiones con sus especies como Estados Unidos, Reino Unido y Holanda. El costo de la aplicación publicado en la tienda virtual de Apple (App Store) para el año 2017 es de 3,99 dólares americanos en el caso del Reino unido y Holanda está disponible en 3,99 euros. (Cisneros Barreiro & Silva Saltos, 2018)
El aplicativo no describe en sus portales información oficial acerca del tipo de audio que grabaría ni del tiempo de duración del análisis, pero la información es bastante clara en cuanto al tamaño. El audio tendrá una duración de 12 segundos. Los resultados se presentan a manera de lista donde se muestran las mejores coincidencias en un orden descendente. El usuario puede acceder a la especie donde recibirá una breve reseña de la especie en particular. (Cisneros Barreiro & Silva Saltos, 2018)

La aplicación se encuentra disponible solamente en inglés y la empresa desarrolladora no ha emitido ningún estudio público acerca del tema por lo que es desconocido el estatus del estudio actual en este caso aplicativo en particular. (Cisneros Barreiro & Silva Saltos, 2018).

2. **Warblr**

Esta aplicación que se genera a partir de un proyecto de ciencia ciudadano fue desarrollada por Florence Wilkinson quien es una productora de cine y Dan Stowell quien es un científico graduado en Wueen Mary University of London, él es el responsable del aprendizaje de máquina, el front end de la aplicación fue creado por la empresa Glastonbridge, su director creativo Thellius Zamprogno, y el logo fue diseñado por Emily Kerr. (Cisneros Barreiro & Silva Saltos, 2018).

La aplicación se encuentra disponible tanto como para dispositivos Android y dispositivos iOS, en el caso de la versión para dispositivos Android tiene un costo de 4,54 dólares americanos en la Play Store (tienda de aplicaciones Android) y en el caso de iOS tiene un costo de 4,99 euros, es importante mencionar que contiene especies solamente del Reino Unido. (Cisneros Barreiro & Silva Saltos, 2018)

El tiempo de la grabación es de 10 segundos y depende de las condiciones de la grabación el tiempo de análisis. Se ha puesto a prueba la app y se obtuvo que bajo condiciones idóneas su fidelidad es del 95%. Muestra los resultados del análisis a manera de lista con el porcentaje de semejanza hacia cada tipo de ave. (Cisneros Barreiro & Silva Saltos, 2018)

La aplicación está disponible en idioma inglés, el último estudio realizado por Dan se evidencia en el artículo publicado denominado “Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning”, que data del año 2014. (Cisneros Barreiro & Silva Saltos, 2018).

3. **Bird Song Id**

La aplicación Bird Song Id fue creada por profesionales ecologistas de la empresa Sunbird Images con sede en Alemania, con el fin de permitir el reconocimiento automático de pájaros únicamente con su canto, según sus pruebas su tasa de éxito en este proceso fue del 85%. (Cisneros Barreiro & Silva Saltos, 2018).

Además de la función de reconocimiento la aplicación ofrece:

- Catálogo para listar y filtrar especies de pájaros. (Sunbird Images, 2017).
- Brinda toda la información de cada especie de ave como nombre, nombre científico, fotografía, descripción, familia, etc. (Sunbird Images, 2017).
- Identificación manual para el usuario que permite identificar cantos de pájaros mediante la estimación de parámetros simples como, por ejemplo, si el canto tiene melodía o no, características del tono y el volumen sean estos bajos o altos, entre otros. (Sunbird Images, 2017).
- Permite tener una lista de cantos personales. (Sunbird Images, 2017).

Es una aplicación offline móvil, es decir que el reconocimiento automático se lleva a cabo en el dispositivo, no es necesario poseer conexión a internet, se encuentra disponible para iOS y para Android en sus respectivas tiendas, el aplicativo es de pago, para iPhone tiene un costo de $3.99 y en Android de $5.96. Permite reconocer el canto de aves de diferentes especies además de contar con un catálogo de especies que permite tener información de la misma como su nombre común, nombre científico, fotografía, descripción, su familia, entre otros datos. (Cisneros Barreiro & Silva Saltos, 2018).

El reconocimiento es offline ya que todo el proceso se lleva a cabo en el mismo dispositivo, todos los cantos y datos necesarios para el reconocimiento se descargan en el momento que se adquiere la aplicación. (Cisneros Barreiro & Silva Saltos, 2018)

La aplicación móvil permite grabar 30 segundos el canto de un pájaro. Una vez que el usuario termine de grabar esta se almacena temporalmente en el dispositivo, cabe recalcar que toda grabación puede ser agregada a la lista personal de cada usuario, con la grabación terminada se presiona el botón para reconocer automáticamente el canto grabado, en ese momento la aplicación procesa la grabación para poder comparar con las demás especies ya registradas en la aplicación y esta demora unos pocos segundos (2-3 segundos) en reconocer y desplegar una lista con los puntajes de acierto para cada especie además de una gráfica en la que se visualiza el grafico de la señal de audio de la grabación. Por último, en la lista de puntajes se puede observar el nombre de la especie, fotografía, descripción y además se puede escuchar el canto de cada especie en la lista para poder comparar con el canto que el usuario grabó. (Cisneros Barreiro & Silva Saltos, 2018).

La fuente de estudio no se encuentra publicada ni disponible en ninguna revista o repositorio científico, así como el código fuente de la aplicación, por lo que la misma únicamente puede ser modificada por los desarrolladores que pertenecen a la empresa Sunbird Images, esto se debe a que el aplicativo móvil es de pago. Además, la aplicación se encuentra disponible en el idioma inglés. (Cisneros Barreiro & Silva Saltos, 2018)

4. SongSleuth

Es una aplicación que permite de manera interactiva y fácil aprender a observar aves en cualquier lugar dentro de los Estados Unidos. Fue desarrollada por Wildlife Acoustics y en colaboración del experto e ilustrador de renombre mundial David Sibley. (Cisneros Barreiro & Silva Saltos, 2018)

Los algoritmos que utiliza la aplicación para la identificación de los cantos son el resultado de más de una década de investigación y experiencia en el diseño profesional de grabadoras bioacústicas y software. (Cisneros Barreiro & Silva Saltos, 2018).

La aplicación cuenta con estas funcionalidades:

- Proceso de Identificación automática de un total de 200 especies de aves más encontradas en Estados Unidos. (Wildlife Acoustics, 2017).
- Guardar, visualizar, reproducir y compartir grabaciones. (Wildlife Acoustics, 2017).
- Cuenta con una completa colección de grabaciones de ejemplo para todas las especies que incluye la aplicación. (Cisneros Barreiro & Silva Saltos, 2018)
- Biblioteca de especies que contiene además de las grabaciones, imágenes e información científica obtenida de ilustraciones del experto David Sibley. (Cisneros Barreiro & Silva Saltos, 2018)
- Permite visualizar las ubicaciones en las que se realizaron las grabaciones mediante un mapa. (Wildlife Acoustics, 2017).

La aplicación se encuentra disponible actualmente para iOS y se la puede descargar con un costo de $9.99. (Cisneros Barreiro & Silva Saltos, 2018)

Permite grabar solo unos pocos segundos el canto de un ave para su posterior procesamiento, para reconocer automáticamente el canto, el usuario oprime el botón de grabar y mientras el graba el canto la aplicación muestra en tiempo real una gráfica de la señal de audio y además el espectrograma correspondiente a la misma , una vez que el usuario termine de grabar solo

debe volver a oprimir el botón de grabación y la aplicación procesara la grabación y la comparará con las especies guardadas en la base de datos local obteniendo una lista con las 3 especies más parecidas. Este último proceso la aplicación lo realiza solo en unos pocos segundos. (Cisneros Barreiro & Silva Saltos, 2018).

La aplicación también permite escuchar el canto grabado y de la lista de las especies más parecidas, compartir la grabación, visualizar la gráfica de la grabación y el espectrograma, agregar una nota a la grabación, e incluso agregarla a la biblioteca de cantos. (Cisneros Barreiro & Silva Saltos, 2018)

La fuente de estudio no se encuentra publicada ni disponible en ninguna revista o repositorio por ser una aplicación comercial de pago, así como el código fuente de la aplicación, por lo que la misma únicamente puede ser modificada por la empresa Wildlife Acoustics. (Cisneros Barreiro & Silva Saltos, 2018).

5. Cicada Hunt

Esta nueva aplicación funciona de manera similar que Shazam. La función del software es identificar cortos clips para empatar con la llamada de la cigarra de New Forrest. El proyecto se enfocó en los millones de personas que anualmente visitan New Forrest. La idea es que los turistas paseen por el bosque con la aplicación ejecutándose en segundo plano en sus teléfonos inteligentes. Una vez que la aplicación ha reconocido la llamada del insecto, suena una alerta para decirle al usuario que grabe un breve clip de sonido que luego se puede enviar por correo electrónico a los investigadores, que luego crean un mapa de calor de la propagación del insecto. (Cisneros Barreiro & Silva Saltos, 2018)

Deseaban aprovechase el poder de la multitud por tanto la cual permite a cualquier persona mediante el uso de un celular inteligente identificar las llamadas únicas de animales en especial aves, pájaros y saltamontes que se encuentran en peligro y a la vez tratar de fijarse en la biodiversidad de ellos. En estudios realizados en Eslovenia la aplicación fácilmente pudo detectar la cigarra. (Cisneros Barreiro & Silva Saltos, 2018) (Marks, 2013).

Esta aplicación desarrollada es de tipo móvil disponible para Android e iOS. Esta aplicación funciona en segundo plano en cualquier dispositivo inteligente, es decir, tiene una disponibilidad offline. No tiene ningún costo. La aplicación necesita de un audio breve de alrededor de 15 segundos (tiene un funcionamiento similar a Shazam). (Cisneros Barreiro & Silva Saltos, 2018). El

tiempo de análisis es de alrededor de 5 segundos y la forma en que se presentan los resultados es escrita y gráfica. Esta aplicación fue diseñada por Alex Rogers y un equipo de la Universidad de Southampton. Este articulo esta publicado en la revista New Scientist el 31 de agosto del 2013. Actualmente se encuentra disponible en inglés. (Cisneros Barreiro & Silva Saltos, 2018).

Cuadro Resumen

	Características	Ventajas	Desventajas
Arbimon II	Sistema web que funciona online, recibe audios hasta de 100 minutos en formato WAV y presenta resultados del análisis en espectogramas.	El sistema es web y presenta los espectogramas del audio.	La disponibilidad solo es online, es decir, para utilizarlo debe haber forzozamente una conexión a internet.
ChirpOMatic	Sistema móvil que funciona offline, recibe audios hasta de 12 segundos y presenta resultados del análisis proporcionando una lista de posibles especies relacionadas.	El sistema es móvil, la disponibilidad es offline y presenta una lista de varios resultados asociados al audio cargado.	El tamaño del audio que recibe para analizar.
Warblr	Sistema móvil, recibe audios hasta de 10 segundos y presenta resultados del análisis proporcionando una lista de posibles especies relacionadas con el respectivo porcentaje.	El sistema es móvil y presenta una lista de varios resultados relacionados, mostrando un porcentaje, al audio cargado.	El tamaño del audio que recibe para analizar.
Bird Song Id	Sistema móvil que funciona offline, recibe audios hasta de 30 segundos y presenta resultados del análisis con fotos con su respectiva descripción.	El sistema es móvil, la disponibilidad es offline y presenta fotos con descripción de las posibles especies relacionadas al audio cargado.	El tamaño del audio que recibe para analizar.
SongSleuth	Sistema móvil, recibe audios de pocos segundos de grabación y presenta resultados del análisis de varias formas: listas, gráficas y espectogramas.	El sistema es móvil y presenta varias formas de presentar resultados.	El tamaño del audio que recibe para analizar.
Cicada Hunt	Sistema móvil que funciona offline, recibe audios hasta de 15 segundos y presenta resultados del análisis de forma escrita con gráficas.	El sistema es móvil, la disponibilidad es offline y presenta resultados en forma escrita y gráfica.	El tamaño del audio que recibe para analizar.

Tabla 1 Cuadro resumen estado del arte

2.4 ANÁLISIS DEL ESTUDIO

Una vez obtenida la información de cada una de las aplicaciones móviles existentes en el mercado, analizaremos las variables más trascendentes para nuestro estudio:

- Encontramos que el 50% de las aplicaciones tienen un costo, algunas bajo suscripción y otras bajo una compra única de la licencia de la aplicación, esto contrasta con un 25% de aplicaciones que no tienen costo y otro 25% que no especifican su licenciamiento.

- En el mercado actualmente, las aplicaciones móviles investigadas el 63% nos permiten el análisis y reconocimientos de audio de una duración de un minuto o menos, mientas el 37% restante tiene un tamaño variable.

- De las aplicaciones investigadas tan solo el 27% utiliza el formato WAV para el análisis de audio mientras el 73% se reparte entre MP3, WMA y OGG.

- El 28% de las aplicaciones tarda en promedio de 2 a 3 segundos en realizar el análisis del audio, mientras que el 62% especifica que depende del tamaño del audio, el entorno de análisis y el formato del audio.

- El 87% de las aplicaciones utilizan una forma mixta (grafica, texto) para presentar los resultados de su proceso de análisis, mientras el 13% solo utiliza gráficos del tipo espectrograma para mostrar los resultados.

- El 62% de los aplicativos se encuentras disponibles a usuarios finales, es decir se encuentran en tiendas de aplicaciones móviles.

- El dominio del inglés en estas aplicaciones es abismal ya que un 89% de las aplicaciones investigadas se encuentran en dicho idioma, mientras solo el 11% se encuentran en español.

2.5 RESULTADOS, CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Tabla de Resultados

	Tipo	Disponibilidad	Costo	Tamaño del audio	Tipo de audio
ChirpOMatic	No especifica	Lista de especies (Escrita)	No	No publicado	Inglés
WarbIr	Depende del tamaño del audio	Lista con porcentajes (Escrita)	No	Publicado	Inglés
Bird Song Id	2 a 3 segundos	Descripción con fotos (Escrita/Gráfica)	No	No publicado	Inglés
SongSleuth	Pocos segundos	Lista, gráficas y en espectogramas (Escrita/Gráfica)	No	No publicado	Inglés
Cicada Hunt	5 segundos	Escrita/Gráfica	No	Publicado	Inglés

Tabla 2 Resultados estado del arte

Conclusiones

- Existen aplicativos móviles, pero ninguno realiza el reconocimiento de cantos de ranas por lo que este proyecto es innovador para el mercador mundial.

- El aplicativo móvil no tendrá costo alguno, ya que es un proyecto netamente de investigación y académico, por lo que será de gran ayuda para la mayoría de las personas.

- La forma de presentación de los resultados en el aplicativo móvil será de forma escrita y gráfica. Existirá una gráfica en donde se presenten las probabilidades de la especie de rana comparada con el análisis del audio cargado y otra gráfica en la que se señalará la parte de la frecuencia del audio en dónde se encuentra el canto de determinada especie de rana. Además, habrá otros gráficos mostrando las segmentaciones del audio cargado al portal.

- El idioma del aplicativo móvil será en español, hasta el momento, facilitando enormemente a aquellas personas que no están familiarizadas con otros idiomas y, obviamente, a las personas que se encuentran en Ecuador

Futuras líneas de investigación

A lo largo del estudio de la situación actual se generó un conocimiento de futuras líneas de investigación que pueden llegar a ser de mucha utilidad para investigadores que ocupen esta aplicación:

- A futuro se puede desarrollar un servicio web que puede permitir el analizar nuevas especies y generar modelos para alimentar automáticamente

el aplicativo móvil siendo la actualización de contenido de una forma más dinámica.

- Seria importante el permitir a investigadores poder cargar sus propias especies y tener una administración de cantos de manera privada y parametrizada para cada usuario permitiéndole utilizar el aplicativo como repositorio de cantos especializado.

- Podemos pensar en el permitir el uso de más formatos de audio para su procesamiento y cargar archivos de mayor duración ya que por lo general estas grabaciones son extensas.

- Finalmente, podemos desarrollar un aplicativo multiplataforma donde el fabricante del dispositivo no sea un limitante y se pueda utilizar desde una mayor cantidad de dispositivos.

CAPITULO 3 ANÁLISIS DE LA TECNOLOGÍA A DESARROLLAR

El objetivo de este capítulo es el entender y describir tanto el producto a construir como la manera en la que vamos a construirlo. Explicaremos las funcionalidades que debe cumplir el aplicativo y con base en eso describiremos los siguientes puntos.

3.1 DESCRIPCIÓN GENERAL DE LA APLICACIÓN A DESARROLLAR

3.1.1 DESCRIPCIÓN DEL APLICATIVO MÓVIL

Con base en los lineamientos del proyecto de investigación se tiene que la aplicación debe cumplir con ciertos requerimientos de funcionalidad e infraestructura. La aplicación con la finalidad de facilitar el proceso de recolección y análisis de muestras auditivas tiene que estar orientada hacia un dispositivo que brinde las facilidades para ser llevado al campo. El aplicativo tiene que permitir a los usuarios recolectar muestras de audio de una manera sencilla y almacenar este audio. Una vez recogida la muestra el aplicativo analizará los audios y recogerá las características que serán comparadas con modelos previamente cargados. El proceso que automatizará el aplicativo será el siguiente:

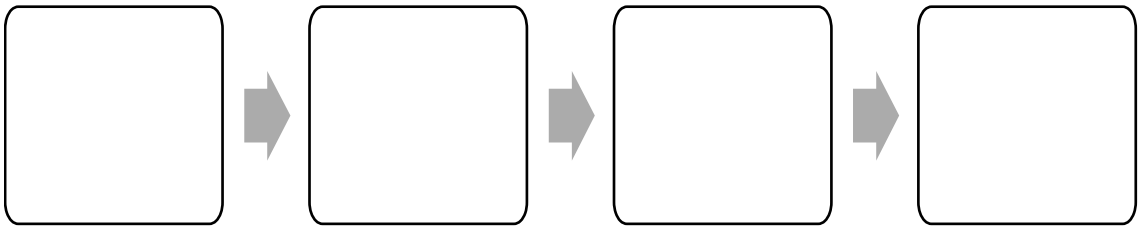


Ilustración 9 Proceso general de reconocimiento de cantos de ranas.

La fase de recolección de la muestra corresponde a la obtención del audio mediante un sensor de audio y a la correcta segmentación del mismo, posteriormente se extraen las características propias del audio recolectado y se genera un modelo matemático. Durante la fase de comparación de los modelos se evalúan las características del audio recolectado y se los coteja con los modelos previamente cargados al aplicativo, esta comparación debe permitirnos visualizar la lista de las especies encontradas en el canto después del reconocimiento.

3.1.2 DECRIPCIÓN DEL PROCESO DE RECONOCIMIENTO

Preparación DataSet – Selección y procesamiento de data:

La preparación del Dataset se lo consigue a partir de archivos de audio en los que se encuentran horas de grabación los cuales pasan por un procesamiento de señal simple para obtener un espectrograma de la señal , luego de esto se segmenta cada archivo de audio en bloques de 6 segundos con el espectro generado de cada bloque se procede a la comparación con otros segmentos de cantos de las especies ya obtenidos previamente , y finalmente de acuerdo a la semejanza con los diferentes cantos se selecciona o se rechaza el segmento de la señal de audio.

Como resultado se obtiene una tabla que contiene todos los segmentos válidos para el entrenamiento de los modelos de reconocimiento.

A continuación se presenta el proceso de la preparación del Dataset :

Ilustración 10 Preparación del DataSet de un canto de rana.

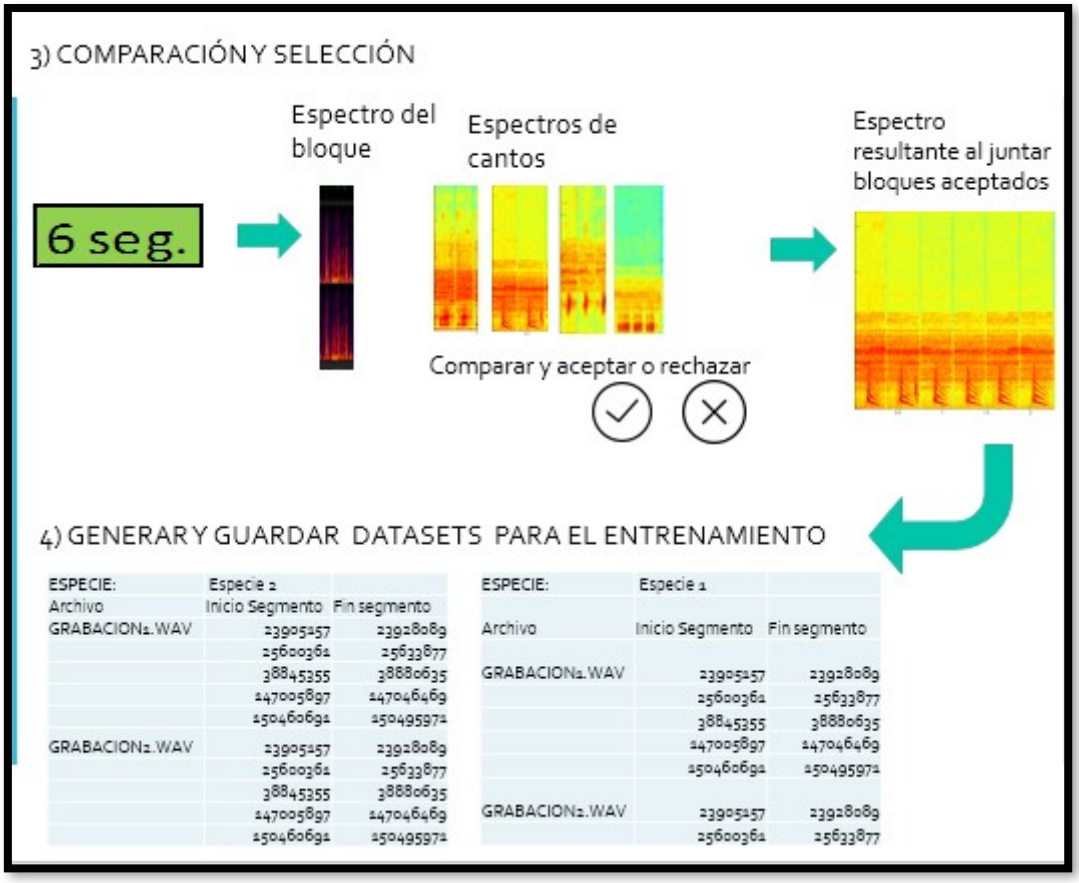


Ilustración 11 Proceso de comparación y selección de cantos de ranas.

Extracción de Características:

Con los datasets de las diferentes especies se procede en esta etapa del reconocimiento a unir los bloques obteniendo así una señal continua en la que se puede apreciar los mejores cantos de las especies, en dicha señal obtenida cada bloque de 6 segundos se subdivide en bloques de 20 milisegundos y son estos los que se someten a la extracción de características por medio del MFCC que nos entrega un vector de coeficientes cepstrales los parámetros para esta extracción de características son:

- Ventana mfcc: 20 milisegundos
- Número coeficientes cepstrales: 20

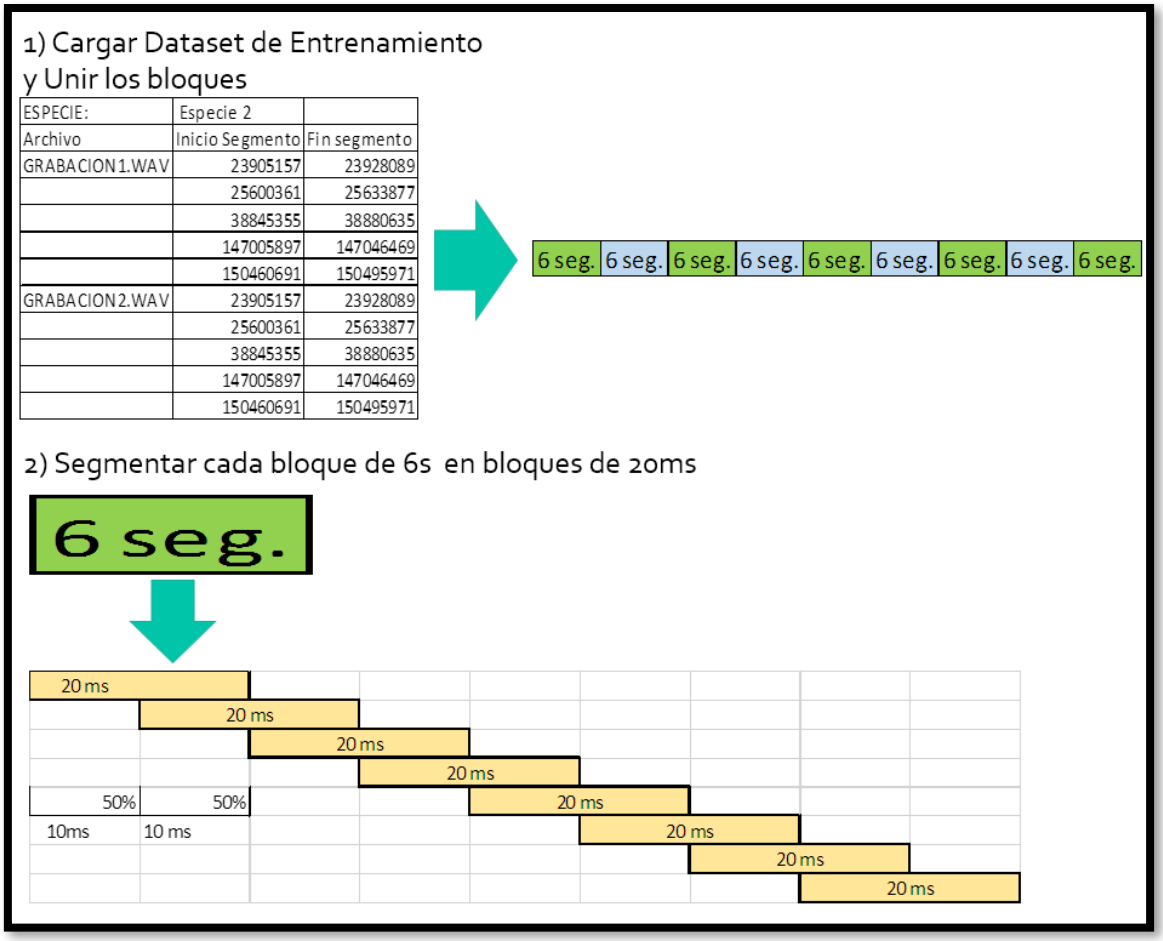


Ilustración 12 Preparación del DataSet.

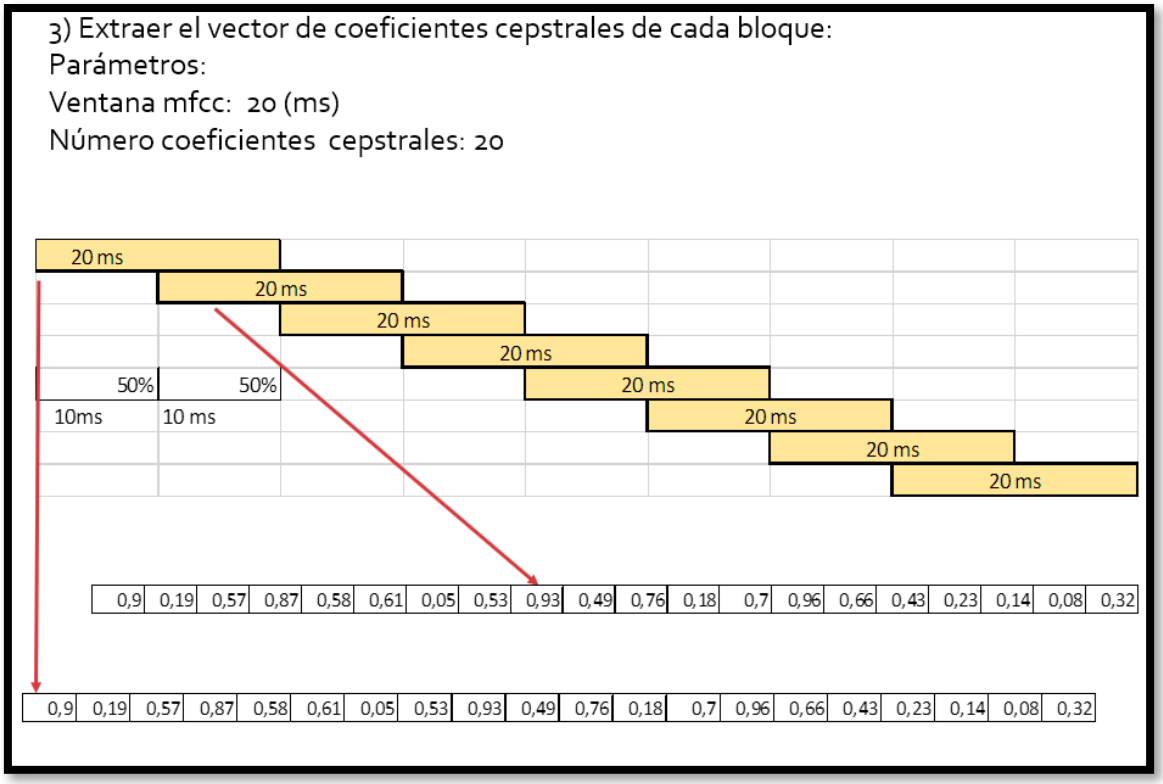


Ilustración 13 Extracción de características.

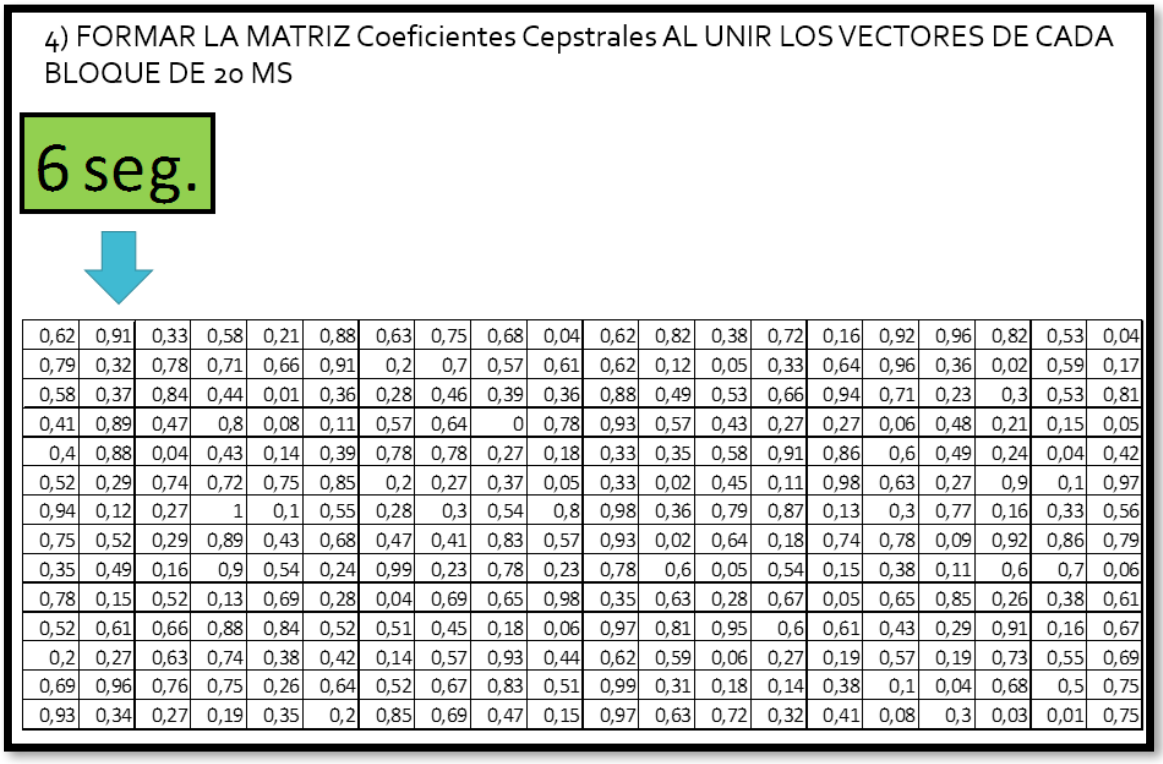


Ilustración 14 Matriz de coeficientes ceptrales.

Entrenamiento y evaluación de modelos

Para el entrenamiento y evaluación de modelos existen diferentes alternativas como las siguientes:

- GMM (Gaussian Mixture Models): Modelos Gaussianos Mixtos
- SVM (Support Vector Machine): Maquinas de Vectores de Soporte
- ANN (Artificial Neural Networks): Redes Neuronales.

GMM: Modelos Gaussianos Mixtos:

- Modelo probabilístico. (Scikit-learn, 2017)
- Asume todos los puntos de datos son generados de una mezcla de un numero de Distribuciones gaussianas con parámetros desconocidos. (Scikit-learn, 2017)
- Se compone por varias funciones de densidad de probabilidad. (Scikit-learn, 2017)
- Encontrar una aproximación o estimación a partir de sus componentes encontrando un acomodamiento de los datos que contienen las componentes. (Scikit-learn, 2017)

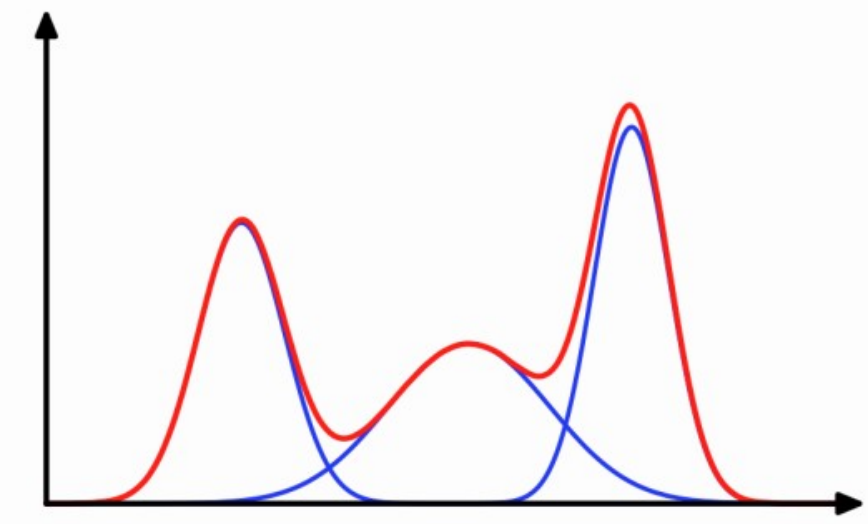


Ilustración 15 Diagrama GMM.

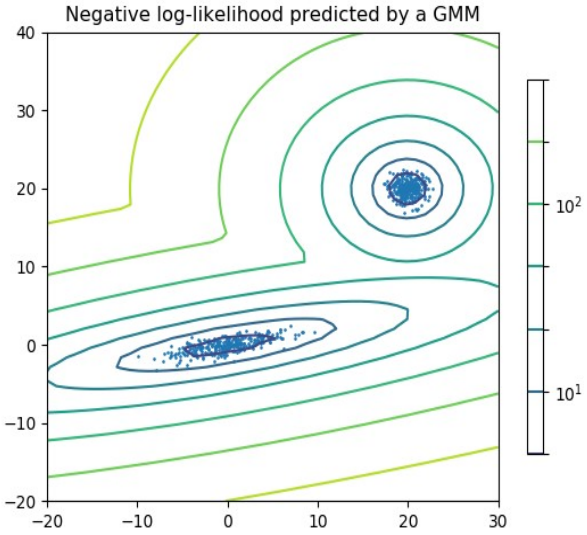


Ilustración 16 Ejemplo GMM.

SVM: Máquinas de Vector Soporte

- Pensados para resolver problemas de clasificación.(C. Cortes & V.Vapnik, 1995)
- Se usan también para resolver problemas como regresión, agrupamientos, multiclasificación.(C. Cortes & V.Vapnik, 1995)
- Usan separadores lineales o hiperplanos en el espacio.(C. Cortes & V.Vapnik, 1995)
- Generan un modelo en base a un conjunto de puntos que es capaz de predecir si un punto pertenece a una categoría o a otra. (C. Cortes & V.Vapnik, 1995)

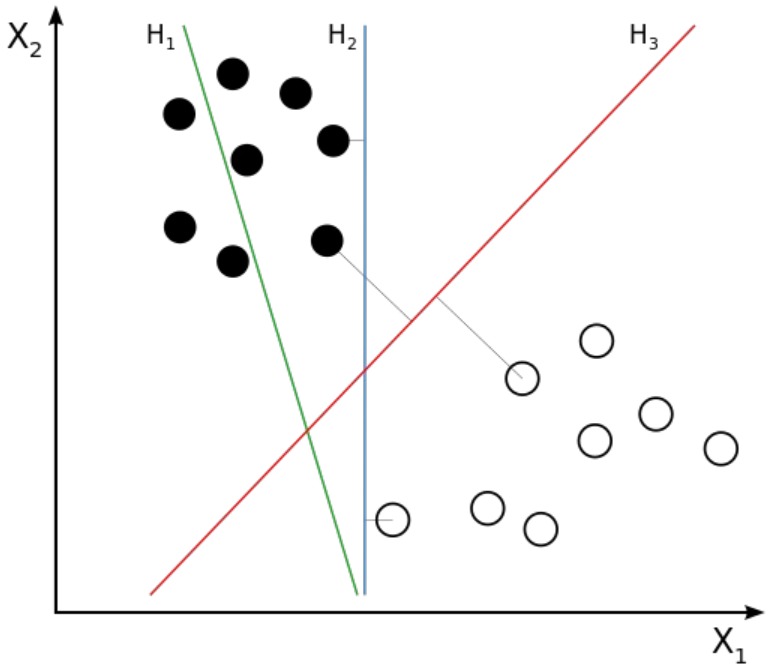


Ilustración 17 Ejemplo SVM

Redes Neuronales

- Es un modelo matemático basado en las redes neuronales biológicas.(Matich, 2001)
- Son usadas como técnicas estadísticas para modelar complejas relaciones entre entradas y salidas de un sistema. (Matich, 2001)
- Son un método de aprendizaje automatizado. (Matich, 2001)
- Diseño de la red es en base a la importancia de las características.(Matich, 2001)

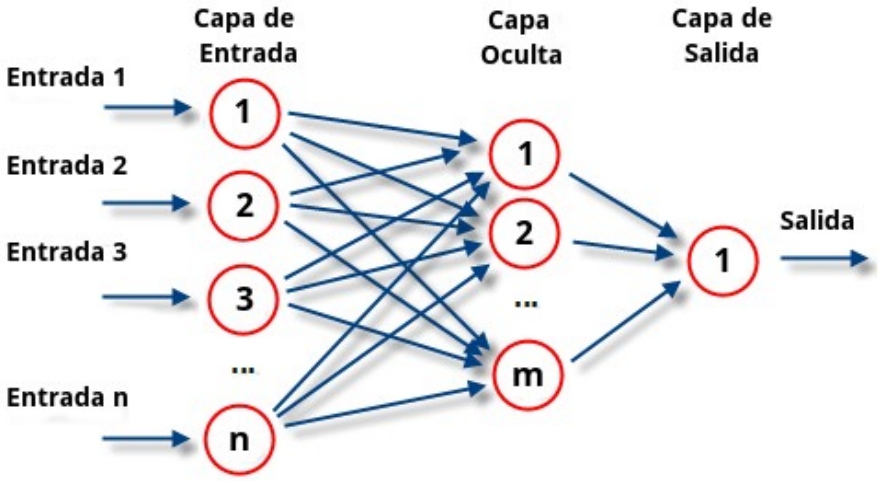


Ilustración 18 Diagrama de red neuronal.

3.2 DESCRIPCIÓN DE LA TECNOLOGÍA A EMPLEAR

Con base en la investigación realizada y después de analizar las necesidades generales del proyecto se ha seleccionado un grupo de tecnologías que nos apoyaran a satisfacer con las mismas.

Debido a que el proyecto tiene que ser orientado hacia un dispositivo fácil de manejar y transportar por los investigadores se ha decidido que los dispositivos objetivos serán teléfonos inteligentes ya que en la actualidad gracias a los estudios realizados tenemos que alrededor de tres millones de personas cuentan con acceso a teléfonos inteligentes (Instituto Nacional de Estadistica y Censo, 2016).

A nivel mundial el porcentaje de venta de dispositivos móviles Android tiene una tendencia al alza por los que apuntalando hacia el futuro definiremos los dispositivos Móviles Inteligentes Android serán los dispositivos encargados de albergar nuestra

aplicación y facilitar los recursos necesarios como; micrófono, pantalla, capacidad de procesamiento, entre otros.

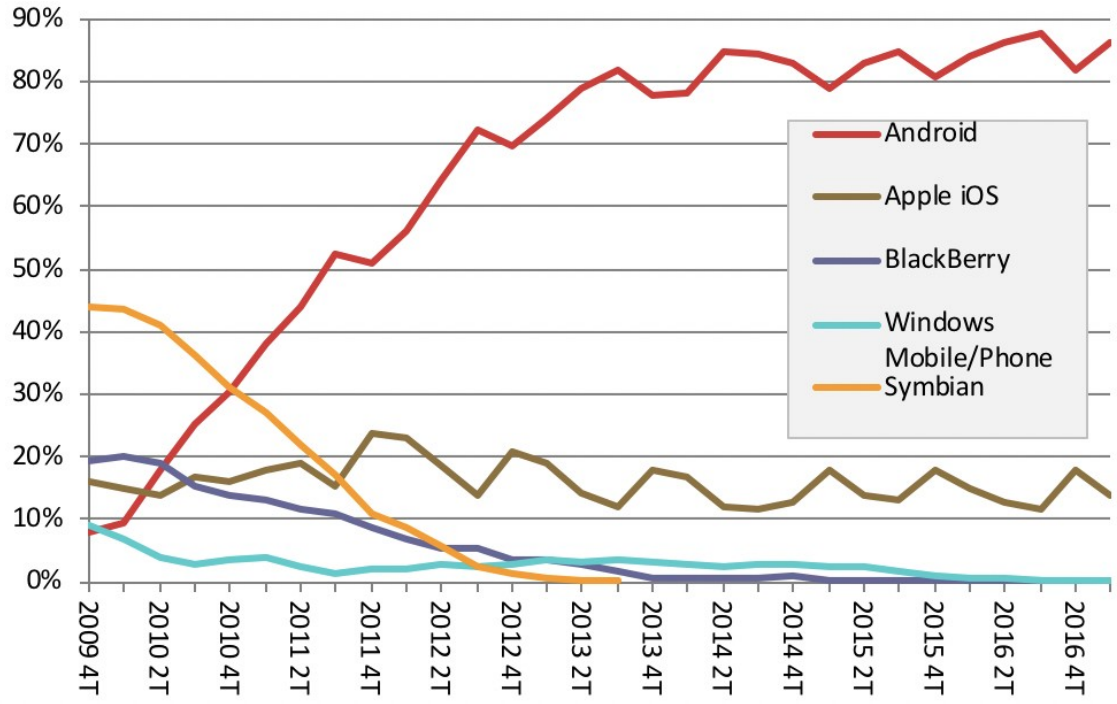


Ilustración 19 Gartner Group. (2016.) Porcentaje de teléfonos inteligentes vendidos en todo el mundo hasta el último trimestre de 2016. [Recuperado de] http://www.androidcurso.com/index.php

3.3 DESCRIPCIÓN DE LA METODOLOGIA DE DESARROLLO A IMPLEMENTAR

En la actualidad las metodologías de desarrollo ágil son las más usadas en la implementación de proyectos de software como aplicaciones móviles y sitios web debido a que ofrece ventajas sobre otras metodologías tales como ofrecer una rápida respuesta a los cambios, permite seguimiento de los proyectos gracias al proceso iterativo, por estas razones y tomando en cuenta que lo que se va a desarrollar es una aplicación móvil se decidió utilizar la metodología de desarrollo SCRUM.(Higuera, Mario, Camelo, & Cediel, 2014)

Para implementar la metodología Scrum se deben tomar en cuenta varios aspectos propios de la metodología los cuales son roles, artefactos y reuniones.

Roles:

- Scrum Master: es el encargado de apoyar al grupo de desarrollo para que todos los objetivos sean cumplidos en su totalidad. (Higuera et al., 2014)

- Product Owner: es el interesado en el producto a desarrollar y proporciona todos los recursos para llevar a cabo el proyecto. (Higuera et al., 2014) (Higuera et al., 2014)
- Stakeholders: son los expertos de negocio y son los que participan en la definición de requerimientos ya que conocen de manera específica todos los detalles del negocio. (Higuera et al., 2014)
- Equipo de desarrollo o desarrolladores: son los que cumplen con las tareas asignadas por el Scrum Master. (Higuera et al., 2014)

Artefactos:

- Product Backlog: consiste en una lista de funcionalidad en las que se especifica en forma de historias de usuario y especificadas en la primera reunión. (Villarreal, 2008)
- Sprint Backlog: Es un subconjunto de elementos del Product backlog que se seleccionan para terminarlos en el sprint. (Villarreal, 2008)
- Incremento: Es el resultado del sprint y consiste en un entregable terminado es decir que debe estar listo para ser utilizado, se puede decir que es una nueva versión del producto totalmente funcional y además es la suma de los elementos del Product Backlog completados en un sprint más los incrementos de los anteriores Sprints.(Villarreal, 2008)

Reuniones:

- Reunión Previa: Todo inicia con la definición de tareas que comienza con una lista de requerimientos que se ordena por importancia, en esta lista se definen los requisitos, su estimación de inicio y el solicitante, todo esto se registra en un documento denominado Product Backlog que será utilizado por todo el equipo SCRUM mientras se desarrolle el proyecto.(Higuera et al., 2014)
- Planificación del Sprint: Antes del inicio de cada sprint se determina cual es el trabajo y cuáles son los objetivos a cumplir por lo que se escogen varios elementos del Product backlog para descomponerlos en tareas y dar como resultado el Sprint backlog.(Villarreal, 2008)

- Daily Scrum (Seguimiento del sprint): Es una breve reunión diaria que toma por lo general 15 minutos en la misma se repasa el avance de cada tarea y el trabajo que se planifico para la jornada. Cada participante de la reunión debe responder 3 preguntas simples y estas son:(Mariño & Alfonzo, 2014)

- ¿Que hice desde la reunión anterior?
- ¿Qué voy a hacer de aquí a la próxima reunión ¿
- ¿Qué impedimentos se deben solventar para poder realizar el trabajo?

- Revisión del sprint: Análisis y revisión del incremento generado en el que solo se tomara en cuenta aquello que cumpla con la definición de terminado lo demás pasará al Product backlog, se realiza una presentación a los involucrados de todo lo que se logró terminar en el sprint, además se les informa lo que se va a seguir haciendo y en caso de que no se haya terminado algo en el sprint también se lo debe informar. En pocas palabras es una presentación de resultados y del software totalmente funcional. (Schwaber & Beedle, 2001)

- Retrospectiva del Sprint: Aquí se evalúa el proceso para ver si se aplicó bien o mal la metodología todo esto con el fin de mejorar. En esta reunión se deben realizar las siguientes preguntas:(Schwaber & Beedle, 2001)

  - ¿Que hicimos bien?
  - ¿Qué podemos mejorar
  - ¿Qué debemos dejar de hacer?

Herramientas:

- Tablero Scrum: Es un tablero en el que se plasman las tareas del Product Backlog, y sirve para verificar el estado en que se encuentran las mismas , este tablero, y se asemeja al tablero Kanban, en el que se identifican las columnas de Tareas por hacer (TO-DO), las que están en ejecución (IN-PROCESS), en prueba (TEST) y las terminadas (DONE), y, algo que se puede destacar es que en este tablero se puede identificar los cuellos de botella o represamiento de tareas. (Higuera et al., 2014)

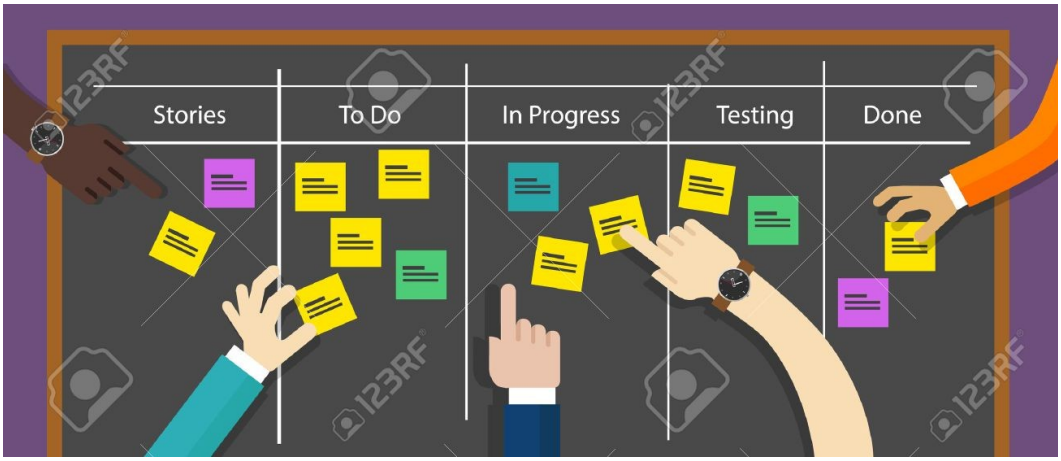


Ilustración 20 123F. (2018.). Software metodología ágil gestión ilustración desarrollo de proyectos tablero scrum. [Recuperado de] https://es.123rf.com/photo_38752664_software-metodolog%C3%ADa-%C3%A1gil-gesti%C3%B3n-ilustraci%C3%B3n-desarrollo-de-proyectos-tablero-sc

- Burndown chart: El gráfico de burndown de tareas del sprint tiene por objetivo ayudar al equipo en la monitorización de su progreso y ser el indicador principal que informa las posibilidades de alcanzar su compromiso al finalizar el sprint. El formato clásico requiere que el equipo estime la duración de cada tarea en horas de forma diaria. El burndown deberá completarse de forma tal que grafica cuántas horas de trabajo restan para concluir el sprint. (Higuera et al., 2014).

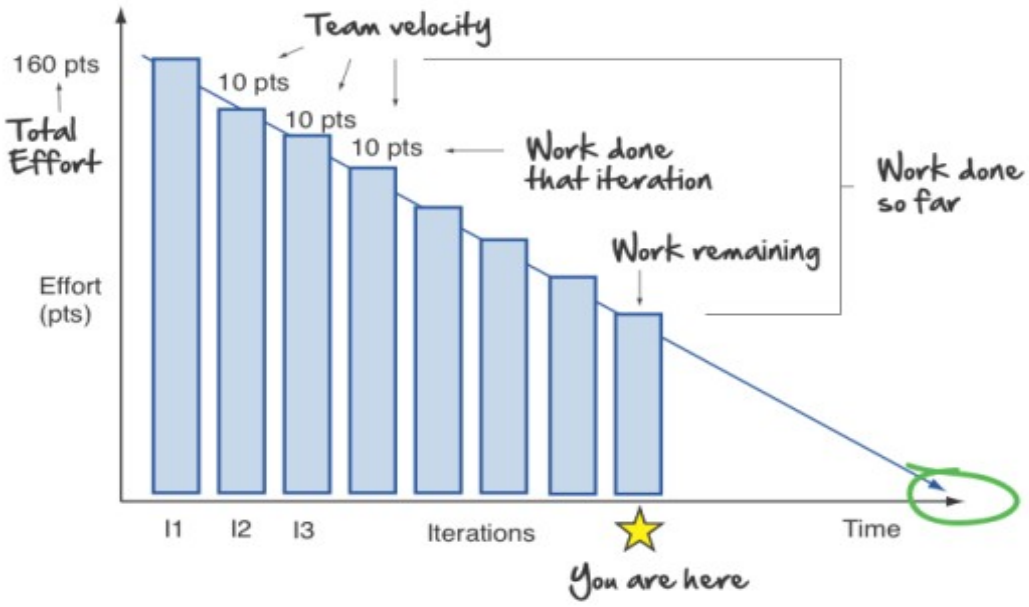


Ilustración 21 Agile in a nutshell. (2018.). Burndown Charts . [Recuperado de] http://www.agilenutshell.com/burndown

CAPITULO 4 SPRINT 0

El objetivo de este capítulo es el iniciar el proceso de desarrollo aplicando la metodología SCRUM, definiremos estrategias y los roles de cada integrante para conseguir finalizar el proceso con éxito. Nos permitirá a todos entrar en sintonía con el grupo e identificarnos con las metas que deseamos alcanzar. El entregable de este Sprint se denomina “Product Backlog”.

4.1 ESPECIFICACIÓN DE ROLES

En este apartado se detalla los responsables de desempeñar cada rol propuesto por la metodología. Mantendremos su nomenclatura en inglés para evitar confusiones con la teoría.

Rol	Responsable
Product Owner	Ing. Damián Nicolalde
Scrum Master	Andrés Páez
Development Team	Edison Gudiño
Development Team	Andrés Páez
Stakeholder	Ing. Andrés Estrella

Tabla 3 Tabla de roles para SCRUM.

4.2 DECLARACIÓN DEL PRODUCTO

Esta será una aplicación para los investigadores o personas naturales que necesiten recopilar audios de ranas y determinar de qué especie se trata. Esto facilitará el proceso actual que se realiza manualmente, ya que podrá acceder a respuestas con un alto porcentaje de certeza instantáneamente.

4.3 CREACIÓN DEL ÁREA DE TRABAJO

Para el desarrollo de la metodología es importante el tener un área de trabajo adecuada donde los miembros puedan comunicarse sin ningún tipo de barreras. (Al., 2014)

Utilizaremos una herramienta en línea que nos ayudara a emular una pizarra con post-its, por lo que basaremos nuestro desarrollo en un ambiente online más que físico.

La herramienta donde crearemos nuestro tablero de actividades es denominada “Trello”, esta es un software que nos permite administrar diferentes tipos de proyectos o actividades (Altassian, 2018).

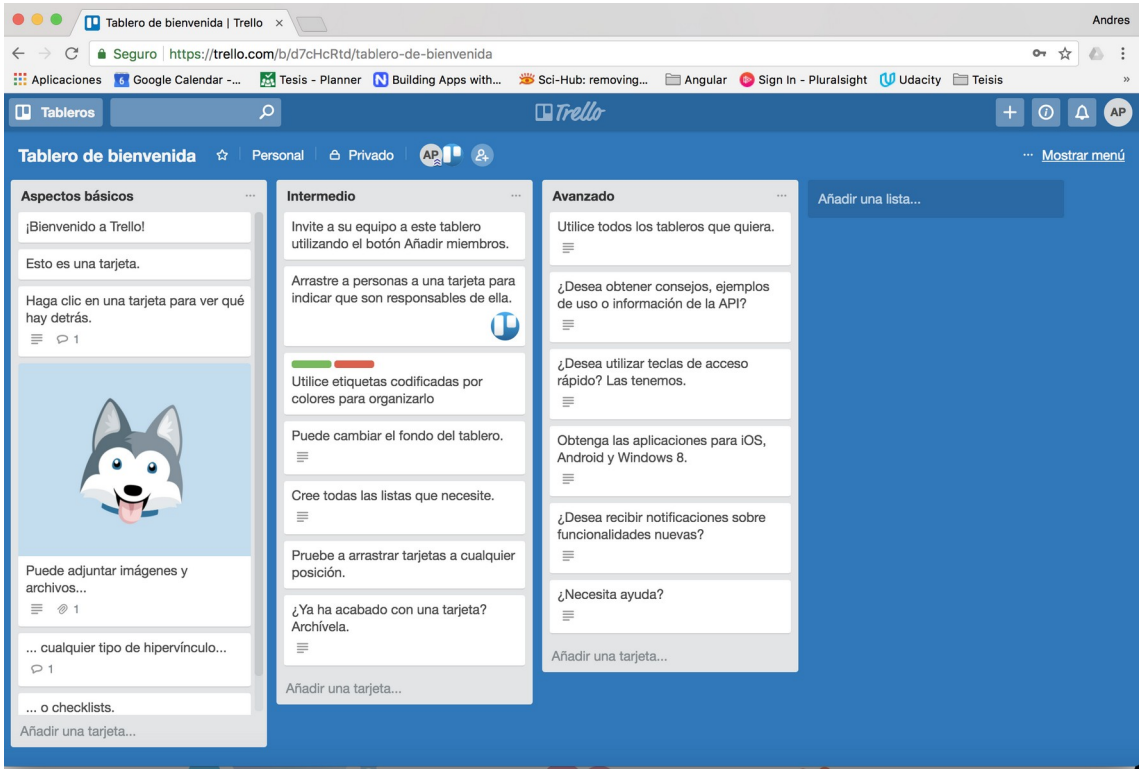


Ilustración 22 Tablero SCRUM.

Crearemos el área de trabajo bajo las especificaciones de SCRUM. Con 4 listas:

- Stories
- ToDo
- On Progress
- Testing

- Done

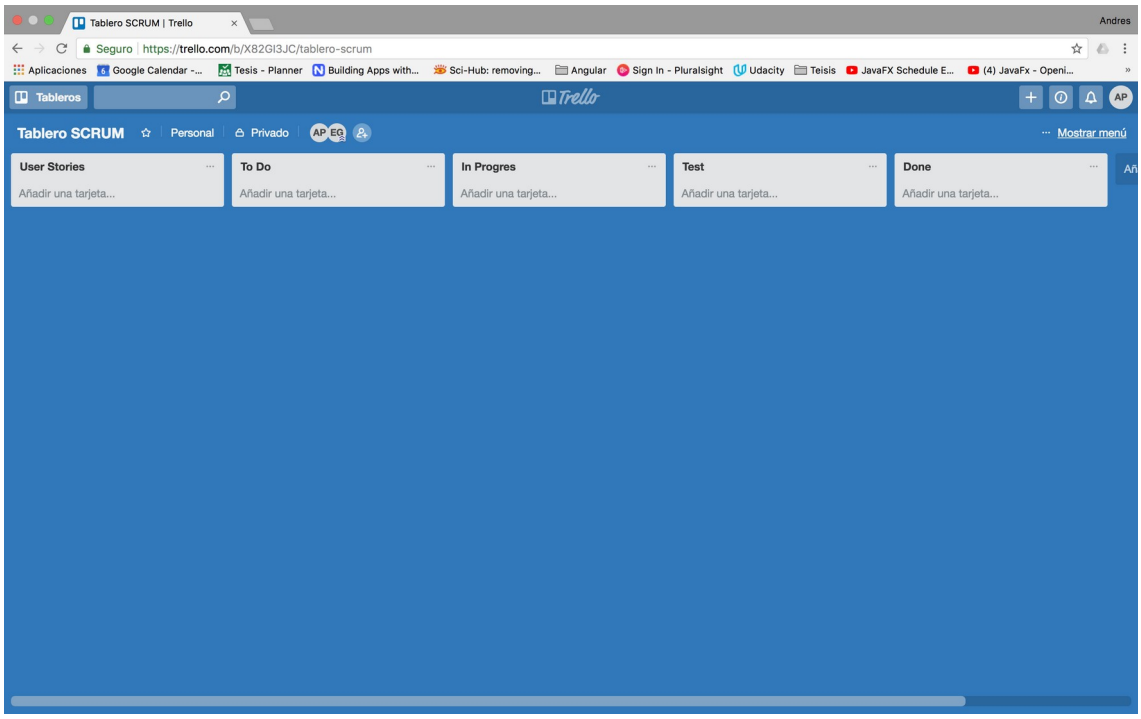


Ilustración 23 Tablero SCRUM para el proyecto.

Este marco de trabajo nos permitirá compartir ideas de una manera gráfica e instantánea que enriquecerá nuestro proceso de desarrollo a lo largo de todo el proyecto.

4.4 PRODUCT BACKLOG

Una vez realizada la reunión para determinar el Backlog se definió las siguientes historias de usuario:

PRODUCT BACKLOG PARA LA APLICACIÓN			
Tarea ID	**Historia**	**Estimado (Días)**	**Prioridad**
2	Como usuario, deseo poder obtener una lista de especies ordenada por la similitud del canto.	5	1
1	Como usuario, deseo poder grabar desde mi celular los cantos de una especie por un	5	2

	minuto o menos.		
3	Como usuario, deseo poder visualizar una lista de todas las especies encontradas en un archivo de audio.	5	3
6	Como usuario, deseo poder cargar un archivo de audio para su procesamiento.	5	4
5	Como usuario, deseo poder escuchar los cantos de las distintas especies.	5	5
4	Como usuario, deseo poder obtener más información (nombre científico, descripción, imagen) sobre las especies de ranas que se encuentren en la aplicación.	5	6

Tabla 4 Backlog del proyecto.

4.5 DEFINICIÓN DE FINALIZADO

La definición de finalizado es importante ya que nos permite revisar paso a paso los requisitos de un determinado trabajo para poder decir que se ha completado satisfactoriamente. (Rubin, 2013)
Definiremos nuestra propia lista para definir que un trabajo ha sido finalizado.

Definition Of Done
☐ **Se reviso el diseño.**
☐ **Código Completo** ☐ **El programa realizo el Build sin ningún error.** ☐ **Código inspeccionado** ☐ **Código esta comentado.**
☐ **Se actualizo la documentación.**
☐ **Pruebas** ☐ **Pruebas unitarias** ☐ **Pruebas de integración**
☐ **Aceptación del PO**
☐ **Funcional en dispositivo Android.**

Tabla 5 Definición de finalizado el Sprint.

4.6 DISEÑO

Con la finalidad de conseguir un entendimiento común entre los miembros del equipo se ha decidido la diagramación del sistema en general. Utilizaremos UML (Unified Modeling Language) que es el estándar para análisis y diseño orientado a objetos (Scott, 1999). Hemos decidido utilizar los siguientes diagramas para la descripción del sistema:

- Diagrama de casos de uso general.
- Diagrama de casos de uso a detalle.

- Diagrama de clases.

4.6.1 DIAGRAMA DE CASOS DE USO GENERAL

Los casos de uso son maneras en las que un usuario o actor puede interactuar con el sistema y sus funcionalidades (Scott, 1999). Nosotros realizaremos un diagrama de clases general que nos brindara un contexto genérico y más amplio de las funcionalidades del sistema para luego descomponerlas en diagramas de caso de uso con mayor detalle.

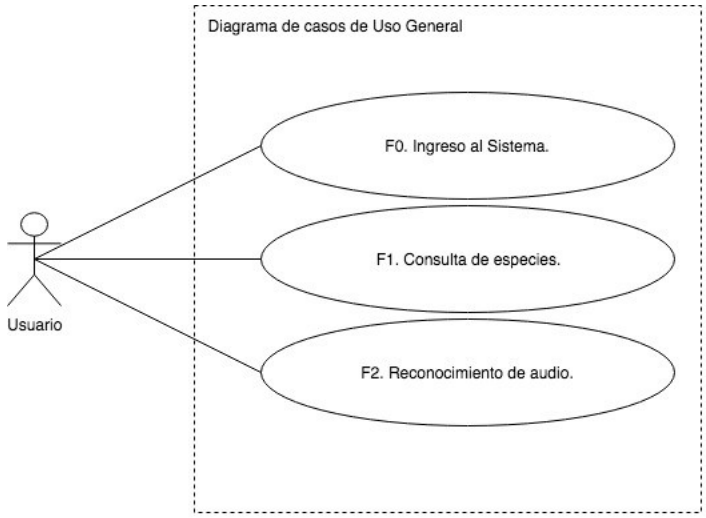


Ilustración 24 Diagrama de Casos de uso General.

4.6.2 DIAGRAMAS DE CASOS DE USO A DETALLE

Detallaremos cada uno de los casos de uso importantes para el proceso, así obtendremos un entendimiento común y mayor abstracción del problema.

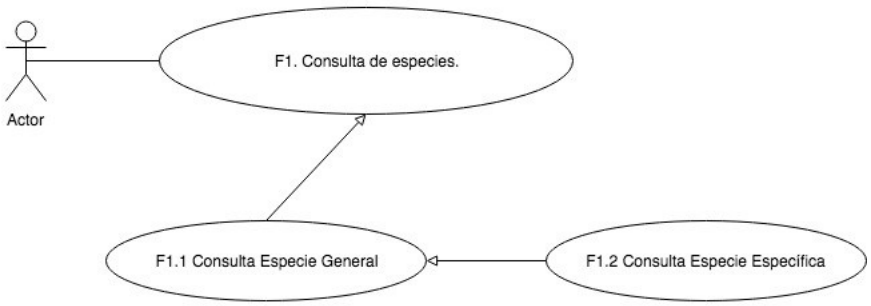


Ilustración 25 Diagrama de casos de uso a detalle.

Sistema:	Aplicación Reconocimiento de cantos de ranas
Caso de uso:	Consulta Especies General
Actor:	Usuario
Excepciones: 1. El actor selecciona “Explorar” 2. El sistema abre la activity de explorar especies. 3. El sistema se conecta con la base de datos. 4. El sistema carga los datos en la activity. (E1) 5. El sistema muestra los datos en forma de lista.	
Flujo Alterno: 4. Consultar con el administrador del sistema.	
Excepciones: **E1:** No hay conexión con la BDD.	**Soluciones:** Llamar al administrador del sistema.

Sistema:	Aplicación Reconocimiento de cantos de ranas
Caso de uso:	Consulta Especies Específica
Actores:	Usuario
Excepciones: 1. El actor selecciona “Explorar” 2. El sistema abre la activity de explorar especies. 3. El sistema se conecta con la base de datos. 4. El sistema carga los datos en la activity.(E1) 5. El sistema muestra los datos en forma de lista. 6. El actor selecciona una especie. 7. El sistema envía la especie al activity especie especifica. 8. El sistema carga los datos en la activity especie especifica. 9. El usuario visualiza los datos.	
Flujo Alterno: 4. Consultar con el administrador del sistema.	
Excepciones: **E1:** No hay conexión con la BDD.	**Soluciones:** Llamar al administrador del sistema.

Este caso de uso se enfoca en la obtención del audio y aplicación del algoritmo de reconocimiento de audio.

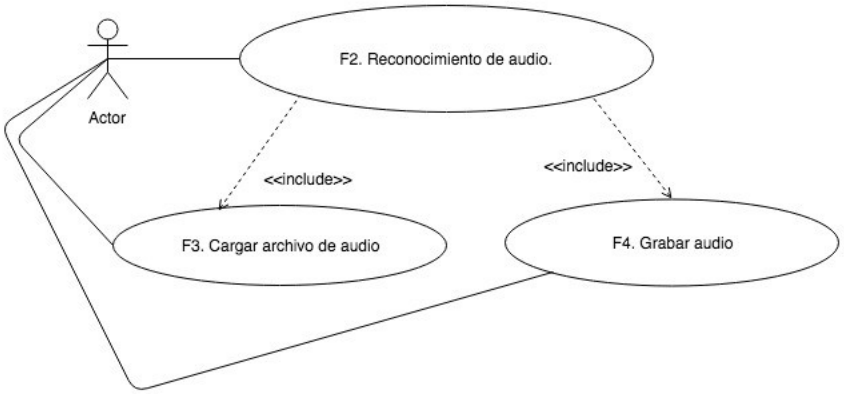


Ilustración 26 Diagrama de casos de uso a detalle.

Sistema:	Aplicación Reconocimiento de cantos de ranas
Caso de uso:	Reconocimiento de audio
Actores:	Usuario
Excepciones: 1. El actor selecciona “Reconocer” 2. El sistema abre la activity de reconocer especies. 3. El sistema se conecta con la base de datos. 4. El sistema carga los modelos de la base de datos. (E1) 5. El sistema muestra la opción de cargar audio o grabar audio. 6. El actor selecciona cargar audio o grabar audio. 7. El sistema analiza el audio ingresado aplicando el algoritmo de reconocimiento. 8. El sistema compara los resultados con los modelos cargados de la base de datos. 9. El sistema muestra los resultados de la puntuación de similitud y la especie a la que pertenece en la consola del activity de reconocimiento. 10. El usuario visualiza los datos y gráficas obtenidos posterior al reconocimiento.	
Flujo Alterno: 4. Consultar con el administrador del sistema.	
Excepciones: **E1:** No hay conexión con la BDD.	**Soluciones:** Llamar al administrador del sistema.

4.6.3 DIAGRAMA DE CLASES GENERAL

Este diagrama nos brindara una mejor abstracción al momento de construir un sistema orientado a objetos, nos mostrara un modelo de clases y las asociaciones que estas tienen entre sí. (Schneider et al., 2010)

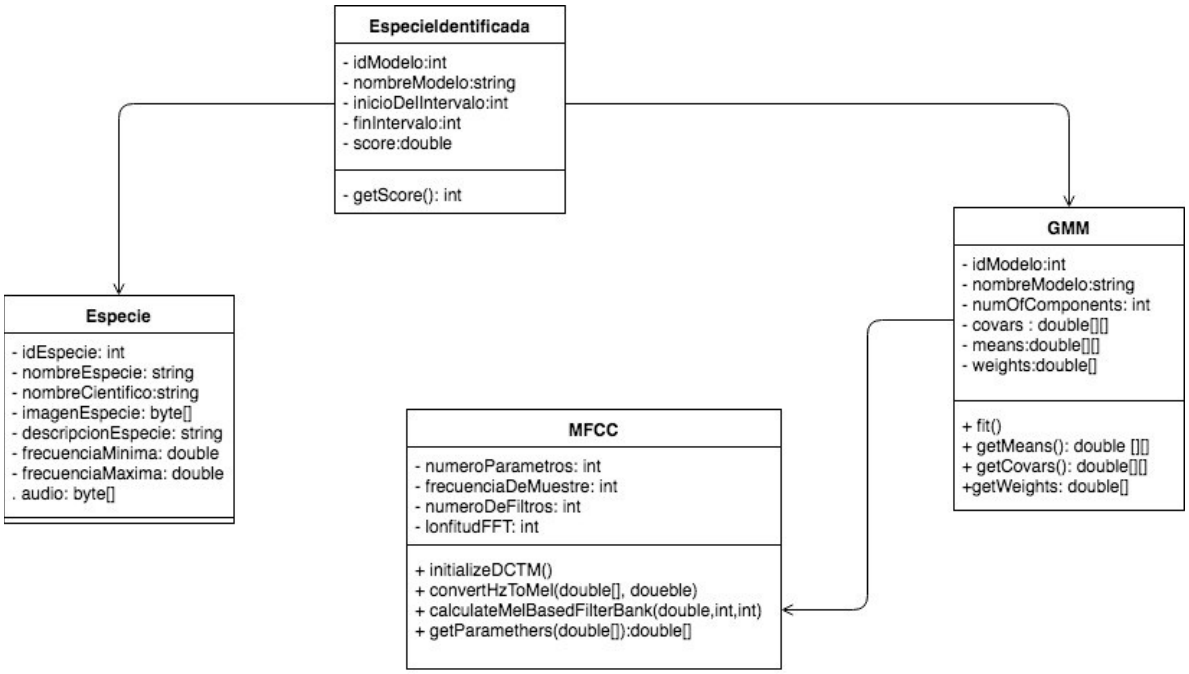


Ilustración 27 Diagrama de clases.

4.6.4 PROTOTIPO DE INTERFACES DE USUARIO

Interfaz principal al iniciar la aplicación, esta nos permitirá el cargar un archivo o grabar un archivo para su posterior análisis.

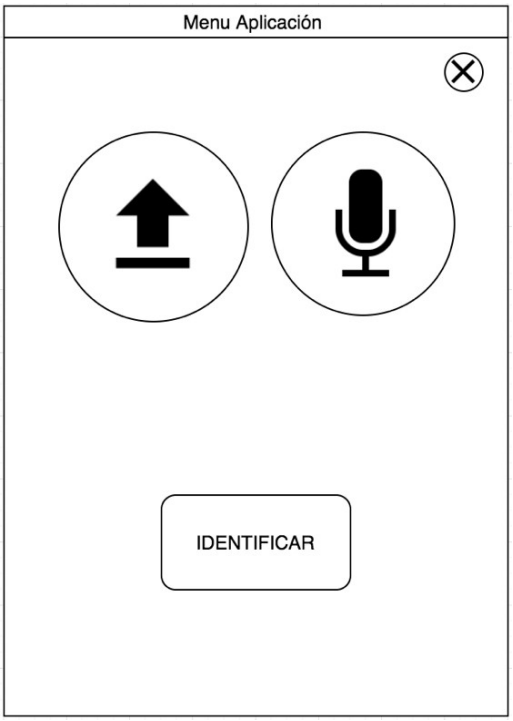


Ilustración 28 Interfaz principal.

El menú de la aplicación se ubicará en la parte de debajo de la pantalla, donde se presentarán las opciones para poder acceder a todas las funcionalidades de la aplicación.

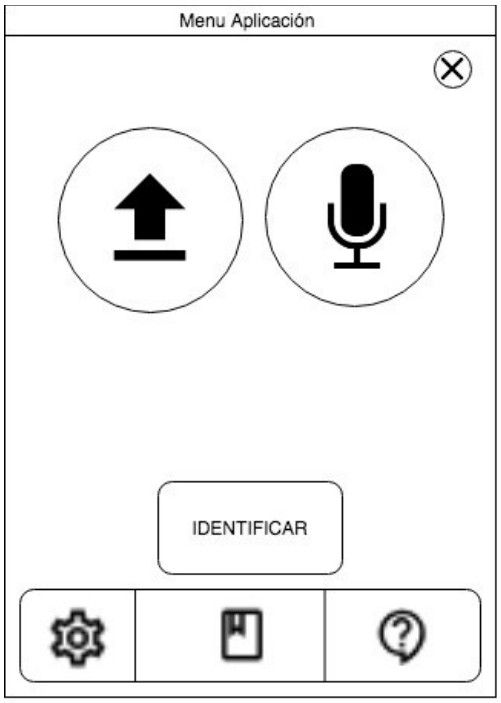


Ilustración 29 Menú de Navegación

Cuando el usuario ha cargado un archivo de audio o a grabado al presionar identificar se presenta la siguiente interfaz.

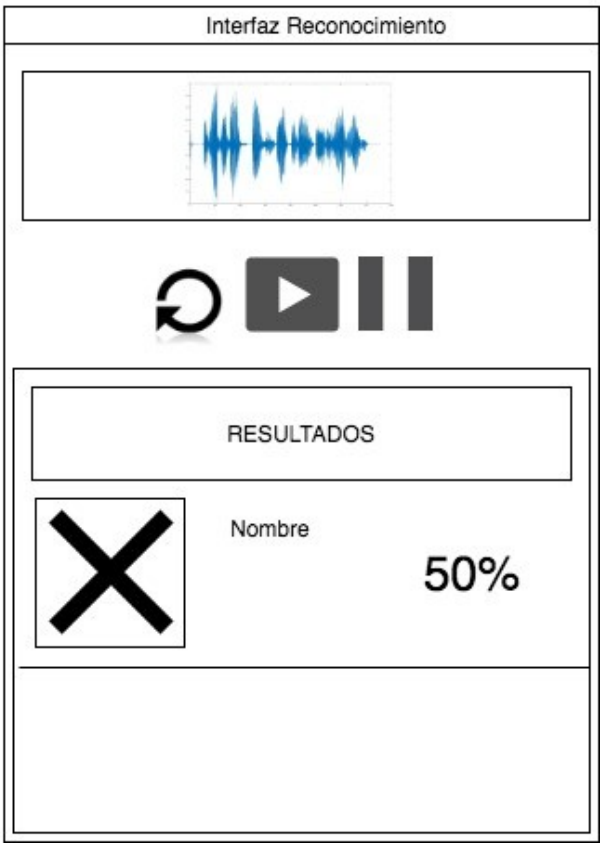


Ilustración 30 Interfaz gráfica de usuario reconocimiento.

Interfaz Catalogo de especies

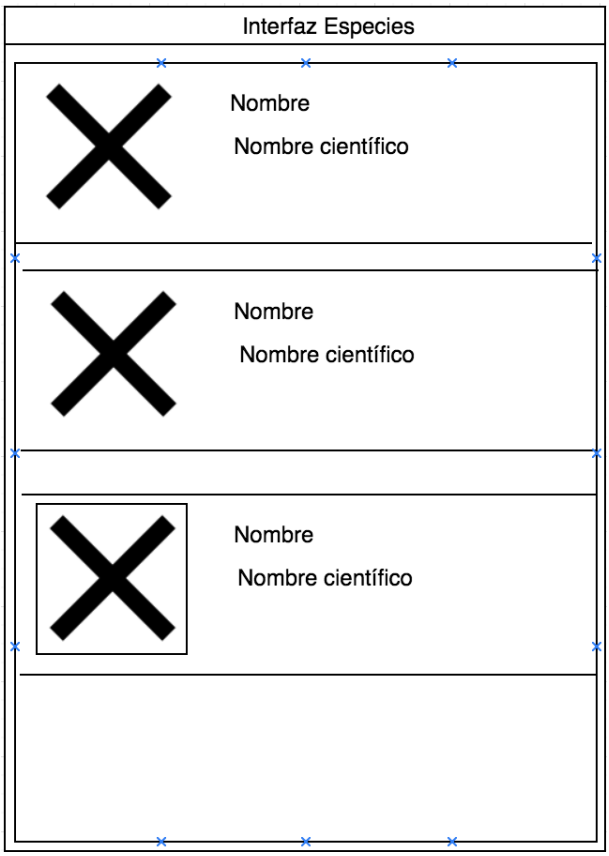


Ilustración 31 Interfaz gráfica de usuario catálogo de especies.

Interfaz especie específica o Información de la especie

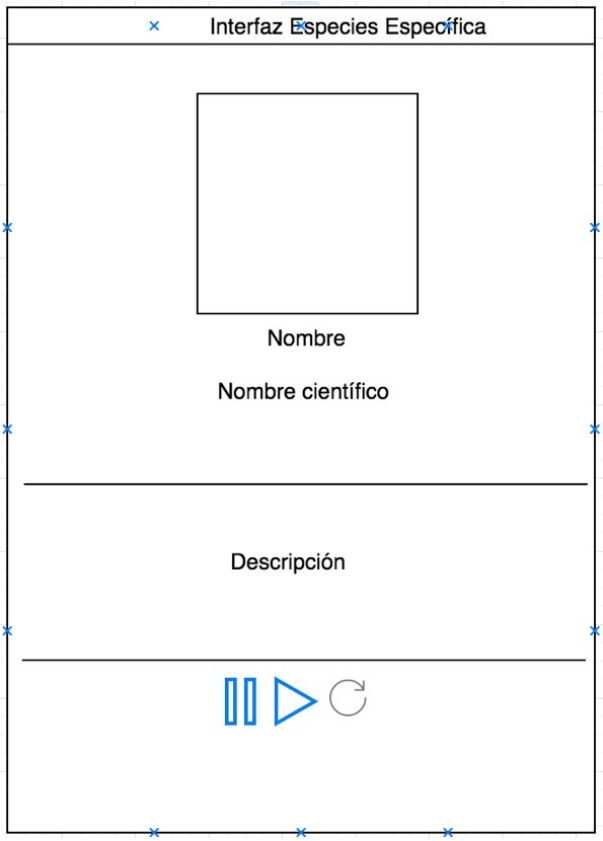


Ilustración 32 Interfaz especie específica.

CAPITULO 5: IMPLEMENTACIÓN Y PRUEBAS

En este capítulo detallaremos la implementación realizada durante los sprints del proyecto que se planificaron para 3 meses, cada sprint tiene su propia duración de acuerdo a lo planificado. Se realizaron en total 4 Sprints cada uno se describe a continuación.

5.1 SPRINT 1

Este Sprint se planifico para 3 semanas donde empezaremos con la creación de las interfaces de usuario con base en los diseños realizados en el sprint 0 y se implementaran algunas funcionalidades que se detallan a continuación.

Sprint Backlog

SPRINT BACKLOG 1			
Tarea ID	**Historia**	**Estimado (Días)**	**Prioridad**
2	Como usuario, deseo poder obtener una lista de especies ordenada por la similitud del canto.	5	1
1	Como usuario, deseo poder grabar desde mi celular los cantos de una especie por un minuto o menos.	5	2

Tabla 6 BackLog Sprint 1

Actividades para el Sprint

Actividades	**Estimado/ (horas)**
Diseño interfaz principal (menú de navegación, logo, etc.) en Android Studio	10
MODULO-CATALOGO: Implementación interfaz para visualizar el catálogo de especies.	6

MODULO-CATALOGO: Implementación interfaz para visualizar la información de cada especie (nombre, descripción, imagen, canto).	8
MODULO-CATALOGO: Diseño y llenado de la base de datos que contendrá el catálogo de especies.	10
MODULO-RECONOCIMIENTO: permitir al usuario cargar un archivo de audio .wav .	6
MODULO-RECONOCIMIENTO: permitir al usuario grabar un archivo de audio .wav y guardarlo.	10
MODULO-RECONOCIMIENTO: permitir al usuario obtener la gráfica de la data de un archivo de audio.	6
TOTAL (horas trabajadas)	56

Tabla 7 Tabla actividades Sprint 1

BurnDown Chart

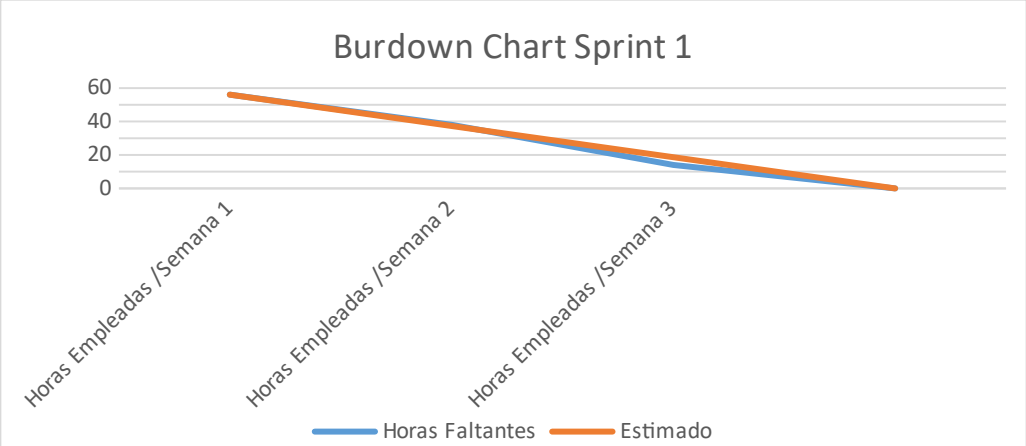


Ilustración 33 Cuadro Burn Down Sprint 1

Código realizado

A continuación, describimos el código necesario para realizar un activity. Un activity es un componente en Android que nos permite realizar interacciones entre el usuario y la aplicación. Por lo general se defina a una aplicación como una unión de activities. (Google Inc., 2017)
Esta activity está compuesta de un archivo de tipo .XML para su layout y una clase .java para el control de los eventos en esta.

Activity catálogo layout:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".CatalogoActivity">
   <ListView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/especies">
   </ListView>

</android.support.constraint.ConstraintLayout>
```

Activity Catálogo Clase:

```
public class CatalogoActivity extends AppCompatActivity {

    public static final String TAG = "CatalogoActivity";


    ArrayList<String > lista= new ArrayList<>();
    ListView especiesView;
    ArrayList<Bitmap> imagenes;
    ArrayList<Especie> especies;
    model modelo;
    int[] images;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        modelo=new model( context: this);
        especies=modelo.consultarEspecies();
        setContentView(R.layout.activity_catalogo);
        especiesView = findViewById(R.id.especies);

        CustomAdapter customAdapter;
        customAdapter = new CustomAdapter();

        especiesView.setAdapter(customAdapter);
        especiesView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent=new Intent( packageContext: CatalogoActivity.this,EspecieActivity.class);
                intent.putExtra( name: "Nombre",especies.get(position).getNombreEspecie());
                intent.putExtra( name: "NombreCientifico",especies.get(position)
 .getNombreCientificoEspecie());
                intent.putExtra( name: "Imagen",especies.get(position).getImagenEspecie());
                intent.putExtra( name: "Descripcion",especies.get(position).getDescripcion());
                startActivity(intent);

            }
        });
    }
    class CustomAdapter extends BaseAdapter {

        @Override
        public int getCount() {
            return especies.size();
        }

        @Override
        public Object getItem(int position) {
            return null;
        }

        @Override
        public long getItemId(int position) {
            return 0;
        }
        @Override
        public View getView(int position, View convertView, ViewGroup parent) {

            View view= getLayoutInflater().inflate(R.layout.especielayout, parent , attachToRoot: false);

            ImageView especieImage = view.findViewById(R.id.imagenEspecie);
            TextView nombreEspecie=view.findViewById(R.id.nombreEspecie);
            TextView nombreCientifico=view.findViewById(R.id.nombreCientifico);
            especieImage.setImageBitmap(BitmapFactory.decodeByteArray(especies.get(position)
 .getImagenEspecie(), offset: 0,especies.get(position).getImagenEspecie().length));
            nombreEspecie.setText(especies.get(position).getNombreEspecie());
            nombreCientifico.setText(especies.get(position).getNombreCientificoEspecie());
            return view;
        }
    }
}
```

Ilustración 34 Fragmento de código Clase Catalogo Activity

El siguiente código nos permite el crear un componente propio para colocar en la lista de especies donde se pueda visualizar la fotografía, el nombre de la especie y otros datos importantes.

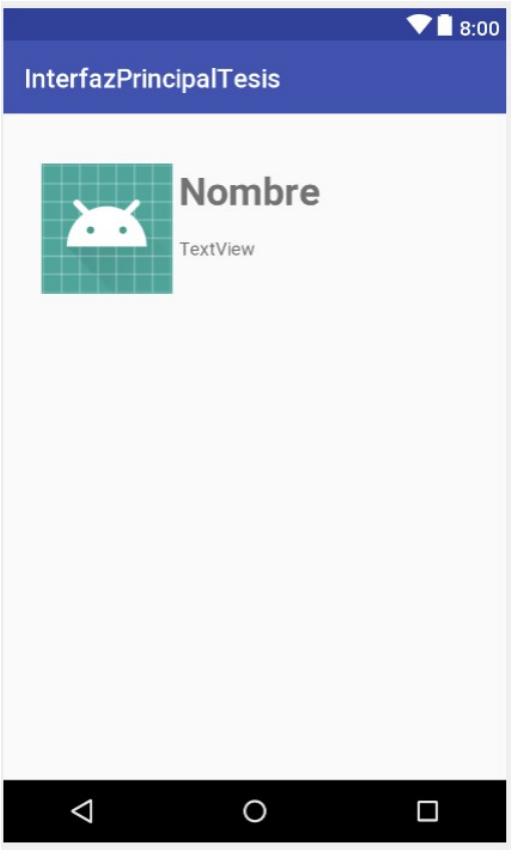


Ilustración 35 Interfaz Gráfica de Usuario Catálogo Especies.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<ImageView
    android:id="@+id/imagenEspecie"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="29dp"
    android:layout_marginTop="38dp"
    app:srcCompat="@mipmap/ic_launcher"
    android:layout_marginLeft="29dp"
    android:layout_alignParentLeft="true" />

<TextView
    android:id="@+id/nombreEspecie"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignTop="@+id/imagenEspecie"
    android:layout_alignParentRight="true"
    android:text="Nombre "
    android:textStyle="bold"
    android:textSize="30sp"/>

<TextView
    android:id="@+id/nombreCientifico"
    android:layout_width="244dp"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignStart="@+id/nombreEspecie"
    android:layout_marginTop="94dp"
    android:text="TextView" />
</RelativeLayout>
```

Ilustración 36 Fragmento Código, especie layout

Interfaces Creadas

Nos basamos en los diseños definidos en el Sprint 0 para crear las interfaces de usuario.

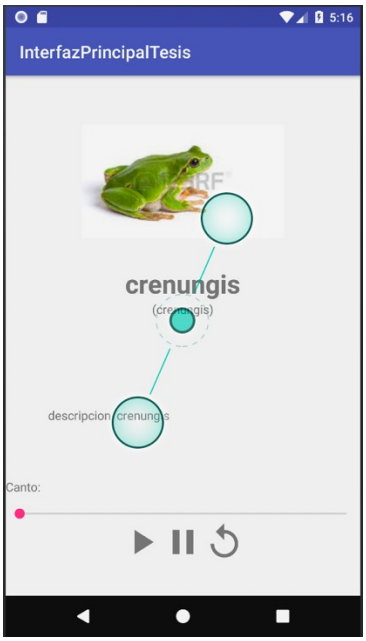


Ilustración 37 Interfaz de Información de la Especie

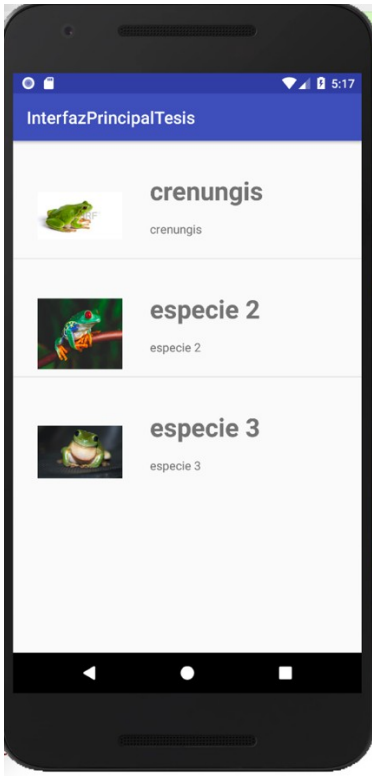


Ilustración 38 Primera versión Interfaz Catalogo de especies

5.2 SPRINT 2

En este Sprint nos centramos principalmente en la creación del módulo de reconocimiento a través del procesamiento de audio y su integración con las interfaces creadas anteriormente.

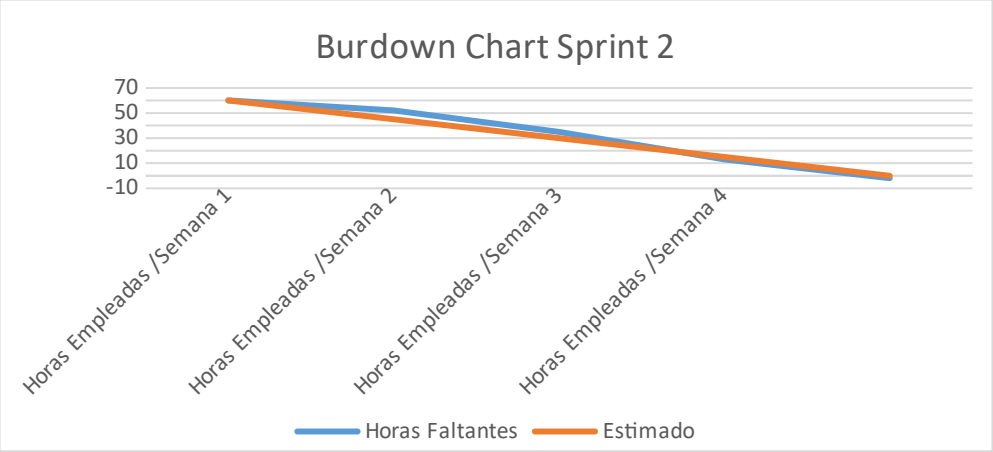
Sprint Backlog

PRODUCT BACKLOG PARA LA APLICACIÓN			
Tarea ID	**Historia**	**Estimado (Días)**	**Prioridad**
3	Como usuario, deseo poder visualizar el puntaje con que se determinó la similitud del canto de varias especies, así como la lista de especies encontradas en el canto.	5	3
6	Como usuario, deseo poder cargar un archivo de audio para su procesamiento.	5	4

Actividades del Sprint

Actividades	**Estimado/ (horas)**
MODULO-RECONOCIMIENTO: Desarrollo de la extracción de las características MFCC de audio.	20
MODULO-RECONOCIMIENTO: Entrenamiento del modelo de reconocimiento.	20
MODULO-RECONOCIMIENTO: Validación y pruebas del modelo de reconocimiento.	15
MODULO-RECONOCIMIENTO: Integración del procesamiento de audio y la lista de especies reconocidas con la interfaz gráfica de usuario.	5
TOTAL (horas trabajadas)	60

BurnDown Chart



Código realizado

```
////////////////////////////////ABRIR ARCHIVO Y OBTENER LA DATA Y FRECUENCIA DE MUESTREO////////////////////////////////////
// Abrir el archivo de audio(File) que se envía como parámetro
Log.i("reconocimiento","Se va a abrir el archivo ");
WavFile wavFile = WavFile.openWavFile(archivo_audio);
// frames que se leeran con el buffer
int numero_frames= 10000;
int total_frames=(int) wavFile.getNumFrames();
Log.i("reconocimiento","Se esta decodificando el audio ");
// Obtener el numero de canales de la señal
int numChannels = wavFile.getNumChannels();
Fs= (int) wavFile.getSampleRate();
// numero de iteraciones para leer el archivo
int iteraciones= total_frames/numero_frames;
int puntero_data=0;

// crear un buffer par almacenar los datos de la señal
float[] buffer = new float[numero_frames * numChannels];
// inicializacion del arreglo que contendra toda la data de la señal de audio
data_archivo = new double[total_frames * numChannels];

int framesRead;
//recorrer la señal de audio
for (int i=0;i<iteraciones;i++){
   // Read frames into buffer
   framesRead = wavFile.readFrames(buffer, numero_frames);
   for (int j=0;j< numero_frames;j++){
      if(puntero_data<total_frames){
         data_archivo[puntero_data]= buffer[j];
         puntero_data++;
```

```
        }
    }
    Log.i("reconocimiento","Se leyo "+ numero_frames+ " frames");

}
// Close the wavFile
wavFile.close();
Log.i("reconocimiento","Se termino de decodificar el audio ");
Funciones.notificar("Se terminó de cargar el archivo de audio", getApplicationContext());



///////FUNCION QUE TOMA COM PARAMETROS LA SEÑAL DE AUDIO Y SU FRECUENCIA DE
MUESTREO PARA RETORNAR UNA LISTA DE VECTORES DE CARACTERISTICAS
MFCC//////////////////////////////////////////
public List<double[]>  obtenerCaracteristicasMFCC(double[] data_archivo, int Fs){
    List<double[]> featuresMfccs= new ArrayList<>();
    // longitud del segmento de la señal de audio
    int long_Segemnto=5000;
    // numero de intervalos en los que se dividira la señal
    int numero_intervalos=data_archivo.length/long_Segemnto;
    // puntero para recorrer la señal de audio
    int puntero_data_archivo=0;
    //******************************Parametros para la extraccion de
caracteristicas******************************
    int nnumberofFilters = 24;
    int nlifteringCoefficient = 20;
    boolean oisLifteringEnabled = true;
    boolean oisZeroThCepstralCoefficientCalculated = false;
    int nnumberOfMFCCParameters = 20; //without considering 0-th
    double dsamplingFrequency = 44100;
    int nFFTLength = 4096;
    if (oisZeroThCepstralCoefficientCalculated) {
        //take in account the zero-th MFCC
        nnumberOfMFCCParameters = nnumberOfMFCCParameters + 1;
    } else {
        nnumberOfMFCCParameters = nnumberOfMFCCParameters;
    }
    //************************Creacion del objeto MFCC que servira para obtener las
caracteristicas MFCC************
    MFCC mfcc = new MFCC(nnumberOfMFCCParameters,
            dsamplingFrequency,
            nnumberofFilters,
            nFFTLength,
            oisLifteringEnabled,
            nlifteringCoefficient,
            oisZeroThCepstralCoefficientCalculated);
    double[] segmento=null;
    // recorrer la señal por los intervalos
```

```
   for (int i=0;i< numero_intervalos;i++){
      segmento =
com.sistemas.puce.interfazprincipaltesis.Util.Funciones.cortarSenal(data_archivo,
puntero_data_archivo, puntero_data_archivo+long_Segemnto-1);
      puntero_data_archivo= puntero_data_archivo+long_Segemnto;
      // parte del ventaneo mfcc que se hara con los siguientes parametros
      //**********************************definir parametros del frame
*****************************
      int long_segmento = segmento.length;// longitud del segmento del canto
      int stepsize = 40;//MFCC step size in msec
      long mfcc_window = 80;// ventana de mfcc ens msec
      mfcc_window = ((mfcc_window) * Fs) / 1000;// Frames in sec to frames in samples
      long numOfFrames = ((long_segmento - mfcc_window) / stepsize) + 1;
      //**********************************Recorrer los
frames************************************
      int curPos = 0;
      for (int j = 0; j < numOfFrames; j++) {// recorrer el frame
        // obtener el frame actual
        double[] frame =
com.sistemas.puce.interfazprincipaltesis.Util.Funciones.cortarSenal(segmento, curPos,
(int) (curPos + mfcc_window - 1));//y(curPos:curPos+mfcc_window-1);
        curPos = curPos + stepsize;// para permitir la segmentacion desde la mitad del frame
anterior + 20 muestras mas
        double[] coeficientes_features = mfcc.getParameters(frame);
        // agregar los coeficientes a la matriz
        featuresMfccs.add(coeficientes_features);
      }
      salida_consola(i,numero_intervalos, numOfFrames);

   }
   System.out.println("Se termino de sacar las caracteristicas del canto ");
   return featuresMfccs;
}

////////////////////////////////RECONOCERA EL AUDIO Y RETORNARA UNA LISTA CON LOS
MODELOS IDENTIFICADOS POR CADA BLOQUE DE LA SEÑAL//////////////////////////////7
public List<ModelosEspecieIdentificada> reconocerAudio(){
   // lista de modeles seleccionados con sus intervalos
   List<ModelosEspecieIdentificada> lista_modelos_seleccionados= new ArrayList<>();
   // variables para recorrer la lista de  vecotres de caracteristicas mfcc
   int puntero_mfcc=0;
   int step=80;
   int numero_intervalos= lista_mfccs.size()/ step;
   double score_temp , score_final=0 ;
   String winner = "";
   ModelosEspecieIdentificada modelo_ganador=new ModelosEspecieIdentificada();
   int id_ganador=0;
   int cont_modelos=0;
   double[][] prueba_mfcc = null;
   // Recorrido de los vectores de caracteristicas para su reconocimiento
   for (int i=0;i<numero_intervalos;i++){
```

```
        //**********************PROCESO DE
RECONOCIMIENTO*****************************
        // obtener una matriz de caracteristicas
        prueba_mfcc =
com.sistemas.puce.interfazprincipaltesis.Util.Funciones
.listToArray(lista_mfccs,puntero_mfcc,puntero_mfcc+step-1);
        // obteneer el modelo más parecido de acuerdo a la amtriz de caracteristicas MFCC
        modelo_ganador= obtenerModelo(prueba_mfcc, puntero_mfcc, puntero_mfcc+step-
1);
        // guardar el mejor modelo para ese intervalo
        lista_modelos_seleccionados.add(modelo_ganador);
        //recorrer el puntero en la lista de mfccs
        puntero_mfcc=puntero_mfcc+step;
    }
    // ultima iteracion , obtener las ultimas filas y  enviarlas para su reconocimiento
    modelo_ganador=new ModelosEspecieIdentificada();
    prueba_mfcc =
com.sistemas.puce.interfazprincipaltesis.Util.Funciones
.listToArray(lista_mfccs,puntero_mfcc,lista_mfccs.size()-1);
    modelo_ganador= obtenerModelo(prueba_mfcc, puntero_mfcc, puntero_mfcc+step-1);
    lista_modelos_seleccionados.add(modelo_ganador);
    // retornar la lista de modelos seleccionados
    return  lista_modelos_seleccionados;
}
```

Interfaces Creadas

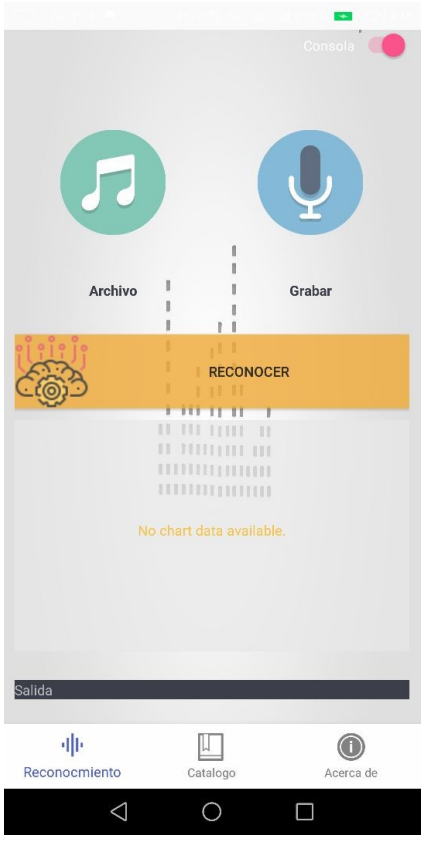


Ilustración 39 Interfaz de reconocimiento

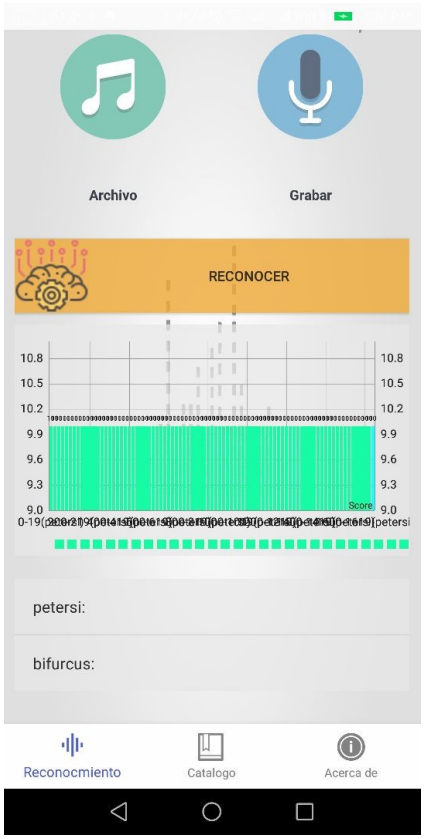


Ilustración 40 Interfaz de reconocimiento después de reconocer un canto

Ilustración 41 Interfaz de reconocimiento. resultados detallados en consola

5.3 SPRINT 3

En este Sprint nos enfocamos en finiquitar temas de la conexión de las bases de datos y las interfaces de usuario.

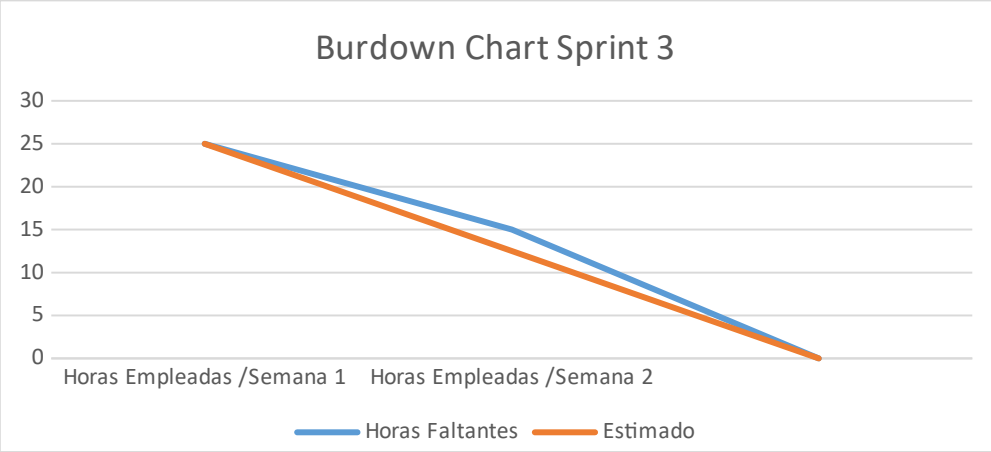
Sprint Backlog

PRODUCT BACKLOG PARA LA APLICACIÓN			
Tarea ID	**Historia**	**Estimado (Días)**	**Prioridad**
5	Como usuario, deseo poder escuchar los cantos de las distintas especies.	5	5
4	Como usuario, deseo poder obtener más información (nombre científico, descripción, imagen) sobre las especies de ranas que se encuentren en la aplicación.	5	6

Actividades del Sprint

Actividades	**Estimado/ (horas)**
MODULO-CATÁLOGO: Integración de: reproducción del audio.	10
MODULO-CATÁLOGO: Integración de base de datos con interfaz especie específica.	15
TOTAL (horas trabajadas)	25

BurnDown Chart



Código realizado

Este código es el encargado de conectarse con la base de datos para la extracción de los datos. Una vez recuperados los datos se almacenan en un arreglo para su posterior visualización.

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.media.Image;
import java.io.ByteArrayInputStream;


import android.util.Base64;
import android.util.Log;

import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;
import com.sistemas.puce.interfazprincipaltesis.Controller.Especie;
import java.io.ByteArrayOutputStream;
```

```
import java.nio.ByteBuffer;
import java.util.ArrayList;
import java.util.List;


public class model extends SQLiteAssetHelper {
   private static final String DATABASE_NAME = "basedatos.db";
   private static final int DATABASE_VERSION = 1;

   public model(Context context) {
      super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }


   public ArrayList<Especie> consultarEspecies() {
      SQLiteDatabase db = getReadableDatabase();
      Cursor c=  db.rawQuery("select * from especies ",null);
      ArrayList<Especie> list = new ArrayList<>();
      Especie actual;
      if ( c.moveToFirst()){
         int nuregistros=c.getCount();
         for(int i=0;i<nuregistros;i++){
            actual=new
Especie(c.getInt(0),c.getString(1),c.getString(2),c.getBlob(4),c.getString(3)
            ,c.getDouble(5),c.getDouble(6),c.getString(7));
            list.add(actual);
            c.moveToNext();
         }
      }
      c.close();
      db.close();
      return list;
   }
}
```

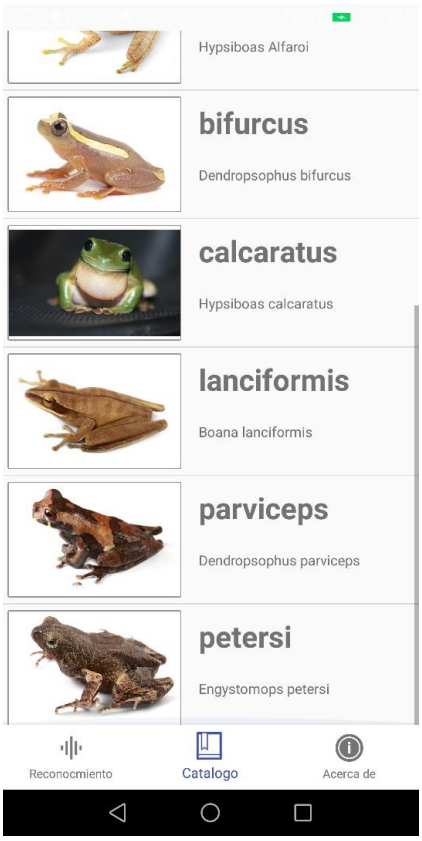


Ilustración 42 Interfaz Catálogo de especies

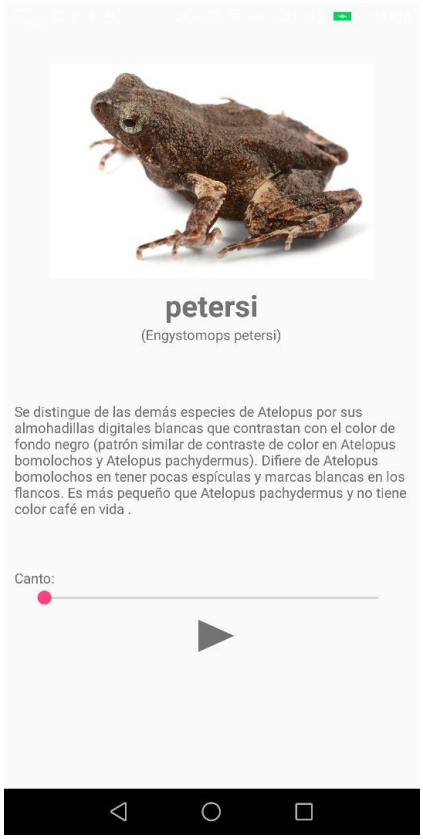


Ilustración 43 Interfaz para visualizar la información de la especie

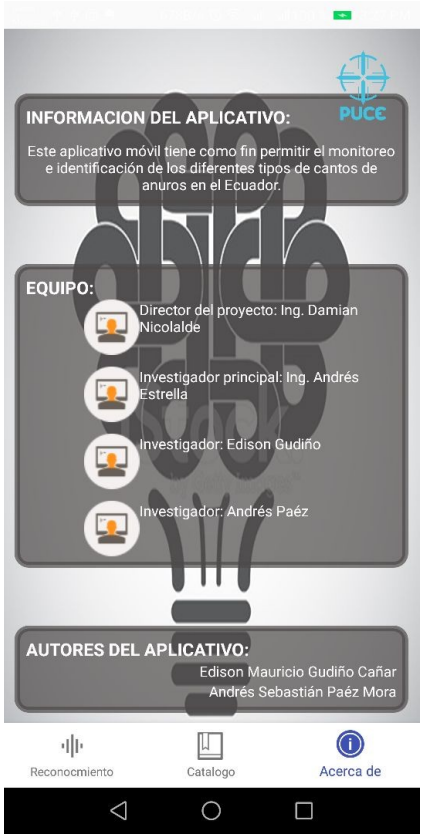


Ilustración 44 Interfaz Acerca de

5.4 PRUEBAS

En la siguiente sección especificaremos las pruebas realizadas a nuestra aplicación, estas se dividen en pruebas unitarias y pruebas de integración. Estas nos brindarán un panorama claro para determinar posibles errores. (Padmini, 2004) El proceso de pruebas y verificación nos permitirá el confirmar que una aplicación o una parte de la misma está realizando lo que se espera y prevenir defectos que nos pueden traer complicaciones al momento de la puesta en producción. (Schneider et al., 2010)

PRUEBAS UNITARIAS:

Estas pruebas se fueron realizando a lo largo del proceso de desarrollo con la finalidad de identificar errores de manera temprana y evitar contratiempos al momento de integrar todo el aplicativo móvil.

Componente / Producto	**Caso de prueba**	**Resultado**	**Conclusión**
Módulo - Reconocimiento	F4. Grabar archivo de audio.	Exitoso	Se logro grabar el audio, procesarlo y guardarlo en un archivo .wav
Módulo - Reconocimiento	F3. Cargar archivo de audio.	Exitoso	Se logro cargar un archivo de formato .wav al aplicativo.
Módulo – Reconocimiento	F4. Grabar Audio	Exitoso	Se logro grabar un archivo .wav y se almaceno en el dispositivo.
Módulo - Reconocimiento	F2. Reconocimiento del audio	Exitoso	Se comprobó que el reconocimiento fue correcto de acuerdo a los datos de prueba.
Módulo -	Extracción de	Exitoso	Se extrajeron

Componente / Producto	Caso de prueba	Resultado	Conclusión
Reconocimiento	características del audio		las características y se realizaron pruebas con datos controlados.
Módulo – Catálogo	Conexión con la base de datos	Exitoso	Se logro una conexión exitosa con la base de datos de acuerdo a parámetros establecidos.
Módulo-Catalogo	Listado de especies.	Exitoso	Se obtuvo la integración entre la conexión de la base de datos y la interfaz gráfica de usuario que muestra
Módulo-Catalogo	Datos específicos de especie.	Exitoso	Se muestra los datos específicos de una especie después de elegir una.

Pruebas de Integración (por hilos):

F1.1 Consulta de especies General.

Hilo: El actor desea visualizar las especies disponibles para su identificación.

Clases que apoyan el proceso:

MainActivity.java

Especie.java

CatalogoActivity.java

Conexion.java

Resultados esperados: El usuario abre la aplicación y se dirige al catálogo de especies, al abrir la ventana se despliega una lista de todas las especies en la base de datos.

F1.2 Consulta de especies específica.

Hilo: El actor desea visualizar una especie en especifico.

Clases que apoyan el proceso:

MainActivity.java

Especie.java

CatalogoActivity.java

EspecieActivity.java

Conexion.java

Resultados esperados: El usuario abre la aplicación y se dirige al catálogo de especies, toca la especie de la que quiere saber mas y la aplicación muestra los datos a detalle de una especie.

F2 Reconocimiento de audio.

Hilo: El actor desea realizar el reconocimiento de un audio.

Clases que apoyan el proceso:

MainActivity.java

Especie.java

MFCC.java

ModeloEspecie.java

GMM.java

ReconocimientoActivity.java

Conexion.java

Resultados esperados: El usuario abre la aplicación y se dirige a reconocimiento, y la aplicación devuelve la especie perteneciente del audio.

F3 Carga archivo de audio.

Hilo: El actor cargar un archivo de audio para su reconocimiento desde.

Clases que apoyan el proceso:

MainActivity.java

ReconocimientoActivity.java

File.java

WAVFile.java

Resultados esperados: El usuario abre la aplicación y se dirige a reconocimiento, el actor selecciona cargar archivo y se abre el navegador de archivos, el actor selecciona un archivo.wav y la aplicación emite un mensaje que se ha cargado exitosamente el audio.

F4 Grabación de audio.

Hilo: El actor desea realizar el reconocimiento de un audio.

Clases que apoyan el proceso:

MainActivity.java

Especie.java

File.java

WAVFile.java

ReconocimientoActivity.java

Conexion.java

Resultados esperados: El usuario abre la aplicación y se dirige a reconocimiento, el actor selecciona grabar audio, el sistema empieza a grabar hasta que muestra un mensaje grabación exitosa.

5.5 INSTALACIÓN

La instalación de la aplicación se realiza mediante un apk generada por el IDE Android estudio, actualmente no estara disponible en la tienda de aplicaciones de google debido a que esta aplicación forma parte de un proyecto de investigación por lo que la Pontificia Universidad Católica del Ecuador es quien tiene que realizar el proceso ya que es la entidad dueña de los derechos.

La aplicación soportara dispositivos que tengan un sistema operativo con un api minimo 26.

Una vez copiado el apk en el dispositivo Android, es necesario el tener un activada la opción de permitir aplicaciones de origenes desconocidos.

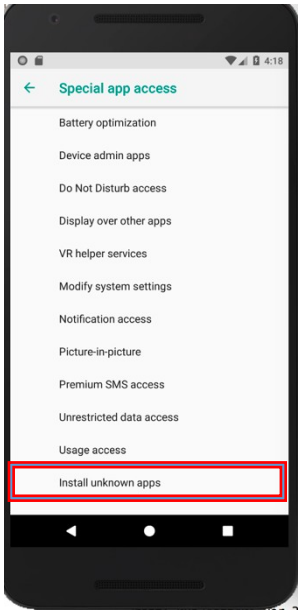


Ilustración 45 Instalación del apk

Accedemos a su carpeta contenedora y tocamos el archivo, seguimos los pasos de instalación propios de android y una vez finalizada la instalación estamos listos para utilizar la aplicación.

5.6 MANTENIMIENTO

Con el objetivo de crear un producto digital que se mantenga con el paso del tiempo y pueda ir creciendo en cuanto a caracteristicas y funcionalidades que aplica, se busca definir maneras en las que el mantenimiento y evolucion tienen que ser aplicadas a este proyecto.

Por la naturaleza de nuestro proyecto y su desarrollo con base en una metodologia agile de desarrollo de software sugerimos que para el mantenimiento y evolución del mismo se aplique una metodología agil, esto aumentara la eficiencia durante este proceso.

Especificamente hablamos sobre la metodologia Kanban que puede ser empleada como complemento de SCRUM para el proceso de mantenimiento de la aplicación. (Ahmad, Kuvaja, Oivo, & Markkula, 2016)

Debido a que la aplicación no esta publicada en una tienda de aplicaciones no puede recibir una retroalimentación de sus usuarios por esto medio, por lo que sugerimos que la retroalimentación se realice mediante entrevistas personales con el personal encargado de utilizar la herramienta. Así conseguiremos identificar las areas de mejora para la aplicación. Una vez identificados los requerimientos de mejora se debe proceder mediante una metodología para realizar los cambios.
Se debe revisar continuamente el software para asegurar su funcionamiento correcto.

I. CONCLUSIONES Y RECOMENDACIONES

a. CONCLUSIONES

Se cocluyo el desarrollo un aplicativo movil para el proyecto de investigación “Diseño de un sistema de estimación de indicadores de biodiversidad en base a un algoritmo de análisis automático de audio digital”, con el fin de brindar a los investigadores una

herramienta para automatizar el proceso de identificación de especies a base de su sonido.

Aplicamos la metodologia de desarrollo de software SCRUM que nos brindo un marco de trabajo sostenido y compatible para el desarrollo de aplicaciones de este tipo; enfocadas en dispositivos moviles, altos grados de incertidumbre en cuanto al tema a desarrollar y empleando tecnologias nuevas para su desarrollo.

Para el desarrollo de aplicaciones Android empleamos Java como lenguaje de programación y el IDE Android Studio que es el recomendado por el fabricante para la construcción de aplicaciones nativas. Estas herramientas nos brindaron a lo largo de todo el proceso un sin fin de recursos y funcionalidades que sirvierón de soporte, la documentación existente es sumamente amplia y especifica para diferentes casos de uso apoyado de una comunidad creciente.

Los diagramas UML nos apoyaron durante el proceso en las fases iniciales para que el equipo tenga un entendimiento común sobre la arquitectura del sistema y sus funcionalidades.

El procesamiento de señales de audio en el transcurso del desarrollo de la aplicación móvil fue una tarea más complicada que su procesamiento en una computadora debido a diferentes variables como lo son los recursos hardware, el lenguaje de programación y el sistema operativo.

La Inteligencia artificial y sus componentes son una herramienta potente para la automatización de cualquier proceso, en este caso el proceso a automatizar fue el reconocimiento de cantos de diferentes especies presentes en una señal de audio. Además, cabe recalcar que para poder lograr con éxito dicha automatización hay que tener en cuenta ciertos aspectos como el tipo de data a tratar , el volumen de la misma , sus características y recursos disponibles para poder posteriormente elegir algún modelo matemático que más se ajuste a las necesidades del problema.

b. RECOMENDACIONES

Es impresindible que el mantenimiento de aplicación se debe realizar en tres aspectos fundamentales; contenido, diseño y funcionalidad. Es importante el entender queel algoritmo es fruto de un proyecto de investigación por lo que si se desea realizar

cambios en el modulo de reconocimiento se tiene que realizar una investigación sostenida para dicho cambio.

Se recomienda el alimentar la base de datos con nuevas especies y modelos entrenados con el algoritmo original desarrollado en MatLab, se debe comprobar los resultados por lo que recomendamos seguir el siguiente diagrama de flujo para cargar una nueva especie al sistema.

Ilustración 46 Diagrama Flujo ingreso nuevos modelos

Se recomienda el aplicar un proceso de desarrollo agili preferenteme Kanban para realizar los mantenimientos y cambios de software.

Siempre se tiene que tomar en cuenta a los usuarios del sistema para realizar cambios tanto en el diseño como en el contenido, ya que el objetivo principal es que logren utilizar esta herramienta ininterrumpidamente con la mejor.

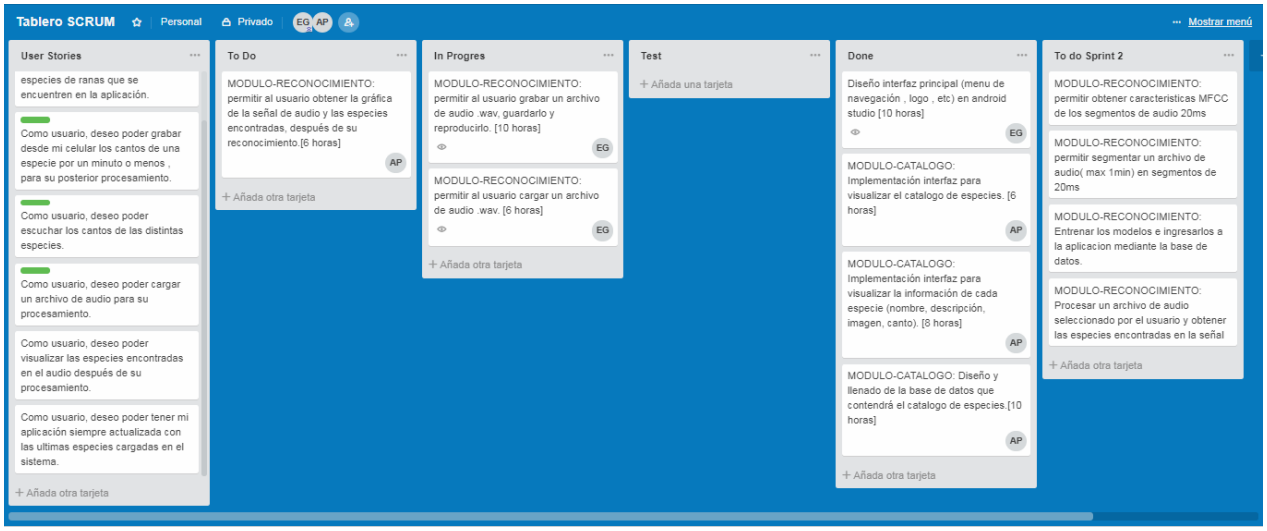
Se recomienda para el reconocimiento de los cantos proveer al aplicativo de archivos de audio con la mejor calidad posible ya que los modelos de reconocimiento fueron entrenados con grabaciones de alta calidad.

Optimizar el algoritmo MFCC o en su defecto cambiarlo por otro más ligero para la extracción de características de los segmentos de audio, debido al tiempo que se demora en procesar este algoritmo en los dispositivos móviles.

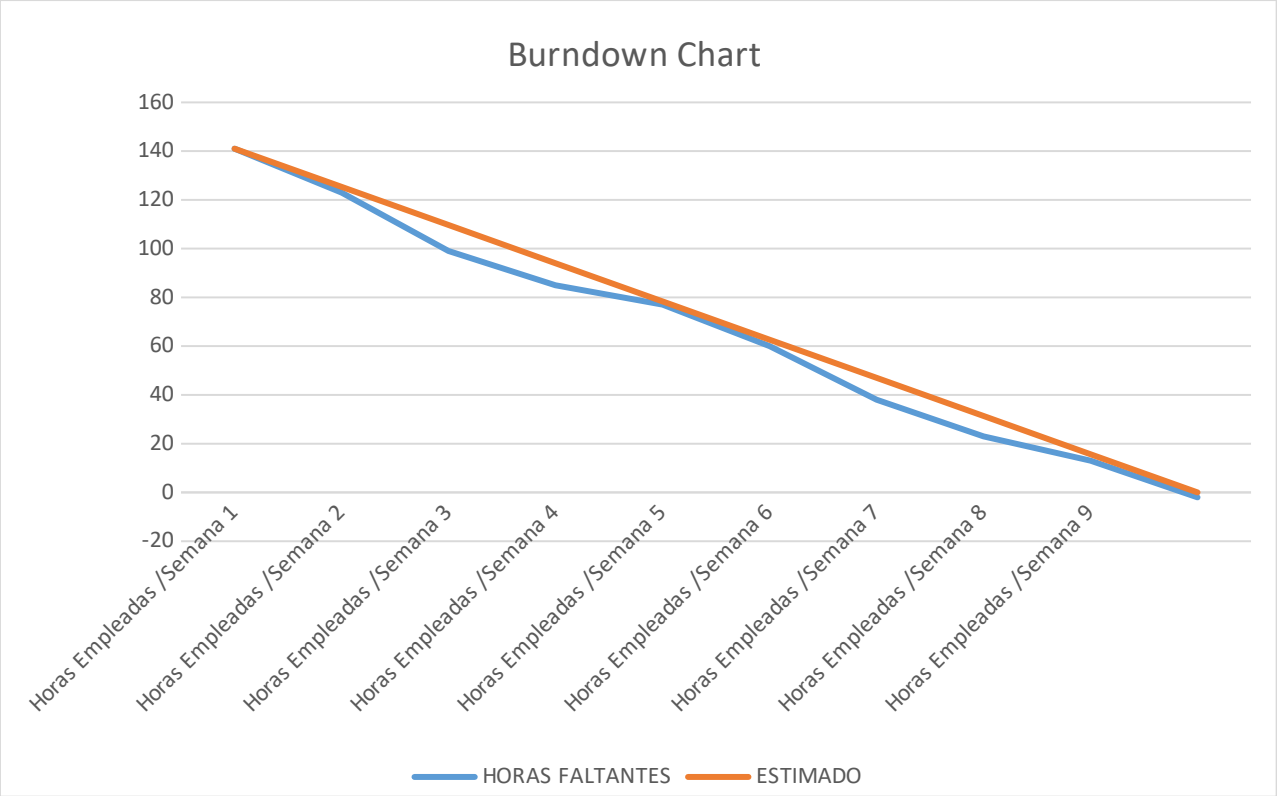
Entrenar nuevos modelos de reconocimiento empleando otras técnicas de Machine Learning para poder comparar su eficiencia con los actuales modelos.

II. ANEXOS

TABLERO SCRUM

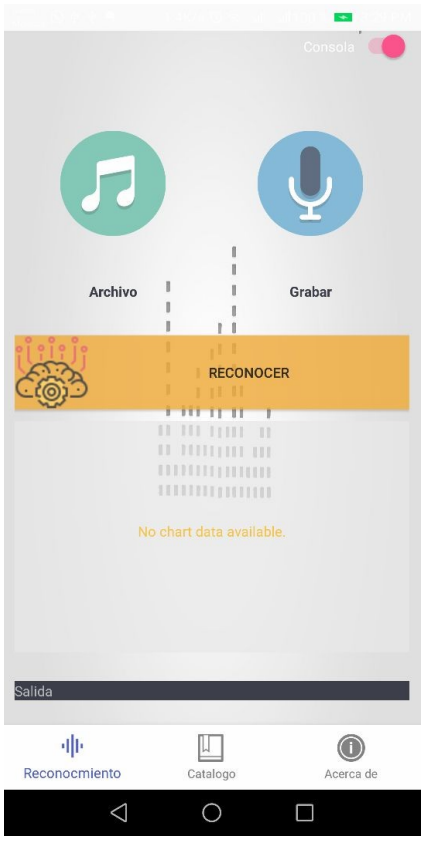


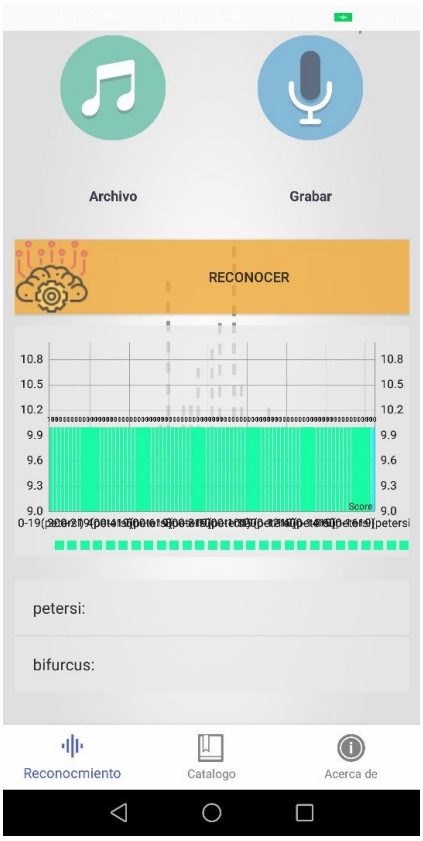
BURNDOWN CHART

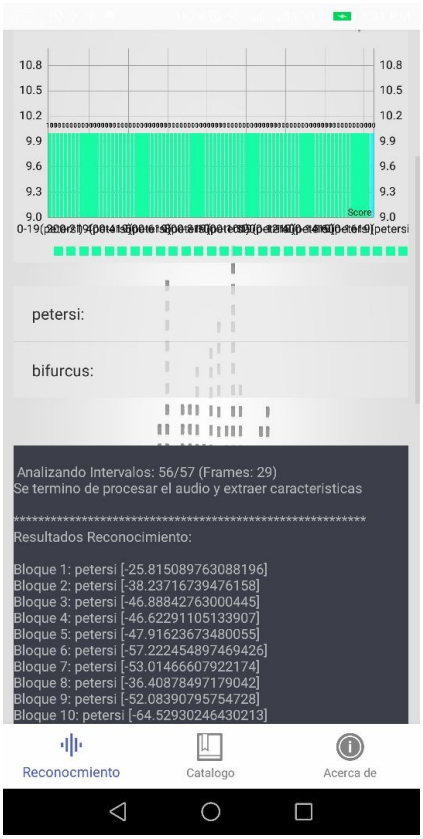


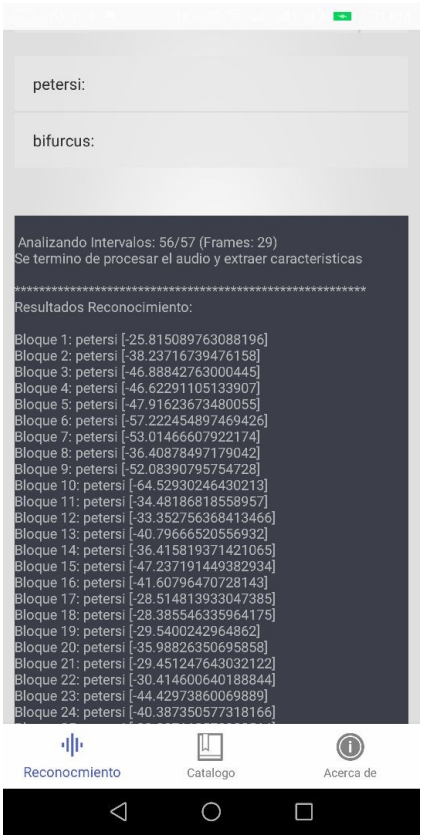
INTERFACES DEL APLICATIVO

Pantalla de Reconocimiento

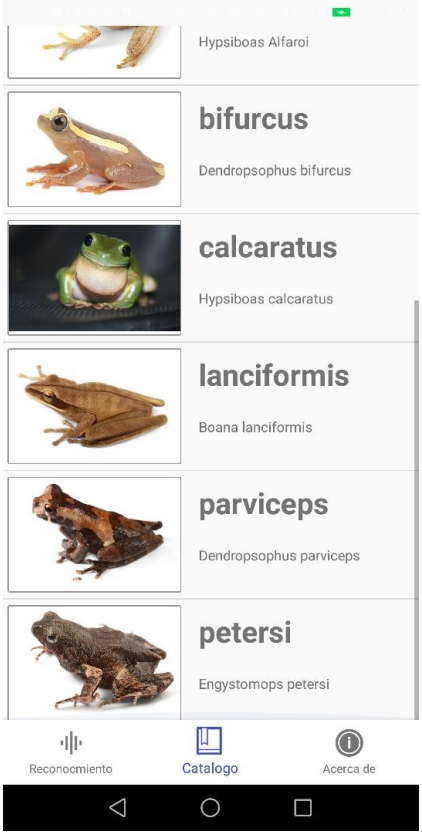


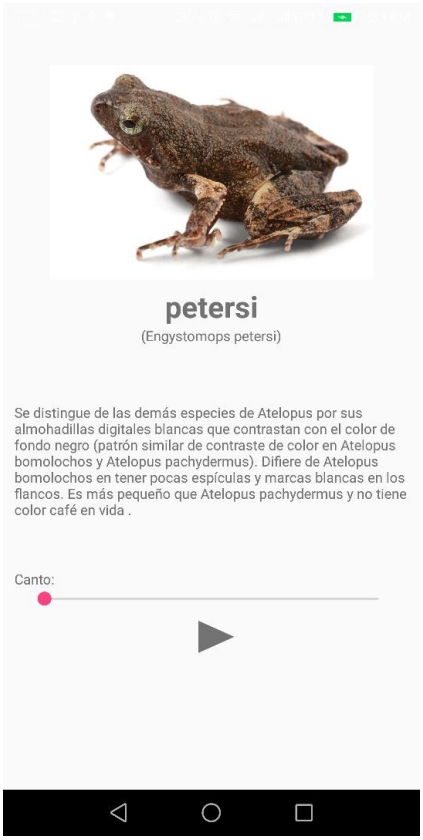


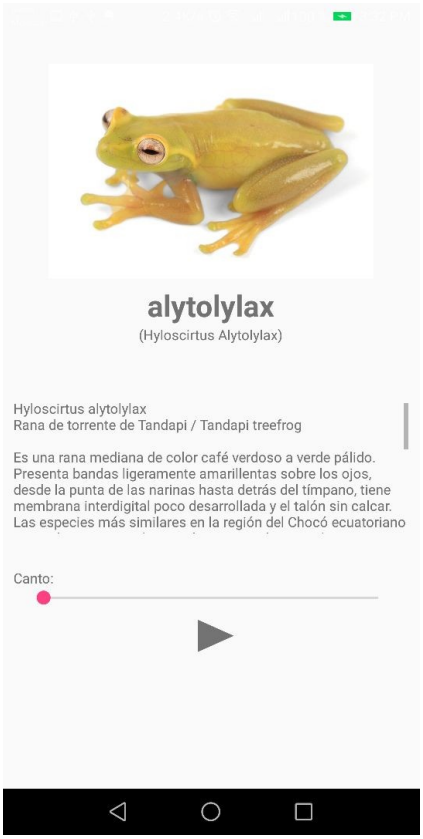


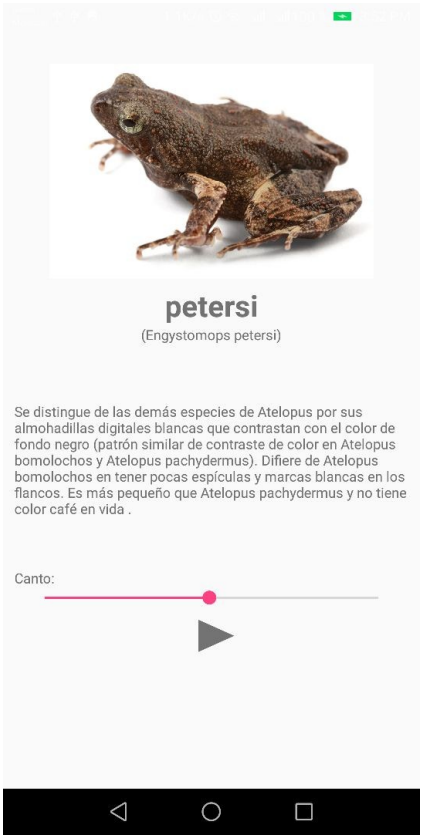


Pantalla Catálogo de Especies

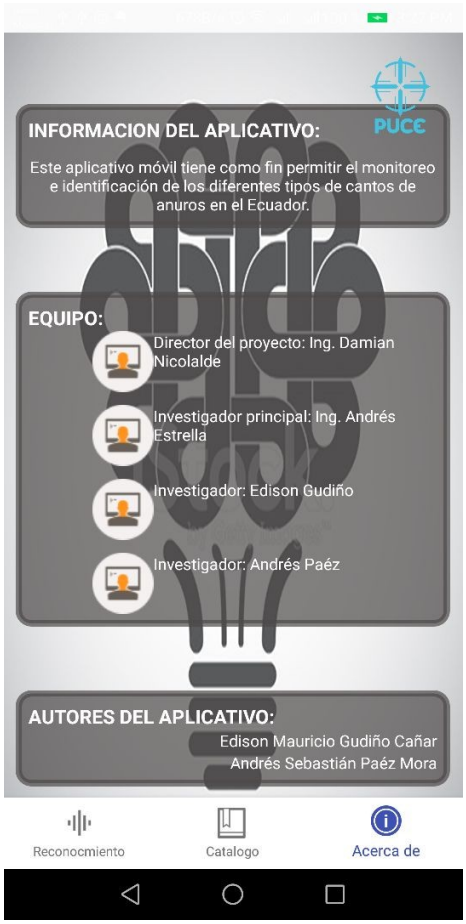








Pantalla Acerca de


INFORMACION DEL APLICATIVO:
Este aplicativo móvil tiene como fin permitir el monitoreo e identificación de los diferentes tipos de cantos de anuros en el Ecuador.
EQUIPO:
Director del proyecto: Ing. Damian Nicolalde
Investigador principal: Ing. Andrés Estrella
Investigador: Edison Gudiño
Investigador: Andrés Paéz
AUTORES DEL APLICATIVO:
Edison Mauricio Gudiño Cañar
Andrés Sebastián Paéz Mora
Reconocmiento
Catalogo
Acerca de

MANUAL DE USUARIO

Adjunto como PDF

III. BIBLIOGRAFÍA

Altassian. (2018, Mayo 24). *Trello*. Retrieved from trello.com: https://trello.com/guide/trello-101
Anfibios del Ecuador. (2017, Diciembre 16). *Introducción. Anfibios.* Retrieved from bioweb.bio: https://bioweb.bio/faunaweb/amphibiaweb
Caycedo, P., Ruiz, J., & Orozco, M. (2013). Reconocimiento automatizado de señales bioacústicas: Una revisión de métodos y aplicaciones. *Ingeniería y Ciencia*, 171-195.
Cisneros Barreiro, G., & Silva Saltos, E. A. (2018). DESARROLLO DE UN PORTAL WEB QUE PERMITA IDENTIFICAR ESPECIES DE RANAS DEL ECUADOR. Quito, Ecuador: PontificiaUniversidad Católica del Ecuador.
Española, R. A. (2017). *DLE*. Retrieved from http://dle.rae.es
Florence y Dan. (2016). *Warblr: the birdsong recognition app*. Retrieved from warblr.net: https://warblr.net/
Galiano, J. L. (2016). *Implantar SCRUM con éxito.* Barcelona: UOC.
Gartner. (2018). *Gartner IT Glossary*. Retrieved from https://www.gartner.com/it-glossary/smartphone
Google Inc. (2017, 11 07). *Android Apps on Google Play*. Retrieved from Google Play: https://play.google.com/store/apps
Instituto Nacional de Estadistica y Censo. (2016, julio 20). *Ecuador en cifras .* Retrieved from http://www.ecuadorencifras.gob.ec/en-cinco-anos-se-quintuplicaron-los-usuarios-de-telefonos-inteligentes/
iSpiny. (2016). *ChirpOMatic*. Retrieved from chirpomatic.com: http://www.chirpomatic.com/features.html
Kivy. (2017). *Kivy*. Retrieved from https://kivy.org
Marks, P. (2013, Agosto 28). *Biodiversity app logs insects by their telltale call*. Retrieved from newscientist.com: https://www.newscientist.com/article/mg21929324.000-biodiversity-app-logs-insects-by-their-telltale-call/#.Uh8tUXc3eWk?utm_source=NSNS&utm_medium=SOC&utm_campaign=twitter&cmpid=SOC%7CNSNS%7C2012-GLOBAL-twitter
Microsoft. (2018). *Visual Studio*. Retrieved from https://www.visualstudio.com/es/xamarin/
Rodríguez, M. (s.f). *Definición de una arquitectura para aplicaciones móviles.* Retrieved from Proyecto CBC: http://www.proyecto-cbc.org.pe/admin/recursos/publicaciones/4-Definicion-de-una-arquitectura-para-aplicaciones-moviles.pdf
Rubin, K. S. (2013). *Essential Scrum .* Michigan.
Scikit-learn. (2017, 10 26). *Gaussian mixture models*. Retrieved from http://scikit-learn.org/stable/modules/mixture.html
Scott, M. F. (1999). *UML Gota a Gota.* Mexico: Pearson.
Sunbird Images. (2017, Noviembre 25). *Bird Song Id USA Automatic Recognition & Reference*. Retrieved from sunbird.tv: http://sunbird.tv/sunbird-apps-ebooks/app-bird-song-id-usa/
Wildlife Acoustics. (2017, Diciembre 20). *SongSleuth*. Retrieved from songsleuth.com: https://www.songsleuth.com/
Xamarin. (2017). *Xamarin*. Retrieved from https://developer.xamarin.com/
Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of software maintenance teams from scrum to Kanban. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2016–March*(i), 5427–5436. https://doi.org/10.1109/HICSS.2016.670
Al., P. J. et. (2014). *Get agile!* (Second). Amsterdam.

Awad, M. A. (2005). A Comparison between Agile and Traditional Software Development Methodologies. *The University of Western Australia*, *1*, 1–300. https://doi.org/10.1145/130840.130843

C. Cortes, & V.Vapnik. (1995). Support Vector Networks. *Machine Learning*, *20*(3), 273~-~297. https://doi.org/10.1007/BF00994018

Castro Gil, R. A. (2015). Estructura básica del proceso unificado de desarrollo de software. *Sistemas & Telemática*, (September), 14. Retrieved from http://univirtual.unicauca.edu.co/moodle/pluginfile.php/17726/mod_resource/content/0/material/3_Procesos_II/Articulos/Lectura_3-_Estructura_bae_del_RUP.pdf

Caycedo-Rosales, P. C., Ruiz-Muñoz, J. F., & Orozco-Alzate, M. (2013). Reconocimiento automatizado de señales bioacústicas: Una revisión de métodos y aplicaciones. *Ingeniería y Ciencia - Ing.Cienc.*, *9*(18), 171–195.

Gasca Mantilla, M. C., Camargo Ariza, L. L., & Medina Delgado, B. (2014). Metodología para el desarrollo de aplicaciones móviles. *Tecnura*, *18 No 40*, 20–35. Retrieved from http://tecnura.udistrital.edu.co/ojs/index.php/revista/article/view/767

Georgiev, T., Georgieva, E., & Smrikarov, A. (2004). M-Learning. *Proceedings of the 5th International Conference on Computer Systems and Technologies - CompSysTech '04*, (June), 1. https://doi.org/10.1145/1050330.1050437

Higuera, J. A., Mario, C., Camelo, D., & Cediel, O. T. (2014). SCRUM : A TRAVÉS DE UNA APLICACIÓN MÓVIL Scrum : Through a Mobile Application, *2*(2).

Mariño, S. I., & Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Scientia et Technica, ISSN 0122-1701, Vol. 19, N°. 4, 2014, Págs. 413-418*, *19*(4), 413–418. https://doi.org/10.22517/23447214.9021

Matich, D. J. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia*, 55.

Nicolalde, D., & Pareja, D. (2016). Selección, anotación y etiquetado de los cantos de las ranas del Ecuador.

Padmini, C. (2004). Beginners Guide To Software Testing, 1–41.

Schneider, G. P., Ph, D., & Shipp, L. (2010). *Software Engineering Ninth Edition*. https://doi.org/10.1136/bmj.1.5802.756-b

Schwaber, K., & Beedle, M. (2001). Agile Software Development with Scrum. https://doi.org/10.1109/2.947100

Toro, N., Giraldo Gomez, S. F., & Salazar Jimenez, T. (2006). Reconocimiento De Especies De Anuros Por Sus Cantos , En Archivos De Audio , Mediante Técnicas De Procesamiento Digital De Señales. *Scientia et Technica*, *XII*(32), 1–6.

Villarreal, G. L. (2008). Notas de Scrum. *Linux+ (LPMagazine)*.