

UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

**Implementación de aplicación Kx para Yagantec
S.A.**

ANDRÉS GARRIDO ROJAS

Profesor Guía: RODOLFO ALLENDES

Proyecto de Memoria de Título
Ingeniería Civil en Computación

Curicó – Chile
Octubre, 2018

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Curicó, 2019

Dedico este trabajo a todas las personas que me apoyaron durante este proceso, en especial a mi novia Natalia Correa, quien siempre ha estado a mi lado en esta etapa de mi vida.

AGRADECIMIENTOS

Agradezco mucho la ayuda de mis maestros, compañeros, y en general a toda la Universidad de Talca por los conocimientos, herramientas y valores que me han entregado durante toda mi estadía. Por estar siempre dispuestos a que el alumno siga adelante a pesar de las diversas dificultades en las que particularmente me encontré. Gracias a todo esto hoy puedo ser un profesional íntegro capaz de ser autovalente en un mundo altamente competitivo y cambiante.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	V
Resumen	VI
1. Introducción	7
1.1. Contexto	7
1.2. Descripción del Problema	7
1.3. Objetivos	8
1.3.1. Objetivo General	8
1.3.2. Objetivos Especificos	8
1.4. Alcances	9
1.4.1. Herramientas a utilizar	9
1.4.2. Contenidos a Desarrollar	9
1.4.3. Entregables	9
1.4.4. Organización del Documento	10
2. Marco Teórico	11
2.1. Análisis del Estado del Arte	11
2.2. Sistema Operativo Android	13
3. Diseño	15
3.1. Descripción General	15
3.2. Arquitectura utilizada para la aplicación móvil	15
3.3. Arquitectura proyecto backoffice y versión web	18
3.4. Consideraciones previas al desarrollo	20
3.4.1. Disponibilizacion y consumo de datos	20

3.4.2. Sistema Operativo y dispositivos	20
3.4.3. Entregas a desarrollar	21
4. Implementación de la Solución	22
4.1. Estimación de horas y fechas	22
4.2. Etapas Realizadas	25
4.2.1. Desarrollo Versión Demo	25
4.2.2. Desarrollo Versión Final	28
4.3. Problemas/Desafíos enfrentados	38
4.3.1. Implementación efecto gráfico de <i>cambio de hoja</i>	38
4.3.2. Almacenamiento y despliegue de libro digital	39
4.3.3. Manejo de recursos multimedia	39
4.3.4. Implementación de sesión para carga y restauración de lectura	40
5. Pruebas y Refactoring	41
5.1. Layout y Organización de Aplicación	41
5.2. Prueba de conexión y manejo de datos	42
5.3. Lectura y Evaluaciones	43
6. Conclusión y Análisis Retrospectivo	44
6.1. Análisis del producto y futuras mejoras	44
6.2. Análisis del proceso de desarrollo	45
6.3. Conclusiones finales del proyecto	46
Bibliografía	47

ÍNDICE DE FIGURAS

	página
2.1. Tablets Kindle	11
2.2. Aplicaciones iBooks, Google Play Books y PageTurner Reader	12
2.3. Arquitectura Android	13
3.1. Arquitectura Plataforma Kx móvil	16
3.2. Arquitectura Plataforma Kx backend-web	18
4.1. Bienvenida y Calibración	25
4.2. Primera página texto Demo	26
4.3. Videos Explicativos	27
4.4. Modelo de Datos de Lectura	28
4.5. Modelo de Datos de Sonidos y Catálogo	29
4.6. Modelo de Datos de Trazabilidad y Notificaciones	30
4.7. Inicio usuario lector y tutor	32
4.8. Librería lector y tutor y detalle de un libro	33
4.9. Menú Aplicación	33
4.10. Controles de lectura	34
4.11. Vista de avances	35
4.12. Alertas de lectores, padres y perfil	36
4.13. Creación y visualización de contacto	37
5.1. Lectura vertical y horizontalmente	41
5.2. Paginación de la librería	42
6.1. Estadística avance de lectura	45

RESUMEN

Es sabido que en la educación a nivel nacional existen muchas deficiencias y debilidades y sobre todo en lo referente a la lectura y comprensión de ésta, es por esto que surge Yagantec para dar una ayuda con herramientas tecnológicas enfocándose principalmente en la educación de primer nivel, y en conjunto con ellos se decide crear Kx.

La herramienta constituye una aplicación creada para dispositivos Android que permite al alumno leer libros pero que incluyan una motivación extra, en este caso entregando una reproducción de sonidos que concuerden con lo que se está leyendo en el momento, esto tomando la velocidad de lectura del alumno como referencia, y a nivel complementario entregar servicios a los padres y apoderados como la estadísticas de evolución de la velocidad y comprensión lectora, esto gracias a preguntas que se realizan en distintos puntos de ésta, y además un sistema de alertas que permite tener informados a todos los usuarios de distintos eventos y recordatorios.

La aplicación se diseñó por un lado montando el modelo de datos, lógica de negocio y exposición de servicios, tarea en mayor parte realizada por Yagantec, y por otro lado la aplicación móvil que entrega toda esta funcionalidad en el dispositivo. Luego de finalizado el período de desarrollo con ellos se les entregó el proyecto para que cerraran el desarrollo por completo y lo implantaran según sus necesidades.

Este documento presenta los distintos aspectos cubiertos por la aplicación, su diseño, el proceso de desarrollo, estudios previos, uso de herramientas complementarias, el desarrollo de una versión demo, pruebas realizadas y un cierre donde se explican los objetivos alcanzados y las mejoras posibles de realizar en el futuro.

1. Introducción

1.1. Contexto

Yagantec, una empresa que desarrolla tecnología para la educación, proporciona ambientes lúdicos en donde niños pueden potenciar sus conocimientos especialmente en el área de lectura y comprensión, todo esto sin desmerecer el importante rol de las escuelas, profesores, y familias que participan en todo el proceso educativo.

Actualmente cuenta con un conjunto de profesionales que apoyan los proyectos tecnológicos, dentro de los que se incluyen diseñadores, escritores, editores de audio y video e ingenieros. Una de las plataformas más importantes con las que cuenta es KalaKai, la cual está diseñada para niños entre 6 y 13 años y que busca contribuir en mejorar el resultado de cada comunidad escolar en relación a la habilidad de comprensión lectora, utilizando para aquello métricas diagnósticas permanentes en fluidez y comprensión lectora, como a su vez, dinámicas de reforzamiento sistemáticas de forma individualizada.

1.2. Descripción del Problema

Uno de los problemas recurrentes en los alumnos de educación básica, es la dificultad para abordar las lecturas entregadas en sus colegios, por lo que muchas veces dejan esta actividad para último momento perdiendo completamente el objetivo principal de la lectura comprensiva. Por otro lado es muy complicado para los padres poder monitorear constantemente las actividades escolares que tienen sus hijos, sobre todo si se trata de una lectura. Por estas razones nace la idea de generar

una herramienta que de a los alumnos una nueva experiencia en la lectura, que los pueda motivar, pero también que sirva tanto a sus padres como a los profesores, en la medida que puedan ir evaluando la lectura realizada en todo momento y no al final cuando no queda tiempo.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una plataforma de lectura para dispositivos móviles (específicamente para los sistemas operativos Android) la que se demonina Kx.

1.3.2. Objetivos Especificos

La plataforma Kx, aparte de ser una aplicación de lectura, contará con un sistema de monitoreo y control para padres y profesores, para este caso particular se desarrollará específicamente la primera mencionada, la cual se espera que cumpla los siguientes objetivos:

- Investigación del estado del arte en el ámbito de tecnologías móviles, para determinar las posibles funcionalidades y características que tendría.
- Se debe crear una arquitectura que permita la integración de múltiples dispositivos móviles, como también un modelo de datos que represente la aplicación.
- Realización de una versión demo, con el objetivo de aproximar a los actuales y posibles clientes a la utilización de la aplicación como complemento educacional.
- Desarrollo la aplicación completa que semanalmente debe tener versiones estables de las funcionalidades planificadas.
- Sometimiento a pruebas que determinen una buena calidad del producto.

1.4. Alcances

La aplicación que se pretende desarrollar (desde ahora denominada Kx -Kalakai Experience-), será creada en conjunto con la empresa, por lo que todos los componentes audiovisuales, como también los textos e interfaces gráficas serán desarrolladas y diseñadas por terceras personas. Dentro de las características funcionales que tendrá la aplicación, caben destacar los siguientes puntos fundamentales para el proyecto.

1.4.1. Herramientas a utilizar

A continuación se listan las herramientas que se usarán en el desarrollo:

- **Editor(IDE):** Se comprende la utilización de Eclipse con el SDK oficial de Android.
- **Dispositivos:** Para la prueba de la aplicación se utilizarán dispositivos proporcionados por Samsung, y en los casos de pruebas más específicas se recurrirá a virtualizar dispositivos.

1.4.2. Contenidos a Desarrollar

Primero que todo como Kx será una aplicación de lectura, se deben investigar los software que ya tengan esta capacidad, y los formatos que ayuden a visualizar y almacenar los libros. Por otro lado se debe diseñar la herramienta de forma que la mayoría de los recursos se manejen de manera remota, y que también permita el almacenamiento de los datos de esta misma manera. Se debe crear la aplicación siguiendo rigurosamente los estándares entregados por Android.

1.4.3. Entregables

El desarrollo de Kx pretende dos entregables, uno será una aplicación de demostración, que trabajará completamente offline, pero que servirá para exponer el producto y los servicios que prestará. Luego se creará la aplicación final que contempla gran parte de las funcionalidades de la anterior, pero agregando las características pendientes, en las que principalmente se encuentra la capacidad de trabajar consumiendo servicios remotos para la mayoría de sus actividades y datos que se utilizan.

1.4.4. Organización del Documento

A continuación se describirán los capítulos que contiene este documento que expone todas las etapas necesarias para el desarrollo de Kx.

Capítulo 2

En esta sección se describe el análisis de las tecnologías actuales tanto en los lectores digitales como en el sistema operativo Android

Capítulo 3

En este capítulo expone el diseño adoptado para la realización de la solución

Capítulo 4

En este apartado se detalla el proceso realizado para la obtención y puesta en marcha del sistema propuesto

Capítulo 5

En esta sección se especifican las pruebas realizadas

Capítulo 6

Se analiza el producto entregado tanto como el trabajo realizado durante el proceso

2. Marco Teórico

2.1. Análisis del Estado del Arte

El desarrollo de la lectura digital es algo en lo que se ha estado trabajando desde hace bastante tiempo, entre 1940 y 1980 se tuvieron las primeras iniciativas de un dispositivo digital para realizar lecturas o almacenar libros, pero no sería hasta en los años 90 donde se desarrollarían dispositivos específicamente para leer. Actualmente uno de los líderes en el comercio es Amazon y su producto Kindle[7] donde el 2013 se estimaron recaudaciones superiores a los \$4 billones de dólares (estimación hecha por Morgan Stanley publicada por AllThingsD[10]).



Figura 2.1: Tablets Kindle

Dentro de la lectura digital existen variados Software que apoyan esta acción, los más destacados son los creados por Apple (iBooks) y Google (Google Play Ebooks), aunque dentro del mercado existen muchos otros software para los distintos dispositivos móviles y sistemas operativos. Para el caso de este trabajo se investigó una aplicación llamada *PageTurner Reader*[22] ya que es de código abierto y realiza muchas de las características que se buscaban para este proyecto, dentro de las principales se encuentran:

- Soporte formato ePub
- Sincronización en la nube para el progreso en la lectura
- Descarga de libros
- Compatible con dispositivos Android 2.1 o superiores
- Soporta búsquedas en catálogos con formato OPDS[1]
- Buscador de libros existentes en la librería
- Animación de página al cambiarse a otra
- Lectura día y noche
- Configuración del tamaño de la fuente y el tipo



Figura 2.2: Aplicaciones iBooks, Google Play Books y PageTurner Reader

2.2. Sistema Operativo Android

Android se ha convertido en uno de los sistemas operativos más utilizados al redor del mundo, con alrededor del 72% del mercado a finales de 2013 (investigación realizada por la firma Gartner[2]), dada la flexibilidad de su sistema a una amplia variedad de dispositivos y al impulso que se le dió para el desarrollo de software por terceros.

Android está compuesto de un conjunto de elementos descritos en el siguiente diagrama de su arquitectura extraído de sitio oficial de desarrollo[3]:

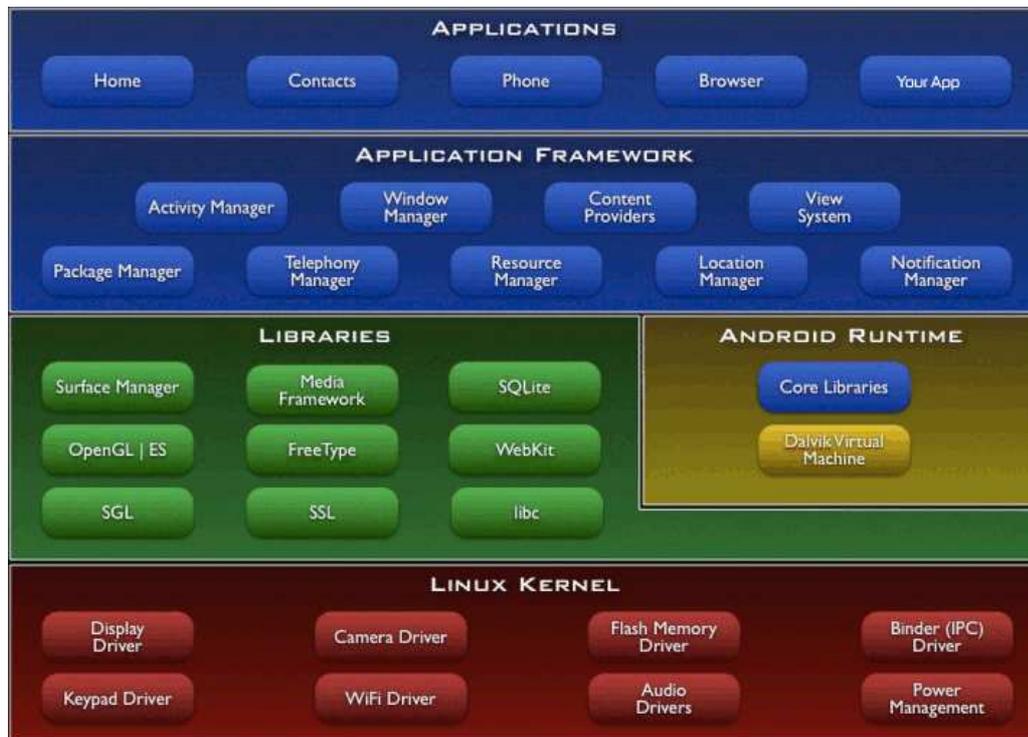


Figura 2.3: Arquitectura Android

Se puede apreciar una arquitectura basada en capas que se compone en su base por un Kernel Linux 2.6 con ciertos cambios realizados por Google y es el encargado de comunicarse con el hardware de los dispositivos. Las librerías escritas en su mayoría en C o C++ y que permiten que el dispositivo maneje distintos tipos de datos, entre ellos se destacan:

- **Surface Manager:** Es utilizado para generar la vista basado en un búfer que luego se combina con otros dibujos que resultan en lo desplegado en la pantalla.
- **Media framework:** Provee diversos codecs que permiten la grabación y reproducción de distintos formatos multimedia.
- **SQLite:** Corresponde al motor de base de datos que utiliza Android para almacenamiento de datos.
- **WebKit:** Corresponde al motor para visualizar contenido en HTML.
- **OpenGL:** Librería OpenGL utilizada para renderizar gráficos en 2D y 3D.

El Runtime de android se compone de dos elementos, una máquina virtual Dalvik[4] que es una JVM para dispositivos móviles que ejecutan aplicaciones de bajo consumo de memoria y procesamiento. Una particularidad es que esta máquina utiliza archivos .dex en vez de .class creados en tiempo de compilación a partir de estos últimos. Por otro lado se encuentran diversas librerías de Java que realizan varias de las funcionalidades definidas en las de Java SE.

El Framework corresponde a distintos componentes con los que se puede trabajar al programar una aplicación. Éstos permiten diversas funcionalidades como por ejemplo un administrador de recursos, de las llamadas de voz, de ubicación, uso de servicios, actividades, entre otros.

Las aplicaciones corresponden a la capa superior y en donde usuario puede interactuar. Una característica particular es que se pueden crear aplicaciones que reemplacen las existentes del sistema por defecto, lo que le da una alta flexibilidad para los desarrolladores.

3. Diseño

3.1. Descripción General

Para la implementación de la solución se desarrolla una arquitectura basada en términos generales en dos partes, un backend que contiene el acceso a datos y que expone de servicios REST para la consulta de éstos y por otro lado la aplicación cliente desarrollada en Android. Cabe señalar que la mayoría del backend es realizada en paralelo a la implementación de este proyecto por otros participantes dentro de Yagantec y que además realizan una versión web de Kx.

3.2. Arquitectura utilizada para la aplicación móvil

El diseño de la aplicación está centrada en dos actividades principales, por un lado una encargada del manejo de los recursos visuales junto a sus datos contextuales, y por otro lado otra que gestiona las conexiones y el procesamiento de datos. Como se aprecia en la Figura 3.1 el diseño incluye patrones entregados por el framework de Android, como lo son la sección de Resources y Activities. Cada uno de los componentes se describe a continuación:

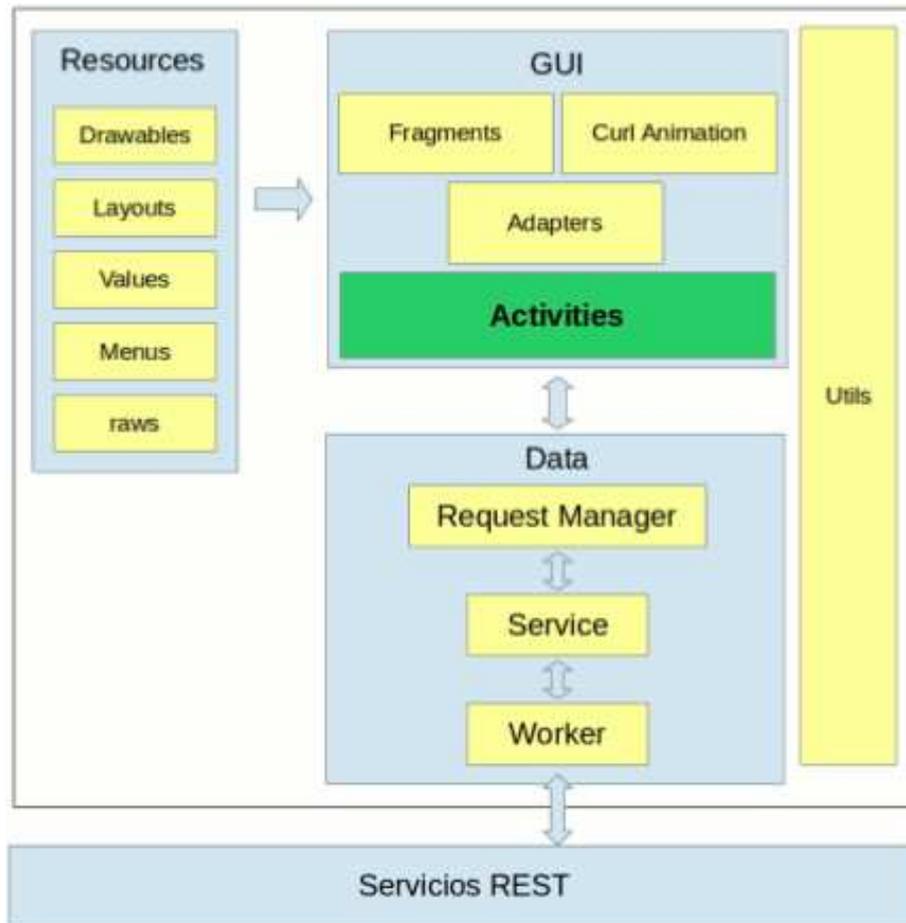


Figura 3.1: Arquitectura Plataforma Kx móvil

1. Resources

Drawables: Esta constituido por dos componentes, los recursos gráficos disponibles para la aplicación como imágenes y logos por ejemplo, y componentes en formato XML que definen formas, colores y gradientes.

Layouts: Estos recursos son los que definen las interfaces de usuario (UI) o subcomponentes de interfaces.

Menus: Se definen en esta sección todos los menús disponibles para ser usados en cualquier *Activity* que lo requiera.

Raws: Son todos los recursos de archivos para disponibilizar a la aplicación, para este caso sólo se utiliza para almacenar audios.

2. GUI

Activities: Son los componentes que proveen una interfáz gráfica donde el usuario puede interactuar y realizar acciones para controlar y usar la aplicación.

Fragments: Representan un comportamiento o una porción de una interfáz de usuario.

Curl Animation: Customización que implementa el efecto gráfico que simula un cambio de página el cual se encuentra disponible para cualquier *Activity* que lo requiera.

Adapters: Sirven para manejar un conjunto de datos y disponibilizarlos gráficamente en una interfaz de usuario.

3. Data

RequestManager: Es el encargado de manejar las peticiones de datos por parte de un *Activity*. Cuando se requiere un dato se crea un *Listener* en la *Activity* solicitante y luego al realizar la llamada el *RequestManager* instancia un servicio a través de un *Intent*.

Service: Corresponde a un servicio Android el cual puede realizar operaciones de larga duración los cuales permanecen en ejecuciones paralelas a la aplicación, incluso cuando ésta se pausa. Para este caso, recibe y extrae los parámetros necesarios de acuerdo a la petición realizada, y ejecuta la solicitud utilizando el *Worker* correspondiente.

Worker: Cada Worker se encarga de la conexión y el parseo de datos, para esto realiza peticiones GET o POST según corresponda y parsea los datos a Json utilizando la librería Gson.

4. **Utils:** Contienen distintas implementaciones que se utilizan a través de toda la aplicación. Existe un Utils que se encarga de la parte gráfica que en general es usada por las Activities para realizar el tratamiento gráfico o lógica comunes. Para el manejo de datos existe un Util que tiene implementaciones para el manejo de *Strings* de conexión y datos.

3.3. Arquitectura proyecto backoffice y versión web

Yagentec en paralelo al desarrollo de aplicación móvil, implementa el acceso a datos y lógica de negocio de Kx, junto con la versión web de Kx, la que se detalla como sigue:

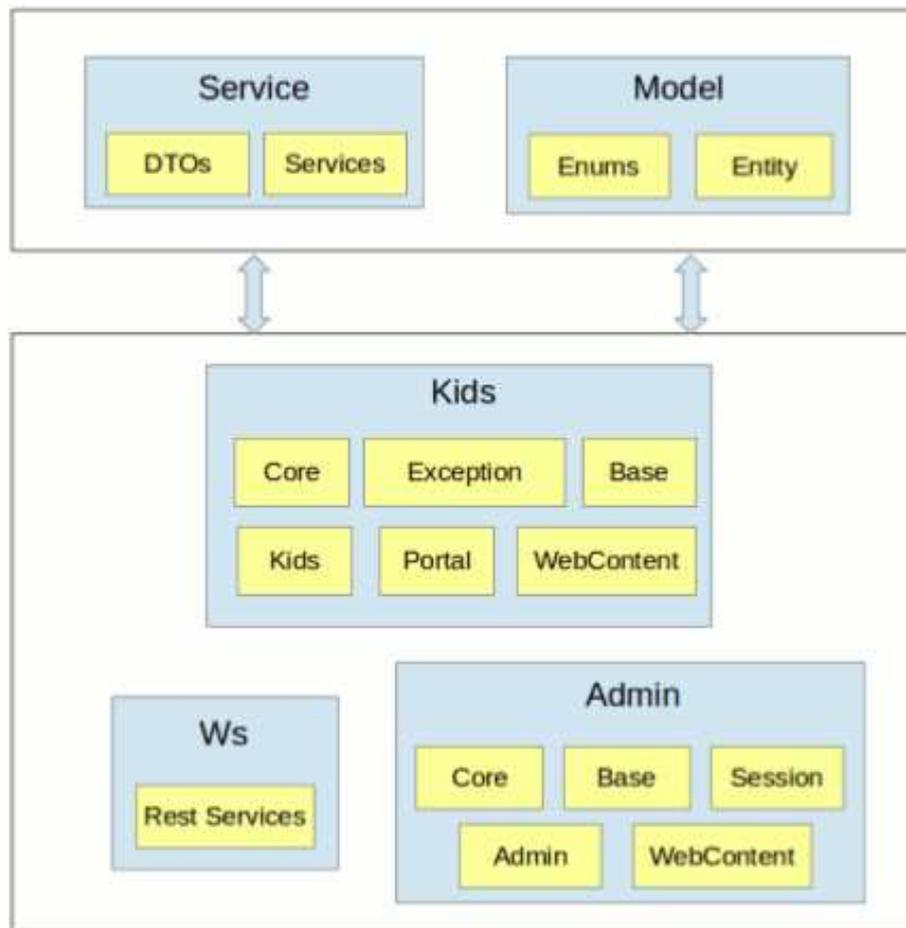


Figura 3.2: Arquitectura Plataforma Kx backend-web

1. Service

DTOs: Corresponden a todos los objetos de negocio que sirven para el transporte de datos.

Services: Implementación de servicios básicos de Kx como los *getter* y *setter*.

2. Model

Entity: Representan las entidades de base de datos usadas para Kx.

Enums: Representan las enumeraciones usadas por la aplicación.

3. Kids

Core: Corresponden a los parámetros globales de aplicación, como urls de conexión y rutas a directorios.

Exception: Manejadores personalizados para las excepciones de la aplicación.

Base: Son todos los elementos básicos de la aplicación.

Session: Es todo lo que maneja la sesión de usuario.

Kids: Contiene toda la lógica de Kx en su versión Web, es una homologación a la aplicación móvil que presenta este documento.

Portal: Corresponde a la lógica del portal que utilizarán Padres y Profesores, donde se gestiona el catálogo de libros por niño/curso y la compra de éstos.

WebContent: Engloba todo el *front-end* de Kx version Web.

4. Admin

Core: Corresponden a los parámetros globales de aplicación, como urls de conexión y rutas a directorios.

Base: Análogo al anterior, también contiene parámetros básicos en su mayoría listado de elementos.

Session: Es todo lo que maneja la sesión de usuario.

Admin: Es la lógica de administración de la aplicación, como son los mantenedores de colegios, cursos, alumnos, profesores, etc.

WebContent: Engloba todo el *front-end* del portal web de administración.

5. Ws

Rest Services: Se implementan todos los servicios necesarios para la ejecución de Kx y son expuestos como servicios REST devolviendo objetos Json.

3.4. Consideraciones previas al desarrollo

3.4.1. Disponibilización y consumo de datos

Una de las restricciones del sistema fue que todos los datos debían estar disponibles de manera centralizada, tanto en la versión para móviles, como el desarrollo simultáneo de Kx para navegadores Web, por lo que surgió la necesidad de disponibilizar los datos de una misma forma para cualquier aplicación que los requiriera, es por esto que se decidió crear un proyecto por separado para estos propósitos y así cubrir esta dependencia. Dentro de las posibilidades disponibles se decidió tomar el estándar REST + Json para la exposición, consulta y modificación de datos.

3.4.2. Sistema Operativo y dispositivos

Como es sabido los dispositivos que utilizan Android como sistema operativo son variados, tanto en sus fabricantes como en sus dimensiones y hardware que utilizan, Android contiene una interfaz que permite una integración transparente dentro de ciertos parámetros, pero esto no asegura que una aplicación sea compatible con todos los dispositivos, especialmente cuando se habla del tamaño de la pantalla, es por esto que Android define estándares para escalar una aplicación tomando en cuenta el tamaño (divididos en cuatro grupos denominados small, normal, large y extra large) y la densidad referidos como dpi (dots per inch) o puntos por pulgada, además pudiéndose diseñar interfaces personalizadas para un tamaño en específico, lo que se utilizó en este caso y que se detallará en el capítulo siguiente.

Así como existen variados dispositivos, también el sistema operativo ha evolucionado teniendo distintas versiones, es por esto que tomando en cuenta el promedio de las utilizadas en los dispositivos se decidió trabajar con Android 2.3 (Gingerbread) pero dando compatibilidad hasta la versión 4.0 (Ice Cream Sandwich).

3.4.3. Entregas a desarrollar

Para compatibilizar el desarrollo del proyecto con las necesidades de los clientes se determinó desde un principio un desarrollo inicial de una versión demo que permite por un lado tener una entrega temprana hacia los usuarios y por otro realizar las pruebas de concepto y análisis técnico necesario para el desarrollo de las funcionalidades principales de la aplicación:

- Almacenamiento y despliegue de libro digital.
- Ejecución de sonidos mientras se lee.
- Calibración de lectura
- Realizar efecto de “giro de hoja” al cambiar de página

Luego de la versión demo se desarrolla la versión completa que contempla las siguientes características:

- Login de usuario
- Visualización de libros en forma de “biblioteca”
- Lectura de libros reproduciendo sonidos
- Módulo de evaluación post lectura de capítulo/sección
- Pausa y reanudación de lectura

4. Implementación de la Solución

4.1. Estimación de horas y fechas

Para el desarrollo del proyecto se trabajó 40 horas semanales en las oficinas de Yagantec, el detalle de los tiempos para cada etapa se detallan a continuación:

1. Estudio de tecnologías y desarrollo de Modelo de la solución

Instalación de entorno de desarrollo	01/05
Estudio de aplicaciones similiares	02/05
Diseño de arquitectura de la aplicación	03/05 - 05/05
Creación de modelo de datos	06/05 - 07/05

2. Demo de Kx

Implementación de presentación	07/05
Implementación de introducción y bienvenida a Kx	09/05 - 09/05
Implementación de etapa de calibración	10/05 - 13/05
Implementación de experiencia de lectura:	
▪ Implementación de efecto de paginación	13/05 - 20/05
▪ Adaptación de libro demo a formato html	21/05
▪ Implementación de musica y efectos de sonido	22/05 - 25/05
▪ Implementación de barra de herramientas	26/05 - 29/05

Implementación de descripción de producto y lógica de servicio:

▪ Explicación de servicios para los padres	30/05 - 02/06
▪ Explicación de servicios para los profesores	30/05 - 02/06
▪ Explicación de servicios para los alumnos	30/05 - 02/06
Cierre de demo	03/06
Testing y correcciones	04/06 - 10/06

3. Kx version completa

Implementación de login al sistema	11/05 - 12/05
Implementación de revisión de lecturas anteriores	12/05 - 14/05
Búsqueda de libros:	
▪ Implementación de vista en galería de libros	15/05 - 18/05
▪ Implementación de vista en detalle de un libro	19/05 - 20/05
Lectura de libro:	
▪ Implementación de carga de libro	21/05 - 29/05
▪ Implementación efecto paginación	30/05 - 06/06
▪ Implantar música y efectos de sonido adaptados a texto	07/06 - 09/06
▪ Adaptación de barra herramientas ya creada	10/06
▪ Implementación de lectura en modo noche	11/06
▪ Implementación de vista de índice	12/06
▪ Implementación de pausa durante lectura	13/06 - 16/06
▪ Implementación de reinicio o reanudación de lectura	17/06 - 19/06
Evaluación de lectura:	
▪ Implementación de lógica de cuestionario	19/06 - 20/06
▪ Creación de plantilla	21/06
▪ Implementación despliegue de resultados	22/06 - 24/06
▪ Implementación de reforzador (para respuestas incorrectas)	24/06 - 25/06
Testing:	

■ Testing a lectura	26/06 - 28/06
■ Testing funcionalidades complementarias	28/06
Correcciones	29/06 - 31/06
Tareas complementarias/paralelas al proceso:	
■ Implementación de persistencia de datos:	
a) Realizando lectura.	
b) Al momento de realizar realizar la lectura.	
c) Al pausar y reanudar una lectura.	
■ Implementación de Web Services:	
a) Creación de interfaces de comunicación.	
b) Implementación de servicios web.	

4.2. Etapas Realizadas

4.2.1. Desarrollo Versión Demo

Bienvenida y calibración: La aplicación comienza con una vista de bienvenida la cual contiene los logos de Kx, Santillana y Samsung junto a un botón *Comenzar*. Luego se despliega un texto que explica la funcionalidad de la demo y además enseña cómo se realiza la lectura de libros, todo este proceso pasa por tres páginas y sirve para calibrar y obtener la velocidad de lectura del usuario.

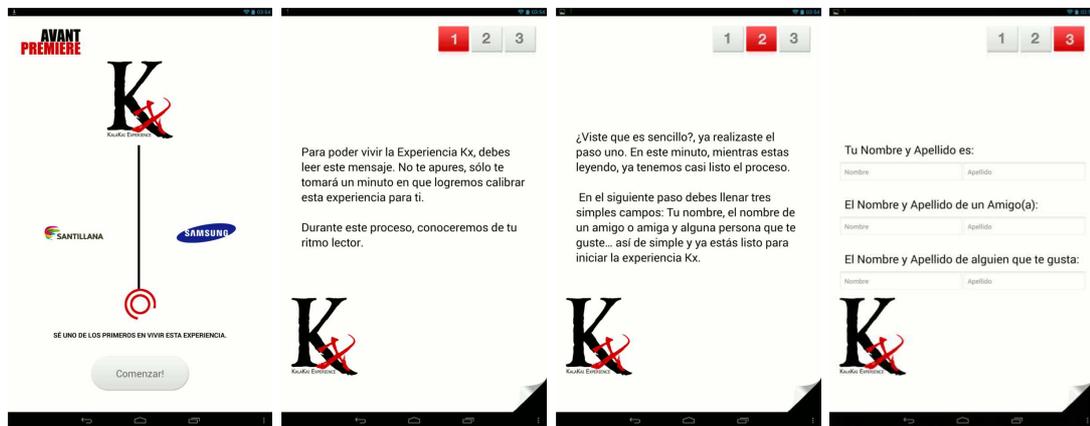


Figura 4.1: Bienvenida y Calibración

Ejecución lectura: Luego de la calibración se comienza la lectura donde se ejecutan efectos de sonido, de ambiente y música de acuerdo a los datos tomados en el paso anterior todo esto en un texto de tres páginas donde se puede apreciar la experiencia de lectura.

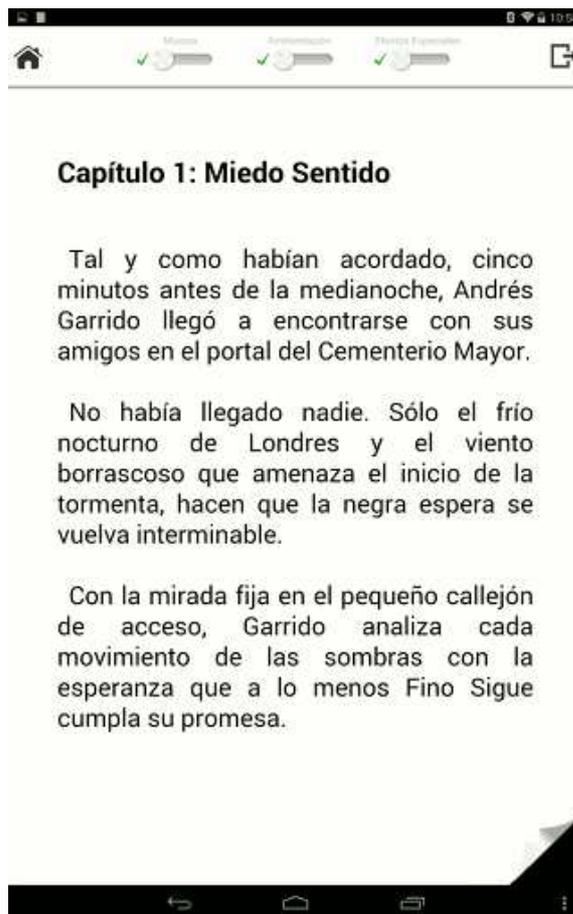


Figura 4.2: Primera página texto Demo

Cierre y videos explicativos: Luego de avanzar la última página del texto, se presenta una pantalla con tres botones denominados: *Escolares*, *Padres* y *Profesores*, cada uno de ellos lanza un video explicativo de Kx aplicado en cada uno de estos tres futuros usuarios de aplicación. Luego de que se visualizan los tres, se cierra la demo con un video de despedida y se lanza la pantalla de bienvenida de la aplicación.

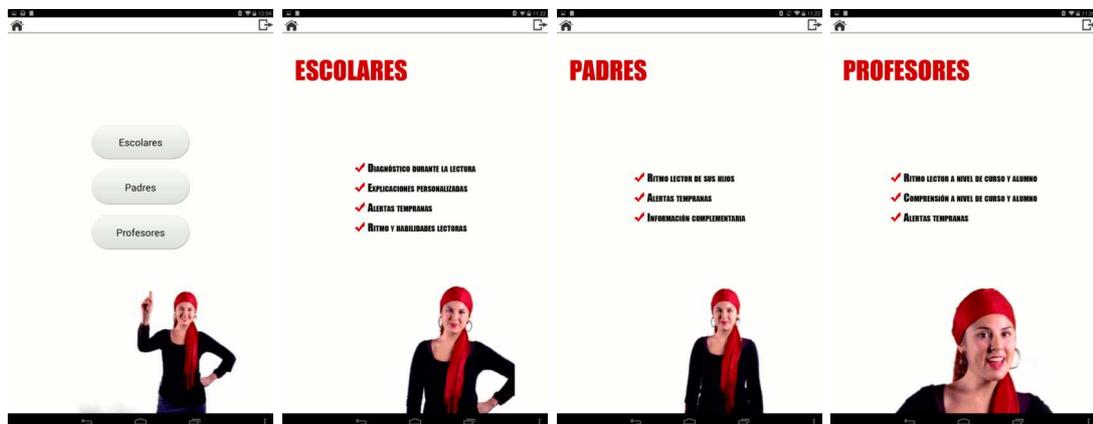


Figura 4.3: Videos Explicativos

Modelo utilizado para los datos de trazabilidad, notificaciones en la aplicación y los datos de contacto que un usuario puede enviar a través de un formulario en la aplicación

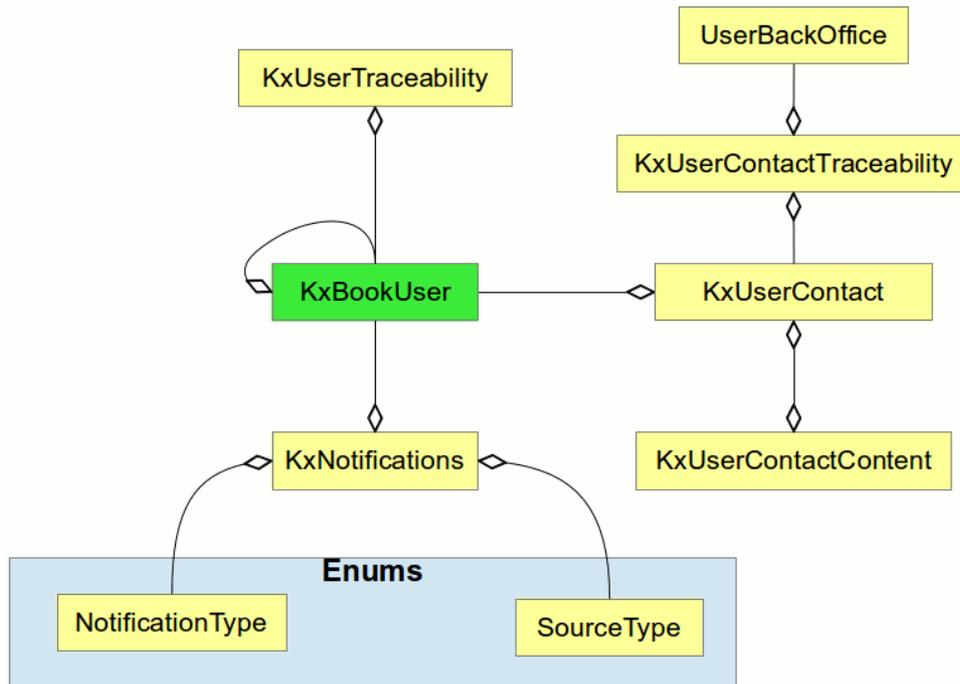


Figura 4.6: Modelo de Datos de Trazabilidad y Notificaciones

Algoritmo de Ejecución de Sonidos y Calibración: La ejecución de sonidos depende inicialmente de dos factores, la velocidad de lectura de cada usuario y de un concepto denominado *span* aplicado tanto a una página de un libro como a su capítulo y que define los lapsos a esperar antes de aplicar un sonido, esto permite generar una reproducción de audios que se puede ejecutar de acuerdo a la velocidad de lectura.

La calibración se va ajustando en cada momento de la lectura promediando la cantidad de palabras leídas por minuto. La ejecución de sonidos se produce mediante una tarea asíncrona que avanza un *span* global cada cierto lapso determinado por la velocidad de lectura y reproduce los sonidos cuando esta variable alcanza cierto valor independiente entre ellos.

```

1 private class KxBookPlayingSoundTask extends AsyncTask<KxBookSoundInfo, Void, Void> {
2     .
3     .
4     .
5     protected Void doInBackground(KxBookSoundInfo... arg0) {
6         KxBookSoundInfo soundsInfo = arg0[0];
7         int delay = initDelay(soundsInfo);
8
9         try {
10            int spanActual = 1;
11            while (spanActual <= soundsInfo.spanChapterMax) {
12                analyzeSounds(soundsInfo, spanActual);
13                spanActual++;
14                Thread.sleep(delay);
15                while (spanActual >= soundsInfo.spanPageMax) {
16                    soundsInfo.spanPageMaxFinished = true;
17                    Thread.sleep(100);
18                }
19            }
20        } catch (InterruptedException e) {
21            Log.d(DEBUG.TAG, "Tarea de reproduccion de sonidos interrumpida: " + e.getMessage());
22        }
23        return null;
24    }
25    private int initDelay(KxBookSoundInfo soundsInfo) {
26        int letrasLeidasXMinuto = soundsInfo.velLectoraPalabrasXMinuto * 8;
27        int letrasLeidasXSegundo = (int) (letrasLeidasXMinuto/60.0);
28        return (int) (1000.0/letrasLeidasXSegundo);
29    }
30 }

```

Listing 4.1: Extracto de algoritmo de ejecución de sonidos

Como se aprecia en el Código 4.1 la tarea asíncrona obtiene el *delay* necesario para controlar la velocidad de la actualización del *span* y la función *analyzeSounds(KxBookSoundInfo, int)* determina qué sonido debe ejecutarse en el *span* actual.

Implementación Sesión y Vistas por Usuario Lector y Tutor: Al ingresar a la aplicación se solicita un correo electrónico como identificador del usuario y una clave previamente creada, la que está almacenada luego de aplicarse una función SHA-256 en la base de datos. Para la aplicación existen dos usuarios, uno lector que generalmente se entiende por un alumno y un tutor o padre. El lector puede visualizar sus libros en una biblioteca, ver su avance, recibir alertas y evidentemente leer libros, el tutor sólo visualiza información de cualquiera de sus lectores asociados, su avance, alertas, y notificaciones.

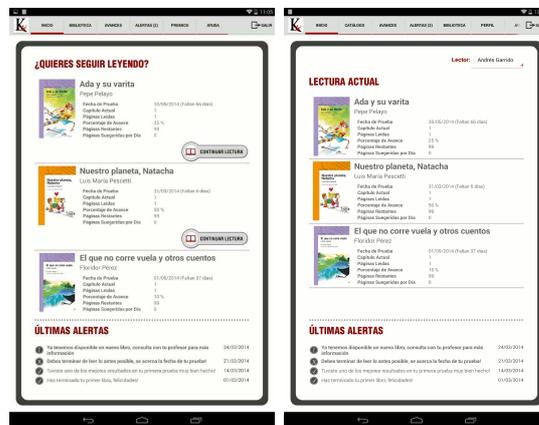


Figura 4.7: Inicio usuario lector y tutor

Despliegue Galería de Libros: Para la visualización de los libros que actualmente el usuario puede leer, se diseñó una vista de una biblioteca donde se despliegan todos los libros disponibles, que además posee una paginación cuando ésta se llena. Al presionar sobre uno de ellos se abre una vista que detalla el libro y despliega el avance en él si es que se ha comenzado.

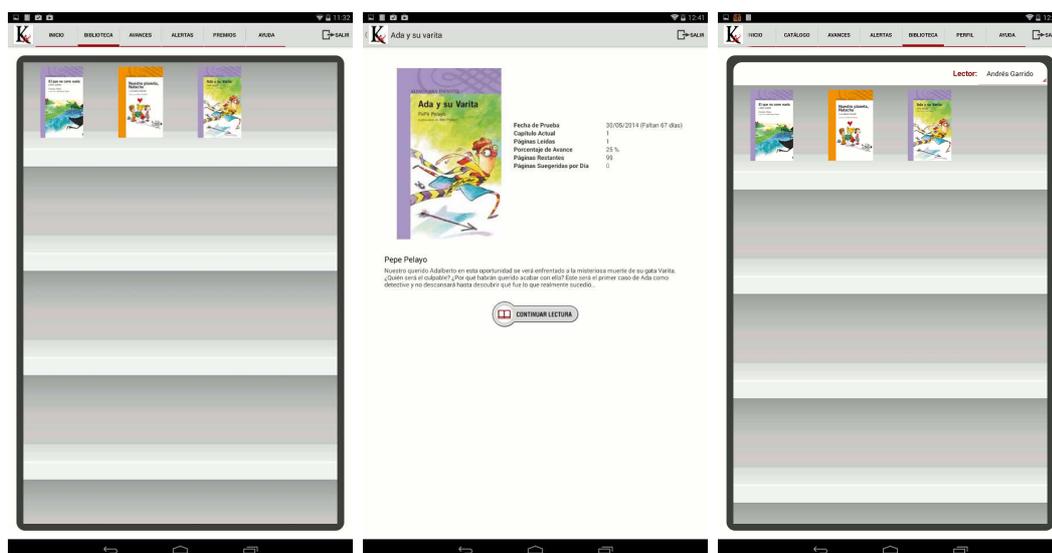


Figura 4.8: Librería lector y tutor y detalle de un libro

Implementación Barra de Herramientas: Para el despliegue del menú para el usuario se implementó una barra de herramientas pero además de esto como se buscó tener retrocompatibilidad lo máximo posible, se utilizó una librería llamada ActionBarSherlock[23] que realiza la implementación con componentes nativos en versiones antiguas y en las nuevas utiliza la ActionBar[11] de Android, lo que asegura la utilización de esta herramienta en la mayoría de los dispositivos, ésta se puede apreciar en la figura 4.9.



Figura 4.9: Menú Aplicación

Una de las funcionalidades requeridas fue controlar los audios utilizados en la aplicación: Efectos de sonido, de ambiente y música. Para esto se realizó una barra de menu la que contiene tres controles SeekBar[19] con diseño personalizados, un botón para salir y otro para volver al inicio.



Figura 4.10: Controles de lectura

Consumo y Almacenamiento de Datos: Tal como se describió en el Capítulo 3, toda la solicitud de datos al servidor parte con una llamada al *RequestManager* el que inicializa un Servicio[20] y usando una llamada POST o GET realiza la petición o envío de datos según corresponda, en donde los datos recibidos son siempre en formato JSON.

Implementación Vista de Avances: Una implementación necesaria fue una sección que desplegara los avances que tiene cada usuario dentro de los libros que posee. Para esto se creó una vista donde se despliegan los siguientes datos por libro:

- Promedio de palabras leídas por minuto
- Páginas promedio leídas por día, durante la última semana
- Cantidad de respuestas correctas e incorrectas
- Fecha de prueba del libro (en caso que aplique)
- Capítulo actual de libro
- Páginas leídas del libro
- Porcentaje de avance del libro
- Páginas restantes de libro
- Páginas sugeridas por día

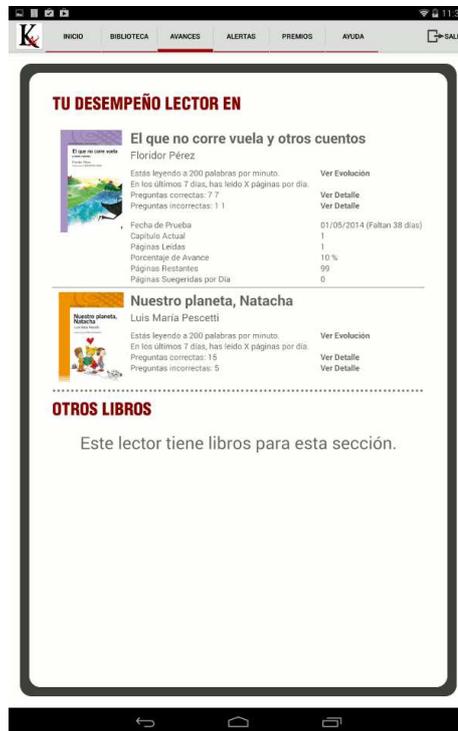


Figura 4.11: Vista de avances

Implementación Descarga Recursos: Dentro de la implementación de Kx se necesitaba la descarga de recursos como imágenes y sonidos, para esto se realizaron dos tareas asíncronas (`AsynkTask[13]`) una encargada de la descarga de imágenes, herramienta principal para la eficiencia en el despliegue de la galería de libros, la otra es la encargada de descargar los audios, que despliega un popup con el progreso de la solicitud.

Implementación Alertas y Perfil: Una de las necesidades dentro de la aplicación fue enviar notificaciones o alertas a los usuarios, con el objetivo de informar sobre evaluaciones, nuevos libros o mensajes de sistema, para esto se crearon cuatro clasificaciones de ellas: Críticas, No Críticas, Positivas e Informativas.

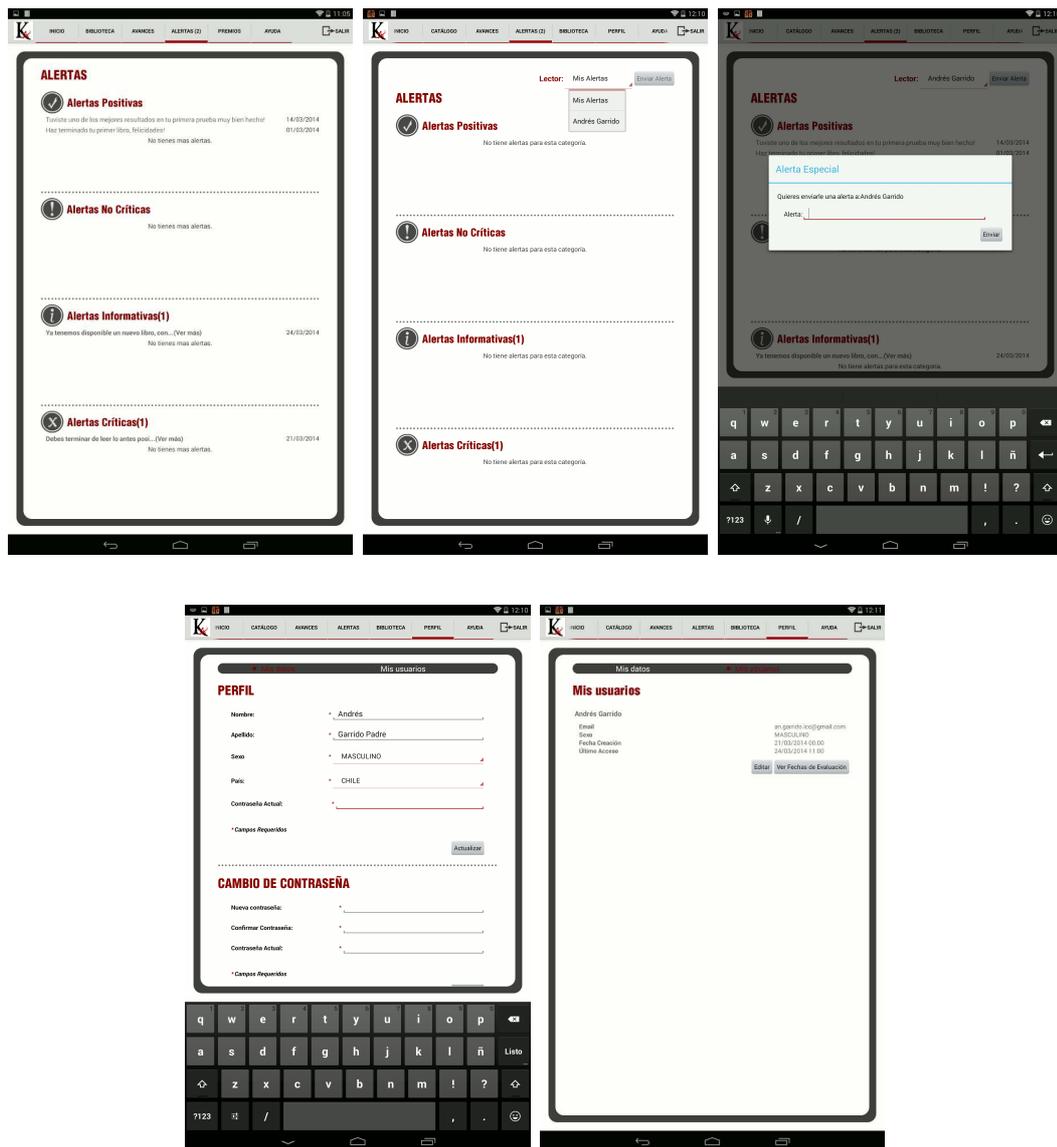


Figura 4.12: Alertas de lectores, padres y perfil

Implementación Premios: Una iniciativa que se presentó durante el proceso de desarrollo fue la posibilidad de generar premios para los niños cuando logran ciertos objetivos dentro de la lectura de un libro (completitud de cierto capítulo o respuesta de ciertas preguntas correctamente). En esta etapa sólo se implementó una vista para el despliegue de estos premios.

Implementación Formulario de Contacto y Ayuda: Otra funcionalidad requerida fue un pequeño formulario en el que se pudieran hacer preguntas sobre la aplicación o reportar problemas. Cuando se realiza una solicitud, ésta va destinada a un usuario para su resolución (dependiendo del tipo que sea) y con esto se puede generar una cola de mensajes que se le puede hacer seguimiento hasta que se corrige o aclara el problema.

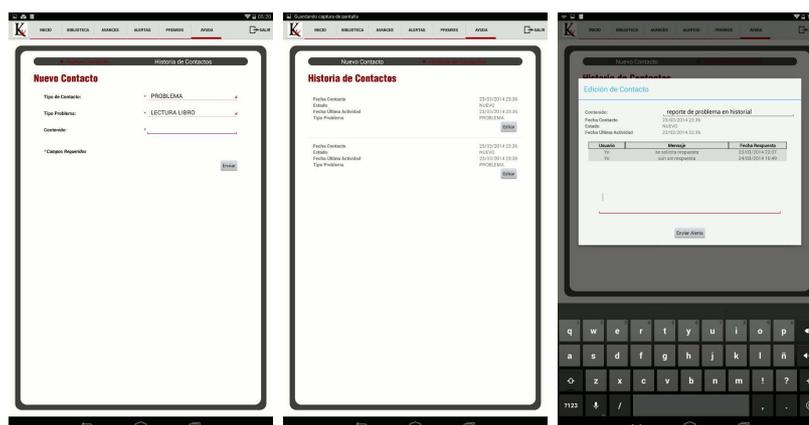


Figura 4.13: Creación y visualización de contacto

Uso de Adapters en Aplicación: Durante el proceso de desarrollo fueron utilizados distintos Adapters[12] en muchas de las vistas que se despliegan en la aplicación. Los Adapters son un puente entre un conjunto de datos que se quiere visualizar y una fuente de datos, para este caso se definieron distintos Adapters personalizados que recuperan la información necesaria y la disponibilizan, el uso general que se les da son en los listados (ListView[16]) donde un Adapter obtiene la información para cada uno de los elementos de forma independiente y usando la misma lógica para todos (ej.: Biblioteca de libros).

4.3. Problemas/Desafíos enfrentados

En la siguiente sección se describen diferentes desafíos, implementaciones particulares que se debieron realizar o dificultades que en algunos casos ralentizaron el desarrollo y en otros cambiaron decisiones iniciales sobre la forma de realizar cierta funcionalidad.

4.3.1. Implementación efecto gráfico de *cambio de hoja*

Uno de los primeros desafíos fue realizar un efecto que emulara el cambio de una página a otra como si fuera un libro. Para esto se investigaron dos librerías que ayudaron a esta acción, ya que Android no incorpora en su SDK un componente que haga esto. Uno desarrollo por *MysticTreeGames*[8] y otro por *harism*[6].

Se decidió por la primera ya que entrega una mayor flexibilidad pudiendo realizarse distintas acciones dependiendo de la necesidad y por otro lado permite animar independiente de la vista que se esté desplegando, en el segundo caso sólo se pueden cargar imágenes y que aunque la animación es mucho más fluida y real no permite trabajar con un recurso distinto.

A continuación se describen algunas características principales de cada uno:

MysticTreeGames:

- Flexibilidad en su uso, es posible animar en demanda pudiendo realizar acciones antes y despues de la animación
- Libertad en el objeto a animar ya que sólo se debe obtener un Bitmap el que se puede cargar dinámicamente
- Posibilidad de cargar cualquier tipo de Vista lo que queda independiente de la animación
- Utiliza Canvas para animación

Harism:

- Facilidad en su uso, encapsula los eventos de gestos por lo que sólo se deben cargar las imágenes a utilizar en cada página

- Eficiencia en procesado lo que provoca una animación más fluida y un mejor renderizado
- Utiliza OpenGL ES 1.0 para animación

4.3.2. Almacenamiento y despliegue de libro digital

Luego de definir la forma de realizar la animación de lectura se necesitaba establecer la forma en que se almacenaría el libro. Primeramente se investigó el formato ePub[5] que permite dar una estructura bastante correcta. Se realizó una prueba de conceptos utilizando la librería epublib[9] la que permite cargar y obtener metadatos que utiliza el libro, por ejemplo, título, autores, tabla de contenidos, entre otros. Si bien este formato resultó funcionar bastante bien habían limitaciones que ya que se requería dar un formato más específico en ciertos textos, además de complicaciones al utilizar distintas fuentes y tamaños de letras, y por otro lado se pensó que un futuro se podría requerir visualizar páginas con animaciones, por estas razones se descartó esta posibilidad.

Finalmente se decidió almacenar el libro en formato Html ya que permite una alta flexibilidad en el contenido a desplegar y por otro lado Android cuenta con un componente (WebView[21]) que permite un fácil manejo de páginas web embebidas en una aplicación.

4.3.3. Manejo de recursos multimedia

Uno de los requerimientos iniciales fue el despliegue de videos, para esto se precargan antes de su ejecución, además se realiza una pausa y reanudación de estos la cuando la aplicación está en estado de Detenida y Ejecución respectivamente. Por otro lado se tuvieron que ejecutar audios que en este caso deben ser al menos tres simultáneos: efectos de sonido(SFX), ambiente y música, por lo que se desarrolló un singleton denominado *KxBookSoundManager* que almacena distintos objetos *MediaPlayer*[17] y con esto se consigue la correcta ejecución de los audios y además el manejo de estos, pudiendo pausarse o silenciarse de manera independiente.

4.3.4. Implementación de sesión para carga y restauración de lectura

Una de las funcionalidades principales es el manejo de los datos de usuario, tanto en su avance como la velocidad de lectura, por esto se implementó una persistencia de datos en distintos niveles, listados a continuación:

- Datos de usuario: nombre, fecha de nacimiento, email, velocidad de lectura, entre otros.
- Registro de respuestas contestadas por usuario
- Evolución en velocidad de lectura en el tiempo
- Cantidad de libros que posee dentro de su biblioteca (modificada por usuario Tutor)
- Página donde el usuario se encuentra actualmente leyendo por cada libro

La aplicación dentro de la ejecución de lectura sincroniza en línea cada vez que el usuario avanza de página, y a su vez cuando la aplicación pasa a estar en estado de Pausa o Detenida, se almacenan los datos contextuales, se detiene la ejecución de sonidos y/o videos y el contador utilizado para calcular la velocidad de lectura. Luego de que la aplicación está en Ejecución la lectura se reanuda completamente.

5. Pruebas y Refactoring

5.1. Layout y Organización de Aplicación

Como se explicó anteriormente el diseño de la aplicación se hizo espacialmente para tablets de 10.1”, y aunque este es uno de los mayores tamaños entre los dispositivos existentes se tuvieron que evaluar ciertas características que tendría la aplicación y como se distribuirían para lograr el correcto funcionamiento.

Una de las primeras pruebas realizadas, fue el uso de aplicación en modalidad horizontal y vertical. Cuando se había tomado la decisión de utilizar la librería creada por *MysticTreeGames* existió una limitación al usar el dispositivo horizontalmente ya que deberían desplegarse dos hojas en vez de una, y se tomó la decisión de no invertir tiempo en desarrollar esa implementación, ya que para esta etapa no se consideró necesario:

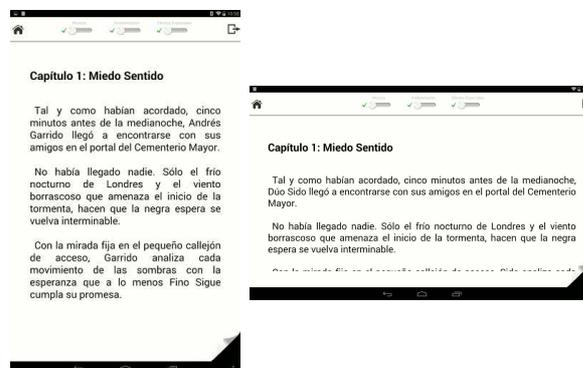


Figura 5.1: Lectura vertical y horizontalmente

Otro aspecto que se debió analizar fue el despliegue de la galería de libros, y si bien se estaba trabajando para dispositivos de gran tamaño, éste podría no ser suficiente, entonces se implementó un paginador extendido de `FragmentManagerAdapter`[15] que permite paginar ocupando un `Fragment`[14] el cual sirve para representar un elemento de una aplicación de manera genérica aplicado a un conjunto de elementos. Con esto se logra que independiente de la cantidad de libros que se tenga en la biblioteca siempre se tenga espacio para poder accederlos.

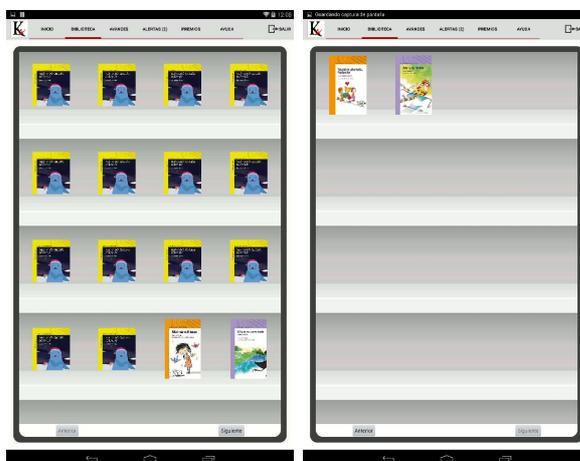


Figura 5.2: Paginación de la librería

5.2. Prueba de conexión y manejo de datos

Para el desarrollo de Kx se trabajó con un servidor dentro de la red local que contenía el modelo y recursos a utilizarse por la aplicación, esto debido a los constantes cambios que se llevaban y por la comodidad de trabajo, lo que también permitió poder realizar pruebas sobre la aplicación provocando tráfico en la conexión, interrumpiéndola o entregando incorrectamente datos. Y uno de las implementaciones más afectadas fue la lectura, ya que se utiliza la conexión a datos cada vez que se avanza en la lectura y más aun cuando se termina una sección o capítulo y se necesitan mostrar videos. Gracias a esto se agregó un control en el Servicio que consume datos, además de un control detallado de excepciones y Logs que permitieron el correcto seguimiento de la aplicación durante su desarrollo. Por otro lado se agregaron controles específicos de Android que despliegan barras de carga incrementales o indefinidos (`ProgressBar`[18]) según corresponda.

5.3. Lectura y Evaluaciones

Dentro de la lectura se encuentran varios detalles importantes que se tuvieron que probar para que funcionara de manera correcta. Una de las primeras interrogantes surgió al querer leer alguna página anterior del libro, y se decidió que un lector solamente puede volar a un capítulo anterior, donde puede moverse libremente y donde no hay reproducciones de sonido ni evaluación de la velocidad de lectura, esto quiere decir que mientras avance en la lectura no se podrá leer atrás en ese capítulo hasta que lo termine.

Una característica que se consideró fue la reproducción de libros cuando éstos ya se han leído, y para esto se decidió hacerlo de manera offline siguiendo la misma lógica del párrafo anterior. Se permite el movimiento libre dentro del libro, sin reproducción de sonido alguno y sin medición de la velocidad de lectura ni compresión lectora.

Otro aspecto fue la sección de preguntas y respuestas efectuadas cada vez que se termina cierta parte del libro. Para esto se consideró un set de preguntas en donde se selecciona cierta cantidad de ellas de manera aleatoria. Se probó el caso en que un lector responde una pregunta y se sale de la aplicación, entonces al volver a entrar se le despliegan nuevamente algunas preguntas de manera aleatoria contando la que respondió y excluyéndola para preguntar.

6. Conclusión y Análisis Retrospectivo

6.1. Análisis del producto y futuras mejoras

La aplicación cumplió con todos los requerimientos iniciales que se planificaron, se implementó primeramente una lectura con efectos de sonido que se adaptan a la velocidad del usuario, se crearon secciones de preguntas y respuestas con reforzamientos, una biblioteca para visualizar los libros disponibles de una manera más lúdica y acceso de usuarios registrados. De manera adicional se implementó una lógica de alertas, una sección de ayuda con gestión de mensajes y una vista para tutores donde se puede monitorear el avance de los lectores a cargo.

Dentro de la aplicación se implementó una pequeña sección para regalos o *premios* que básicamente consiste en contenidos descargables por el lector que se le van dando mientras avanza en sus libros.

Una idea que se evaluó y que se preñe en un futuro realizar por Yagantec es la implementación de animaciones dentro de la lectura que reemplazarían a las ilustraciones y que permitan darle a la lectura una característica más lúdica y dinámica para el lector que en este caso generalmente son niños.

Una implementación que se dejó pendiente fue un visor para el avance del lector, que idealmente sería una gráfica con la evolución de la velocidad lectora dentro de un período de tiempo, lo que se ejemplifica en la Figura 6.1.

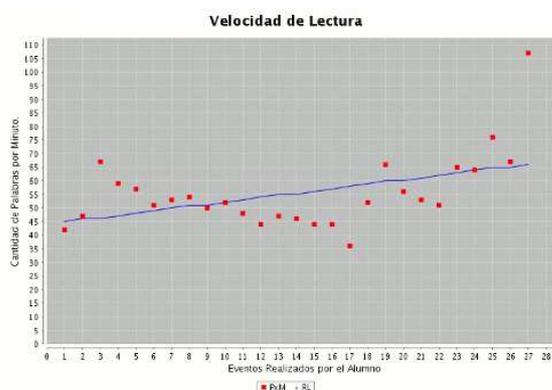


Figura 6.1: Estadística avance de lectura

Si bien cuando un tutor ingresa a la aplicación tiene un catálogo donde puede visualizar los libros disponibles para comprar, ésta es solamente una vista web dentro de la aplicación. Una posible implementación podría consistir en una versión 100% compatible con móviles capaz de realizar una consulta por libros y la adquisición de éstos para luego ser administrados dentro de sus lectores.

6.2. Análisis del proceso de desarrollo

Si bien el desarrollo se extendió bastante tiempo más de lo planificado, se trabajó para realizar un producto con correctas funcionalidades y entregando la mejor implementación posible incluso rehaciéndolas en ciertos casos.

El trabajo desarrollado en paralelo con otros integrantes de Yagantec permitió un trabajo separado donde personalmente me concentré 80% del tiempo en la implementación de la solución en Android y el restante 20% en desarrollos de lógica en el backend, exposición de todos los servicios como webs y diseño/refinamiento del modelo de datos.

Al realizarse nuevas funcionalidades como las alertas o el visor de tutores y los sucesivos ciclos en el desarrollo, produjeron un retraso que extendió la implementación hasta marzo de 2013, y aunque este tiempo fue bastante, sirvió para entregar un buen producto donde se desarrollaron distantes características y detalles que le dieron a la aplicación un diseño atractivo, nuevas funcionalidades, control en

transacciones de datos, entre otras características que justifican el desplazamiento en los tiempos de entrega.

6.3. Conclusiones finales del proyecto

Como cierre al proyecto me cabe mencionar la satisfacción tanto con la aplicación desarrollada, como también con el aprendizaje adquirido en el desarrollo, la tecnología orientada a dispositivos móviles es un área en crecimiento que incentiva al aprendizaje, ya que por un lado el paradigma de las aplicaciones tradicionales cambia porque éstas tienen un ciclo de vida, almacenamiento reducido, itinerancia e interrupción continua en el acceso a la red, y una amplia variedad de dispositivos, sólo por nombrar algunas características. Esto hace que las tecnologías móviles sean un área para desarrollarse profesionalmente, y el trabajo a tiempo completo en Kx me da las herramientas para poder seguir especializandome en estas tecnologías en mi desarrollo profesional.

Bibliografía

- [1] Open Publication Distribution System | Official Specification & Blog. <http://opds-spec.org/>, 2012.
- [2] Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013. <http://www.gartner.com/newsroom/id/2623415>, 2013.
- [3] androidteam | Android Engineering Application & Consulting Services Team. Android System Architecture Diagram. <https://code.google.com/p/androidteam/wiki/AndroidSystemArch>, 2013.
- [4] Dan Bornstein. Dalvik | Code and documentation from Android's VM team. <https://code.google.com/p/dalvik/>, 2013.
- [5] International Digital Publishing Forum. EPUB | International Digital Publishing Forum. <http://idpf.org/epub>, 2012.
- [6] harism. android_page_curl – GitHub. https://github.com/harism/android_page_curl, 2012.
- [7] Amazon.com Inc. Amazon Kindle: Welcome. <https://kindle.amazon.com/>, 2012.
- [8] MysticTreeGames. android-page-curl – GitHub. <https://github.com/MysticTreeGames/android-page-curl>, 2012.
- [9] Paul Siegmann. Epublib | A java epub library Android. <http://www.siegmann.nl/epublib/android>, 2012.

- [10] Jason Del Rey | AllThingsD. How Big Is Amazon's Kindle Business? Morgan Stanley Takes a Crack at It. <http://allthingsd.com/20130812/amazon-to-sell-4-5-billion-worth-of-kindles-this-year-morgan-stanley-says/>, 2012.
- [11] Google – Android. ActionBar | Android Developers. <http://developer.android.com/guide/topics/ui/actionbar.html>, 2012.
- [12] Google – Android. Adapter | Android Developers. <http://developer.android.com/reference/android/widget/Adapter.html>, 2012.
- [13] Google – Android. AsyncTask | Android Developers. <http://developer.android.com/reference/android/os/AsyncTask.html>, 2012.
- [14] Google – Android. Fragment | Android Developers. <http://developer.android.com/reference/android/app/Fragment.html>, 2012.
- [15] Google – Android. FragmentStatePagerAdapter | Android Developers. <http://developer.android.com/reference/android/support/v4/app/FragmentStatePagerAdapter.html>, 2012.
- [16] Google – Android. ListView | Android Developers. <http://developer.android.com/reference/android/widget/ListView.html>, 2012.
- [17] Google – Android. MediaPlayer | Android Developers. <http://developer.android.com/reference/android/media/MediaPlayer.html>, 2012.
- [18] Google – Android. ProgressBar | Android Developers. <http://developer.android.com/reference/android/widget/ProgressBar.html>, 2012.
- [19] Google – Android. SeekBar | Android Developers. <http://developer.android.com/reference/android/widget/SeekBar.html>, 2012.
- [20] Google – Android. Service | Android Developers. <http://developer.android.com/reference/android/app/Service.html>, 2012.
- [21] Google – Android. WebView | Android Developers. <http://developer.android.com/reference/android/webkit/WebView.html>, 2012.
- [22] Digital Trees. PageTurner Reader. <http://www.pageturner-reader.org/>, 2012.

- [23] Jake Wharton. ActionBarSherlock – Development. <http://actionbarsherlock.com/development.html>, 2012.