



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**“INGENIERÍA EN SISTEMAS INFORMÁTICOS”**

**TEMA:** DESARROLLO DE APLICACIÓN MÓVIL ANDROID ENFOCADA EN EL  
DESARROLLO INFANTIL

**AUTOR:** SANTIAGO MAURICIO ERAZO YÉPEZ

**TUTORA:** Mg. Tannia Mayorga Jácome

**AÑO:** 2016

## **Agradecimiento**

Agradezco a mis padres, con quienes inicié mi camino de aprendizaje de vida y educación y quienes me han apoyado a superar los obstáculos económicos, de tiempo y sobre todo de distancia con esfuerzo y mucho amor, gracias por sus palabras de aliento y por los momentos de felicidad que me regalan siempre.

A mis mentores, José Luis Landázuri y Eduardo Proaño, y a todo el grupo de Mushoq Multimedia con quienes he compartido 9 años de profesionalismo acompañado de una amistad sincera. Gracias por el tiempo y los recursos facilitados para materializar este proyecto.

A mis profesores de la Universidad Tecnológica Israel, por sus valiosas enseñanzas, por su calidad de educación, en especial a la Mg. Tania Mayorga por su apoyo y entusiasmo en el desarrollo de este proyecto.

A mi familia política por el ánimo para continuar con mi carrera y por compartir la alegría de cada triunfo.

Agradezco de sobre manera a mi amada esposa Verónica, quien con su amor, ternura, paciencia y dedicación se ha convertido en el pilar más fuerte de mi existencia. Quien me dio la fuerza y la inspiración para seguir superándome como profesional y con quien he compartido hermosos años de felicidad y éxitos, superando cada reto que nos pone la vida.

## **Dedicatoria**

A mi hijo Martín, mi Martín valiente de ternura infinita, sonrisa hermosa y mirada profunda, te dedico este trabajo producto de mi inspiración por ti, porque lo pensé para ti, lo creé para ti. Espero que Nina lo disfrute tanto como tú.

## **Resumen**

El presente trabajo describe los componentes y el proceso de desarrollo de la aplicación móvil “TinPoint”, construida en código nativo Android y que permite tener control sobre compensaciones en el hogar por el buen comportamiento de los niños y niñas. Se utilizó la metodología de desarrollo ágil XP (Extreme Programming) a través de la cual se determinaron los requerimientos y alcances como la versión del sistema operativo, tamaños de pantalla y densidades, versiones e idiomas. El lenguaje de programación utilizado fue Java con librerías nativas y externas de Android empleando el entorno de desarrollo Android Studio. La aplicación se publicó en la tienda de aplicaciones Google Play y se elaboró un manual sobre el entorno de Google Play Developer Console.

Finalmente, la herramienta Google Analytics que presenta diferentes estadísticas e información agrupada sobre el tráfico, audiencia, adquisición, comportamiento y conversiones, fue instalada en la aplicación “TinPoint”, facilitando información para el mejoramiento o actualización de la misma.

Palabras clave: desarrollo, aplicación móvil, código nativo, Android, metodología XP, Android Studio, Google Play, Google Analytics.

## **Abstract**

This paper describes the components and the development process of "TinPoint" mobile application, built in Android native code, it allows to have control over compensation at home for children's good behavior. The methodology used for this development was XP (Extreme Programming), which allowed to determine requirements and the operating system scope, as well as screen sizes, densities, versions and languages. The programming language used was Java with native and external Android libraries using the Android Studio development environment. The app was published in the Google Play store applications and a manual on the environment Google Play Developer Console was developed.

Finally, the Google Analytics tool presents different statistics and information grouped on traffic, audience acquisition, behavior and conversions, which were installed as part of the "TinPoint" application, providing information for improving or updating it.

Keywords: development, mobile application, native code, Android, Extreme Programming, Android Studio, Google Play, Google Analytics.

## Índice

Introducción .....	1
Objetivos.....	2
Objetivo general.....	2
Objetivos específicos.....	2
Fundamentación teórica.....	3
Metodología de desarrollo .....	4
Fases de la metodología XP.....	5
Requerimientos y alcances de la aplicación .....	6
Versión de Android. ....	6
Tamaño de pantalla y densidades. ....	7
Uso.....	7
Versión.....	7
Diseño de interfaz y experiencia de usuario .....	8
Wireframes .....	8
Interfaz de usuario (UI) y experiencia de usuario (UX).....	9
Construcción en código nativo android.....	10
Herramientas de desarrollo .....	10
Android Studio.....	10
Lenguaje Java.....	10
Librerías utilizadas.....	11
App Lock .....	11
UCrop.....	12
CircleImageView .....	13
Google Play Services Analytics .....	14
Código Fuente .....	15

Estructura de archivos .....	15
Programación de diseño de pantallas .....	16
Hojas de estilo.....	16
Pantallas listas.....	17
Publicación de la aplicación en Google Play .....	18
Google Play Developer Console .....	18
Pasos para crear una cuenta de desarrollador .....	18
Tus Aplicaciones.....	18
Detalles de tu cuenta.....	19
Ficha de Play Store .....	20
Información del producto .....	20
Elementos gráficos.....	21
Categorización .....	22
Datos de contacto .....	23
Políticas de privacidad .....	23
Pruebas alfa y beta .....	24
Configuración de APK de producción.....	25
Resultados de estadísticas de Google Analytics.....	26
Audiencia.....	26
Datos demográficos.....	27
Edad.....	27
Sexo.....	27
Idioma .....	28
País.....	29
Usuarios recurrentes vs usuarios nuevos .....	31
Versiones del sistema operativo y operadores de red .....	31
Visión general del comportamiento .....	32

Comportamiento en pantallas .....	32
Flujo del comportamiento.....	33
Eventos .....	34
Conclusiones.....	35
Recomendaciones.....	37
Bibliografía .....	38



## Índice de gráficos

<b>Gráfico 1</b> - Fases de XP fuente (Figueroa, Solís, & Cabrera, 2008) .....	5
<b>Gráfico 2</b> – Uso de versiones de Android .....	6
<b>Gráfico 3</b> – Densidades de pantallas Fuente (Android Developer Guide, 2016) .....	7
<b>Gráfico 4</b> - Wireframes .....	8
<b>Gráfico 5</b> - Catálogo de formas gráficas .....	9
<b>Gráfico 6</b> - Bloqueo App Lock .....	11
<b>Gráfico 7</b> - Recorte de imagen con UCrop fuente (Yalantis, 2016) .....	12
<b>Gráfico 8</b> - Imagen de perfil CircleImageView Fuente (Dodenhof, 2016).....	13
<b>Gráfico 9</b> - Google Analytics Fuente (Google Analytics, 2016) .....	14
<b>Gráfico 10</b> - Estructura de archivos.....	15
<b>Gráfico 11</b> - Pantallas terminadas.....	17
<b>Gráfico 12</b> - Tus aplicaciones Fuente (Google Play Developer Console, 2016).....	19
<b>Gráfico 13</b> - Datos de perfil Fuente (Google Play Developer Console, 2016).....	19
<b>Gráfico 14</b> - Información de la aplicación Fuente (Google Play Developer Console, 2016) .....	20
<b>Gráfico 15</b> - Elementos gráficos de la ficha de aplicación Fuente (Google Play Developer Console, 2016) .....	21
<b>Gráfico 16</b> – Categorización Fuente (Google Play Developer Console, 2016).....	22
<b>Gráfico 17</b> - Datos de contacto Fuente (Google Play Developer Console, 2016)....	23
<b>Gráfico 18</b> - Políticas de privacidad Fuente (Google Play Developer Console, 2016) .....	23
<b>Gráfico 19</b> - Pruebas alfa y beta Fuente (Google Play Developer Console, 2016)..	24

<b>Gráfico 20</b> - Configuración de APK de producción Fuente (Google Play Developer Console, 2016).....	25
<b>Gráfico 21</b> – Usuarios totales Fuente (Google Analytics, 2016).....	26
<b>Gráfico 22</b> – Edad Fuente (Google Analytics, 2016) .....	27
<b>Gráfico 23</b> - Sexo Fuente (Google Analytics, 2016) .....	27
<b>Gráfico 24</b> – Idioma Fuente (Google Analytics, 2016).....	28
<b>Gráfico 25</b> - Mapa países Fuente (Google Analytics, 2016).....	29
<b>Gráfico 26</b> - Lista países Fuente (Google Analytics, 2016) .....	30
<b>Gráfico 27</b> - Usuarios recurrentes vs usuarios nuevos Fuente (Google Analytics, 2016) .....	31
<b>Gráfico 28</b> - Versiones del SO y operadores de red Fuente (Google Analytics, 2016) .....	31
<b>Gráfico 29</b> – Interacción Fuente (Google Analytics, 2016).....	32
<b>Gráfico 30</b> - Flujo Fuente (Google Analytics, 2016).....	33
<b>Gráfico 31</b> – Eventos Fuente (Google Analytics, 2016).....	34

## INTRODUCCIÓN

La Real Academia de la Lengua (Española, 2014) afirma: “Disciplina doctrina, instrucción de una persona, especialmente en lo moral”. La disciplina es además el conjunto de reglas de comportamiento que en los niños se halla profundamente relacionado con su educación, castigos, recompensas y su comportamiento en general.

Los padres y educadores hacen uso de recompensas con el deseo de lograr que los niños adopten el tipo de conducta que se considera correcto, por lo cual se podría afirmar que las recompensas son auxiliares para establecer un método de compensación, así como lo son las buenas calificaciones.

Para el 2016 se prevé que el número de usuarios de teléfonos inteligentes, alcance los 2.08 mil millones (The Statistics Portal, 2016), lo que representa una ventaja para acceder a sistemas, como la aplicación a ser desarrollada que aporta las herramientas que facilitarían establecer compensaciones dentro del hogar para premiar a los niños, sobre la base de cuantificar metas, que se verán reflejadas en acciones de los niños.

El presente proyecto está enfocado en el desarrollo de una aplicación móvil Android en lenguaje nativo que permita tener control sobre compensaciones en el hogar por el buen comportamiento de los niños, la misma que dentro sus propósitos contempla el estimular y consolidar la aparición de determinados tipos de conducta apropiados y positivos, al presentarle al niño una situación agradable, al alcanzar su meta. Cabe indicar que existen estudios psicológicos que han confirmado que la recompensa obtiene resultados más eficaces y perdurables sobre la conducta infantil que el escarmiento.

## **OBJETIVOS**

### **Objetivo general**

Desarrollar la aplicación móvil Android en código nativo "TinPoint"

### **Objetivos específicos**

- Analizar los requerimientos y alcances de la aplicación
- Diseñar la interfaz y experiencia del usuario (UX/UI)
- Construir la aplicación en código nativo Android
- Publicar la aplicación en Google Play
- Comprobar estadísticas de uso a través de la herramienta Google Analytics

## FUNDAMENTACIÓN TEÓRICA

El presente trabajo implica el desarrollo de una aplicación móvil Android que puede ser utilizada, entre otras cosas para identificar procesos de cambios conductuales en los niños, por lo cual es necesario realizar una breve explicación de las teorías conductistas del aprendizaje, sustentadas por sus más destacados representantes: Thorndike y Skinner.

Thorndike postuló “leyes” del aprendizaje, de las cuales las más notables son las del ejercicio y el efecto.

De acuerdo con la **ley del ejercicio**, la repetición de una respuesta condicionada fortalecería el vínculo estímulo-respuesta. Según la **ley del efecto**, “las respuestas seguidas por una recompensa se fortalecerían y las respuestas seguidas por un castigo se debilitarían”. (Good & Brophy, 1995) .

Skinner usó el término **condicionamiento operante** que aplica principios del reforzamiento para condicionar o moldear conductas operantes, las mismas que son “respuestas voluntarias, no producidas de manera automática por algún estímulo conocido, que son usadas para operar sobre el ambiente. (Good & Brophy, 1995).

El mecanismo principal para lograr el moldeamiento del comportamiento es el reforzamiento contingente el cual implica reforzar la ejecución de una conducta blanco gestionando reforzamientos sólo cuando la conducta ha sido ejecutada conforme a algún criterio.

En un programa de **contrato de contingencias**, el maestro (tutor, padres, guías) diseña un contrato individual con cada estudiante, describiendo exactamente qué debe hacer éste para ganar un privilegio particular o una recompensa. En algunos programas, los estudiantes participan en las decisiones de cuáles conductas serán reforzadas y las recompensas que pueden ganarse. El proceso mismo de negociación es una experiencia educativa, puesto que los estudiantes aprenden a fijar metas razonables y a sujetarse a los términos de un contrato. (Craig & Woolfolk, 1990).

El presente trabajo está orientado al desarrollo de una aplicación móvil Android, con la cual los niños pueden obtener una recompensa, un deseo, que se han propuesto como meta, por lo que la misma estimula a los niños a realizar buenas acciones y a mantener un buen comportamiento

Si un comportamiento produce el efecto que deseamos, entonces nos sentimos inclinados a repetirlo. En otras palabras, el comportamiento depende en gran medida de sus consecuencias. Obtener recompensas –Reforzamiento positivo- es uno de los instrumentos más valiosos que los educadores tienen a su disposición, pero ellos (los maestros), para ser efectivos, deben tomar en cuenta lo referente a la frecuencia de las recompensas, en momento oportuno de las mismas y el tipo de ellas que quiere emplear. (Graig, Mehrens, & Clarizio, 1988).

De esta manera se describe el marco psicológico del presente trabajo técnico (desarrollo software), cabe mencionar que el 82.8% del total de usuarios de teléfonos inteligentes, utilizan dispositivos con el sistema operativo Android (Future, s.f.), lo cual facilita el acceso directo a la aplicación que puede llegar a tener un alcance a nivel global.

## **Metodología de desarrollo**

Dentro de las metodologías ágiles para el desarrollo de software el Extreme Programming “es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, ... XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes.” (Canós, Letelier, & Penadés, 2003). Esta metodología de desarrollo es la más adecuada a ser utilizada en este proyecto debido al tiempo de ejecución y es la más utilizada para la planeación, diseño, codificación y pruebas.

## Fases de la metodología XP

“Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.” (Figueroa, Solís, & Cabrera, 2008). Bajo esta premisa el proyecto se desarrolló utilizando las fases de XP de manera cíclica.



**Gráfico 1** - Fases de XP  
fuente (Figueroa, Solís, & Cabrera, 2008)

## REQUERIMIENTOS Y ALCANCES DE LA APLICACIÓN

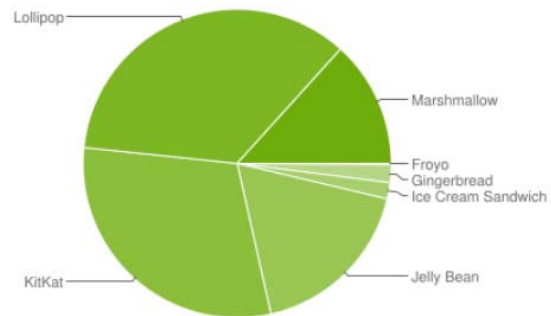
### Versión de Android.

Para el uso de la aplicación a ser desarrollada, se requiere un dispositivo con sistema operativo Android, desde la versión 4.0 en adelante.

En versiones anteriores a la 4.0 existen funciones y librerías, que serán utilizadas en el desarrollo de la APP, que se encuentran obsoletas y no es posible utilizarlas, además deja la posibilidad de seguir actualizando con librerías nuevas disponibles.

Desde el API 14 (Android 4.0) en adelante, se tiene un uso del 97.3% de los dispositivos a nivel mundial (Dashboards – Android Developers, 2016).

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.7%
4.1.x	Jelly Bean	16	6.4%
4.2.x		17	8.8%
4.3		18	2.6%
4.4	KitKat	19	30.1%
5.0	Lollipop	21	14.3%
5.1		22	20.8%
6.0	Marshmallow	23	13.3%



**Gráfico 2** – Uso de versiones de Android  
Fuente (Dashboards – Android Developers, 2016)



## Tamaño de pantalla y densidades.

La aplicación está desarrollada para ser utilizada en Smartphone o Tablet Android sin importar el tamaño del dispositivo; la configuración de estilos se acopla a las densidades de los diferentes dispositivos.

	Low density (120), <i>ldpi</i>	Medium density (160), <i>mdpi</i>	High density (240), <i>hdpi</i>	Extra-high-density (320), <i>xhdpi</i>
<i>Small</i> screen	QVGA (240x320)		480x640	
<i>Normal</i> screen	WQVGA400 (240x400) WQVGA432 (240x432)	HVGA (320x480)	WVGA800 (480x800) WVGA854 (480x854) 600x1024	640x960
<i>Large</i> screen	WVGA800** (480x800) WVGA854** (480x854)	WVGA800* (480x800) WVGA854* (480x854) 600x1024		
<i>Extra-Large</i> screen	1024x600	WXGA (1280x800) <sup>†</sup> 1024x768 1280x768	1536x1152 1920x1152 1920x1200	2048x1536 2560x1536 2560x1600

**Gráfico 3** – Densidades de pantallas  
Fuente (Android Developer Guide, 2016)

## Uso

La aplicación no solo está orientada a que sea usada en el hogar; a esta se le puede dar uso académico en un centro de desarrollo infantil o la escuela, con interacción entre profesor y alumno.

## Versión

La aplicación está publicada para su descarga gratuita, sin restricciones de dispositivos, y puede ser descargada desde cualquier parte del mundo.

La aplicación de forma nativa está en idioma inglés para todos los dispositivos y en español para los que usen este idioma en la configuración del dispositivo.

# DISEÑO DE INTERFAZ Y EXPERIENCIA DE USUARIO

## Wireframes

En la primera etapa de la construcción de la aplicación, se definieron los Wireframes para el flujo del User Experience (UX).

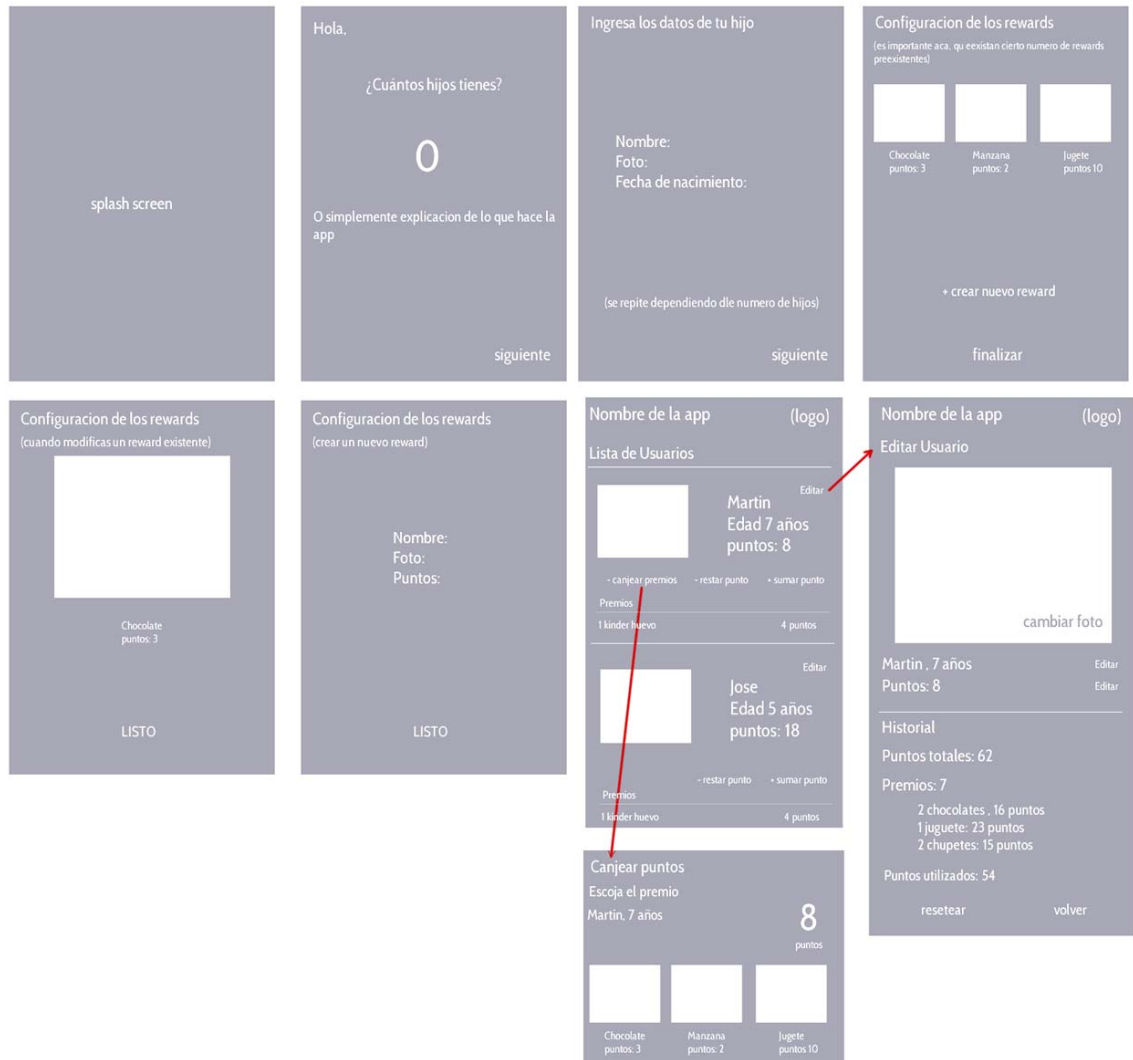


Gráfico 4 - Wireframes

## Interfaz de usuario (UI) y experiencia de usuario (UX)

Usualmente la interfaz y la experiencia de usuario están dirigidas a un solo individuo que usa una aplicación, en este caso en particular, al ser una aplicación que usa un adulto y es de interés de un niño, este concepto se separó y se realizó un diseño gráfico UI, usando colores y elementos pensando en el agrado de los niños y niñas, mientras que la experiencia de usuario UX que se define como “la percepción de una persona y las respuestas subjetivas de esta como resultado de la utilización y/o el uso de un producto, sistema o servicio” (ISO/IEC, 2010). Está ligada al usuario tutor y cumple con las expectativas de lineamiento, orden y secuencia. Cabe destacar que la aplicación es inclusiva, pues permite a las personas con discapacidad utilizarla de manera efectiva y ágil mediante la correcta configuración de herramientas de accesibilidad propias del sistema operativo Android.

A continuación se muestra el catálogo de las formas gráficas que se elaboró para ser utilizado durante el desarrollo de la aplicación.



Gráfico 5 - Catálogo de formas gráficas

## CONSTRUCCIÓN EN CÓDIGO NATIVO ANDROID

### Herramientas de desarrollo

#### Android Studio

El entorno de desarrollo Android Studio es el más óptimo al momento de construir una aplicación Android, al ser el IDE oficial de Google para el desarrollo de Android, nos brinda un mejor entorno, nuevas herramientas y mejor estructura de archivos, además que su ritmo de actualizaciones es bastante alto.

Actualmente en Android Studio la instalación y uso de librerías externas se simplifica gracias a la compilación por Gradle que es el siguiente paso evolutivo en herramientas de construcción basados en JVM. Se basa en herramientas establecidas como Ant y Maven y toma sus mejores características llevándolas al siguiente nivel “siguiendo un enfoque de construcción por convención debido a que Gradle es nativo de JVM, le permite escribir lógica personalizada en el idioma que le resulte más cómodo, ya sea Java o Groovy.” (Muschko, 2014)

#### Lenguaje Java

“SUN caracteriza a Java como un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, securizado, independiente de las arquitecturas, portable, eficaz, multihilo y dinámico” (Groussard, 2012)

Java fue elegido como el lenguaje para el entorno de desarrollo de Android, el sistema operativo móvil líder en smartphones y tablets. Android es por tanto el sistema operativo (es una versión de Linux) y Java el lenguaje utilizado para crear apps en él. (Curso Java para Android y Programación Orientada a Objetos, 2010)

## Librerías utilizadas

Programando de forma nativa se puede desarrollar todo lo que se necesita para cualquier aplicación pero existe mucha información o herramientas adicionales que facilitan ciertos aspectos de la programación y sobre todo acortan el tiempo de trabajo. Actualmente existe una gran disponibilidad de herramientas y librerías externas para ser utilizadas de forma gratuita dentro del desarrollo de aplicaciones, una gran cantidad de desarrolladores liberan librerías, herramientas o simplemente líneas de código de las cuales se puede hacer uso.

En el presente desarrollo se incluyeron algunas librerías externas detalladas a continuación.

### App Lock

Es una librería simple para bloquear y desbloquear las actividades (por ejemplo, bloqueo para niños) con un código PIN.

El objetivo de App Lock es permitir a los usuarios introducir y confirmar un PIN para bloquear temporalmente la aplicación que se utilice, hasta que se re-introduce el PIN por parte del usuario.

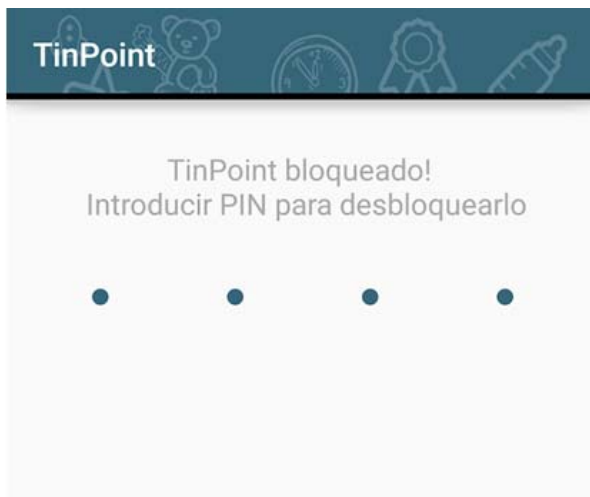
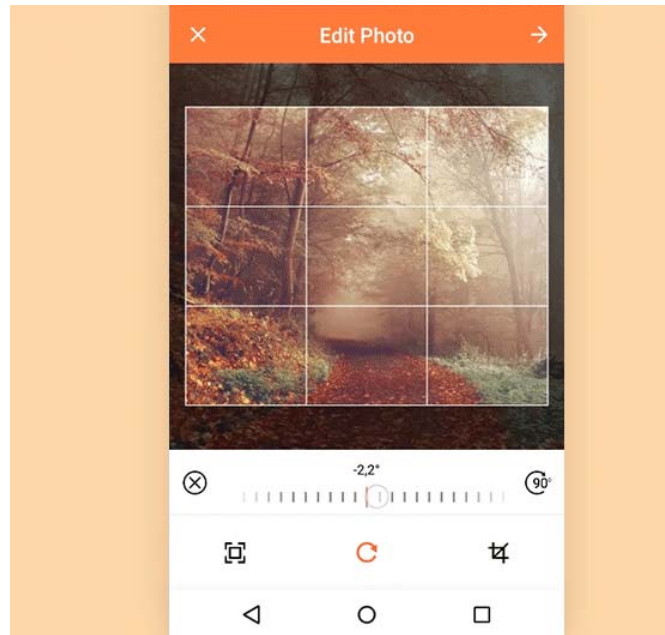


Gráfico 6 - Bloqueo App Lock

## UCrop

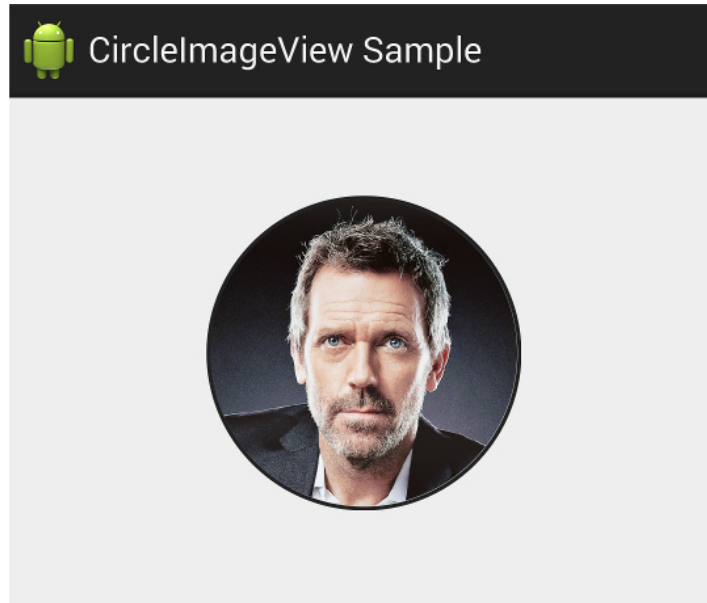
UCrop es una librería que permite recortar las imágenes para su uso posterior en Android. Sus características incluyen la ampliación de imágenes, rotación de imágenes, cambiar la relación de aspecto, el apoyo a los gestos táctiles y más.



**Gráfico 7** - Recorte de imagen con UCrop  
fuente (Yalantis, 2016)

## CircleImageView

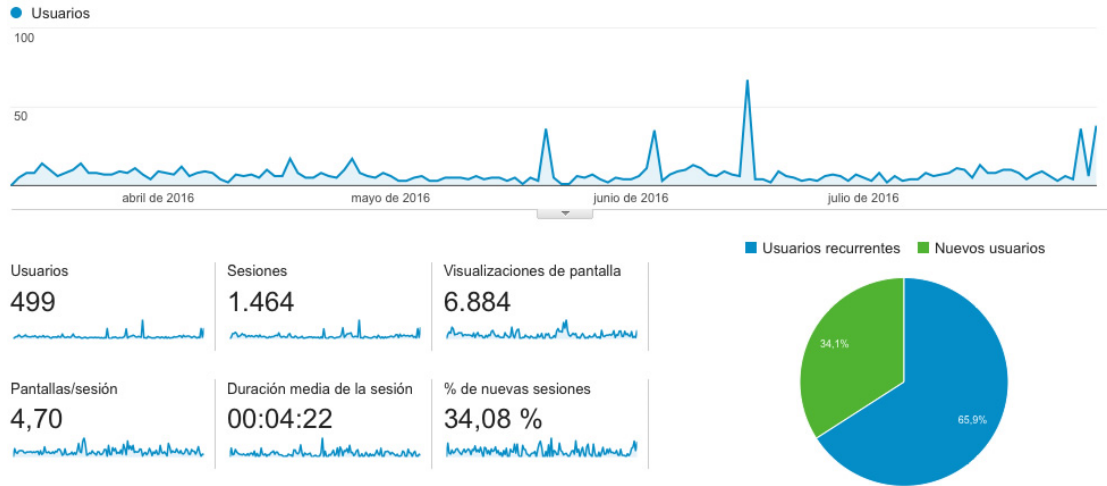
Es una librería que permite obtener una imagen de forma rápida, circular perfecta para un perfil.



**Gráfico 8** - Imagen de perfil CircleImageView  
Fuente (Dodenhof, 2016)

## Google Play Services Analytics

Esta librería permite utilizar las herramientas de Google Analytics para medir la actividad de los usuarios en las pantallas o actividades de nuestra aplicación.



**Gráfico 9** - Google Analytics  
Fuente (Google Analytics, 2016)



## Código Fuente

Referirse al Anexo 1.

## Estructura de archivos

La siguiente imagen muestra la estructura de archivos java y layout xml de la aplicación “TinPoint” desarrollados en Android Studio.

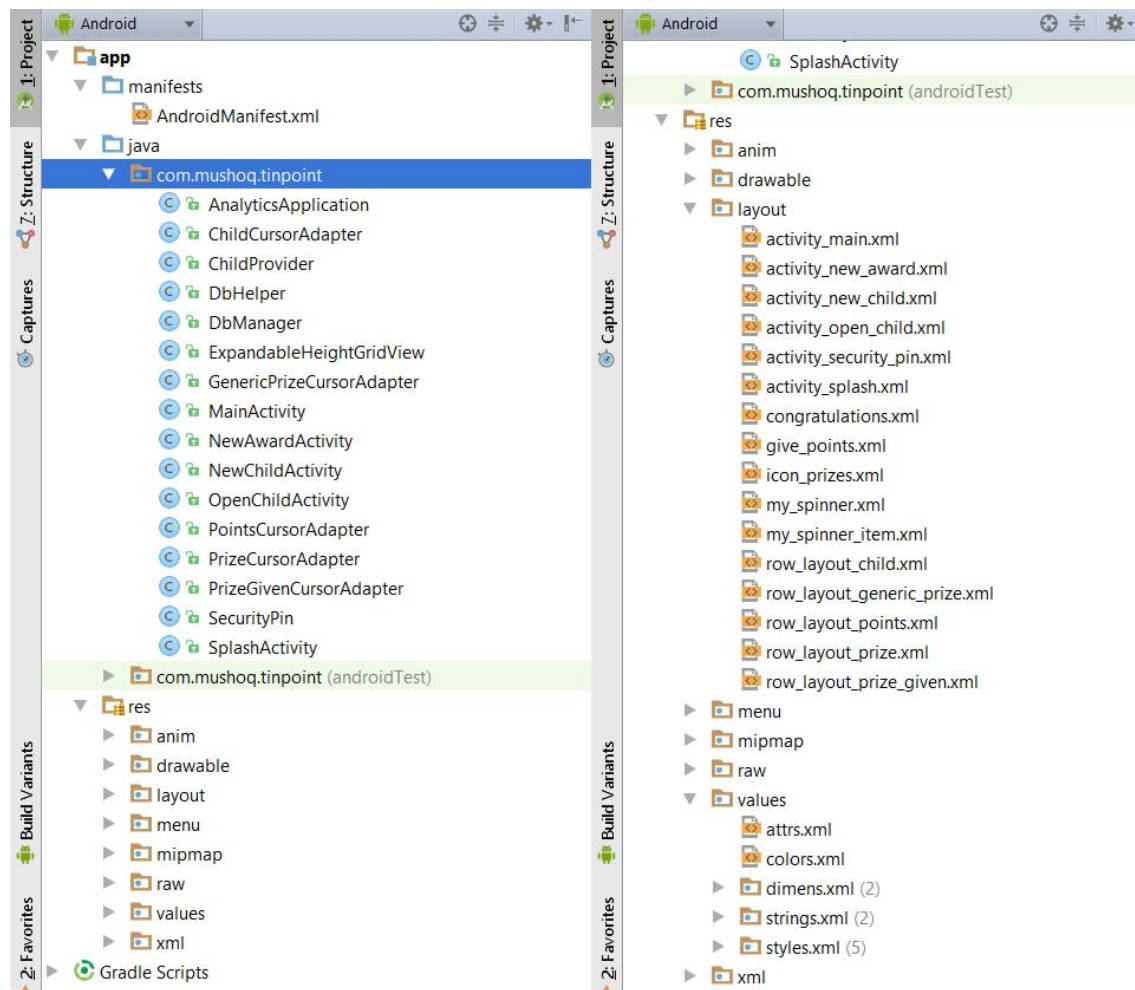


Gráfico 10 - Estructura de archivos

## **Programación de diseño de pantallas**

Los archivos de recursos visuales xml para el layout mantienen por separado una gran cantidad de código de las clases java principales. Esto aumenta la comprensión de la estructura del proyecto y reduce la cantidad de tiempo para el diseño de la interfaz de usuario.

Android Studio dispone de un editor visual para crear diseños en tiempo real, observando los cambios de la interfaz para ajustar cambios. Este editor se presenta como un emulador y podemos cambiar de tamaño de pantalla o versión de sistema operativo para verificar las visualizaciones en diferentes dispositivos.

### **Hojas de estilo**

Los elementos de un archivo Layout pueden tener diferentes propiedades pero es posible independizarlos del diseño a través de un archivo de recurso de estilos. “Un estilo es una colección de propiedades que definen el formato que tendrá una vista. Podemos especificar cosas como tamaño, márgenes, color, fuentes, etc. Un estilo se define en ficheros XML diferente al fichero XML Layout que lo utiliza.” (Gironés, 2012)

Utilizar hojas de estilos permite configurar varias densidades de pantallas para la visualización de la aplicación en diferentes dispositivos (teléfonos pequeños, teléfonos grandes, tablets pequeñas y tablets grandes) y que en todos estos la lectura y la interacción del usuario sea lo más cómoda posible.

## Pantallas listas

A continuación se muestra el resultado de las capturas de pantallas programadas con estilo gráfico listo.

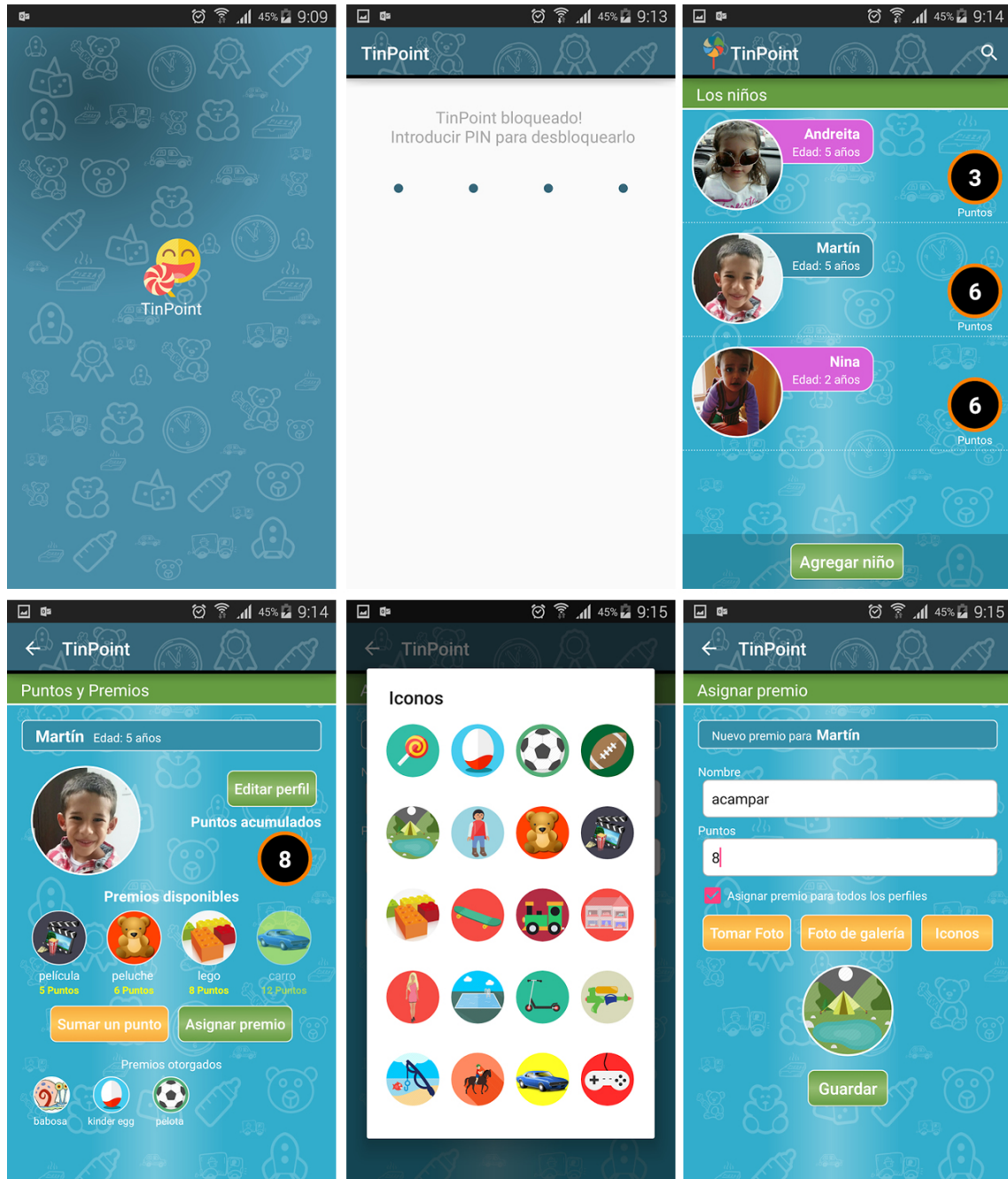


Gráfico 11 - Pantallas terminadas

## **PUBLICACIÓN DE LA APLICACIÓN EN GOOGLE PLAY**

### **Google Play Developer Console**

Google ofrece una serie de herramientas para facilitar la publicación de aplicaciones la tienda Google Play.

La Consola para desarrolladores de Google Play es el lugar donde realizas las operaciones de publicación y donde tienes tus herramientas. Carga aplicaciones, crea páginas para tus productos, configura precios y distribución, y publica. Puedes administrar todas las fases de publicación en Google Play a través de la Consola para desarrolladores, desde cualquier explorador web. (Google Inc., 2016)

### **Pasos para crear una cuenta de desarrollador**

- Crear un perfil de desarrollador en Google Play Developer Console <https://play.google.com/apps/publish/signup/> utilizando una cuenta de Gmail
- Aceptar el Acuerdo para desarrolladores (Términos y condiciones)
- Pagar tarifa de registro (un solo pago de 25 dólares)
- Completar los datos de la cuenta (nombre del canal, descripción, logotipo)

A continuación se presentan las pantallas y las herramientas que contiene la consola de desarrollador de Google.

### **Tus Aplicaciones**

La siguiente imagen muestra la pantalla inicial de la consola de desarrolladores, la cual proporciona una vista rápida de las aplicaciones al aire o en desarrollo y permite saltar a estadísticas, detalles del producto, o cargar una nueva aplicación.

NOMBRE DE LA APLICACIÓN	PRECIO	INSTALACIONES ACTUALES/TOTALES	VALORACIÓN MEJORORES Y ANRS	ÚLTIMA ACTUALIZACIÓN	ESTADO	
Calculadora de Herencia	—	—	★ —	—	Borrador	
Go Red 1.0.2	Gratuita	1 / 22	★ 5,00 / 5	1	21/11/2014	Publicada
Infosedis 1.0.4	Gratuita	5 / 11	★ 5,00 / 2	1	21/7/2016	Publicada
PuntoNet Móvil 1.0.5	Gratuita	586 / 3.406	★ 3,42 / 73	—	24/4/2015	Publicada
TinPoint 1.0.3.1	Gratuita	375 / 499	★ 4,79 / 19	2	18/3/2016	Publicada

Gráfico 12 - Tus aplicaciones  
Fuente (Google Play Developer Console, 2016)

## Detalles de tu cuenta

La pantalla Detalles de tu cuenta, especifica “información básica del perfil de desarrollador o tu compañía. Esto te identifica ante Google Play y tus clientes. Puedes volver en cualquier momento para editar la información y cambiar tu configuración.” (Android Developers, 2016)

ACCOUNT DETAILS Saved

DEVELOPER PROFILE

Developer name \*   
The developer name will appear to users under the name of your application.

Email address \*

Website

Phone Number \*   
Include plus sign, country code and area code.  
For example, +1-650-253-0000.

Email updates  I'd like to get occasional emails about development and Google Play opportunities.

Gráfico 13 - Datos de perfil  
Fuente (Google Play Developer Console, 2016)

## Ficha de Play Store

Utiliza la Consola para desarrolladores para configurar una página de Directorio de la tienda. Esa página será la página de inicio de tu aplicación en Google Play. Es la página que los usuarios ven en sus teléfonos celulares o en la Web y donde obtienen información sobre tu aplicación y la descargan. Carga recursos personalizados de la marca, capturas de pantalla y videos para destacar las mejores características de tu aplicación. Proporciona una descripción localizada, agrega notas acerca de la última versión y más. Puedes actualizar el directorio de tu tienda en cualquier momento. (Android Developers, 2016)

### Información del producto

#### FICHA DE PLAY STORE

#### INFORMACIÓN DEL PRODUCTO

Para publicar la aplicación, debes rellenar los campos marcados con \*.

Español (Latinoamérica) – es-419

Idiomas (2) ▼

Gestionar traducciones ▼

#### Título\*

Español (Latinoamérica) – es-419

TinPoint

8 de 30 caracteres

#### Descripción breve\*

Español (Latinoamérica) – es-419

Estimula el buen comportamiento, suma puntos por sus acciones y canjea premios

78 de 80 caracteres

#### Descripción completa \*

Español (Latinoamérica) – es-419

Estimula el buen comportamiento de los niños, suma puntos por sus buenas acciones o actividades extraordinarias, asigna diferentes premios como dulces, golosinas, juguetes, paseos, o lo que a los niños más les guste. Canjea premios por los puntos asignados o que los niños ahorren puntos para un premio mayor.

La disciplina es un conjunto de reglas de comportamiento que en los niños se encuentra íntimamente relacionado con su educación, castigos, recompensas y su comportamiento en general.

Los padres y educadores hacen uso de recompensas con el deseo de conseguir que los niños adopten el tipo de conducta que se considera correcto, por lo cual se podría afirmar que los premios son auxiliares para establecer un método de compensación, así como lo son las buenas calificaciones.

1418 de 4000 caracteres

Consulta la [Política de Metadatos](#) para evitar infracciones comunes relacionadas con los metadatos de la aplicación. No olvides consultar el resto de [políticas del programa](#) antes de enviar tus aplicaciones.

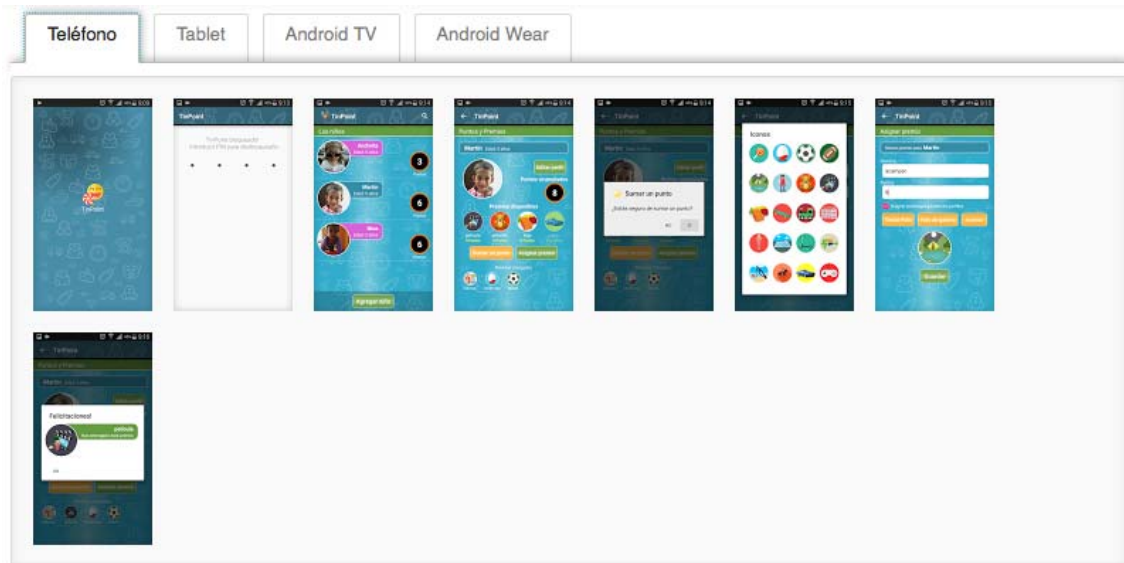
Si tu aplicación o ficha de Play Store [cumple los requisitos para enviar información por adelantado](#) al equipo de revisión de aplicaciones de Google Play, [ponte en contacto con nosotros](#) antes de publicar contenido.

**Gráfico 14** - Información de la aplicación

Fuente (Google Play Developer Console, 2016)

## Elementos gráficos

La ficha de la aplicación permite cargar capturas de pantalla de la aplicación terminada en diferentes formatos (teléfono, tablet, Android TV, Android Wear) así como videos publicitarios o tutoriales. Estas pantallas se visualizan en la tienda de Google para guía de los usuarios.



### Icono de alta resolución \*

Español (Latinoamérica) – es-419 (predeterminado)

512 x 512

Archivo PNG de 32 bits (alfa)



### Imagen destacada \*

Español (Latinoamérica) – es-419 (predeterminado)

1024 x 500

Archivo JPG o PNG de 24 bits (no alfa)



**Gráfico 15** - Elementos gráficos de la ficha de aplicación  
Fuente (Google Play Developer Console, 2016)



## Categorización

La ficha de aplicación cuenta con un sistema de categorización en la cual se llena un cuestionario y se revisa de forma automática el código fuente y los contenidos gráficos de nuestra aplicación.

“Nuestro sistema facilita la comunicación de las calificaciones del contenido con relevancia local y familiar a los usuarios. Además, ayuda a mejorar la participación en la app, ya que están dirigidas al público adecuado para el contenido.” (Ayuda de Developer Console, 2016)

**Tipo de aplicación \***

---

**Categoría \***

---

**Clasificación de contenido \***

[Más información sobre la clasificación de contenido](#)

---

**Nueva clasificación de contenido \***

 **CLASIFICACIÓN APLICADA**  
ID de certificado de la IARC:  
d2d4858e-bf96-4412-a525-e216d98fdc02  
Enviada: 10 de mar. 9:37  
[Ver detalles](#) [Más información](#)



**Gráfico 16** – Categorización  
Fuente (Google Play Developer Console, 2016)



## Datos de contacto

Los datos de contacto son mostrados públicamente con la aplicación donde los usuarios pueden tener un contacto directo con el desarrollador.

<b>Sitio web</b>	<input type="text" value="http://www.mushoq.com"/>
<b>Correo electrónico *</b>	<input type="text" value="serazo@mushoq.com"/> <small>Indica una dirección de correo electrónico en la que los usuarios puedan ponerse en contacto contigo. Esta dirección se mostrará públicamente con tu aplicación.</small>
<b>Teléfono</b>	<input type="text"/>

*Gráfico 17 - Datos de contacto*  
Fuente (Google Play Developer Console, 2016)

## Políticas de privacidad

Es posible proporcionar una URL donde se especifique las políticas de privacidad de la aplicación, lo cual no es obligatorio.

### POLÍTICA DE PRIVACIDAD \*

Si quieres proporcionar una URL de la Política de Privacidad de esta aplicación, indícala a continuación. Consulta también la [Política de Datos de Usuario](#) para evitar infracciones comunes.

<b>Política de Privacidad</b>	<input type="text" value="http://..."/>
	<input checked="" type="checkbox"/> No indicar por ahora ninguna política de privacidad <a href="#">Más información</a>

*Gráfico 18 - Políticas de privacidad*  
Fuente (Google Play Developer Console, 2016)

## Pruebas alfa y beta

Google Play facilita la distribución de versiones de una aplicación a grupos de prueba alfa y beta en cualquier parte del mundo previas al lanzamiento de la versión de producción.

En la sección APK de tu Consola para desarrolladores de Google Play encontrarás las pestañas Pruebas alfa y Pruebas beta. Aquí puedes cargar versiones de los archivos APK de tus aplicaciones y definir una lista de evaluadores como un Grupo de Google o una Comunidad de Google+. Una vez que hagas esto, recibirás una URL que enviarás a tus evaluadores, desde la cual podrán acceder al programa de prueba. (Android Developers, 2016)

The screenshot displays the 'BETA TESTING' configuration interface in the Google Play Developer Console. At the top, there are three tabs: 'PRODUCCIÓN' (Version 5), 'BETA TESTING' (selected), and 'ALPHA TESTING'. Below the tabs, the 'ADMINISTRAR TESTERS' section is visible, with a button for 'Inhabilitar Beta Testing'. The main area is titled 'ELIGE UN MÉTODO DE TESTING' and offers two options: 'Beta Testing cerrado' (Closed Beta Testing) and 'Beta Testing abierto' (Open Beta Testing). The 'Beta Testing abierto' option includes a 'NUEVO' (New) badge and a 'Configurar Beta Testing abierto' button.

PRODUCCIÓN	BETA TESTING	ALPHA TESTING
Versión 5	Configurar el betatesting de tu aplicación	Configurar el testing alpha de tu aplicación

**ADMINISTRAR TESTERS**  
Elige como quieres ejecutar el programa de testing. [Más información](#) Inhabilitar Beta Testing

**ELIGE UN MÉTODO DE TESTING**

- Beta Testing cerrado**  
Añade usuarios individuales por su dirección de correo electrónico. Los usuarios deben estar incluidos en la lista de direcciones de correo electrónico de la prueba para participar en el programa.  
Configurar Beta Testing cerrado
- Beta Testing abierto**  
Ejecuta un programa público al que los usuarios puedan unirse a través de un enlace específico o de la ficha de Play Store de tu aplicación en Google Play [Más información](#) **NUEVO**  
Configurar Beta Testing abierto

**Gráfico 19** - Pruebas alfa y beta  
Fuente (Google Play Developer Console, 2016)

## Configuración de APK de producción

Al finalizar las pruebas alfa y beta se puede pasar el APK a producción en un solo clic, o subir un nuevo APK directamente.

<b>PRODUCCIÓN</b> Versión <b>5</b>	<b>BETA TESTING</b> Configurar el betatesting de tu aplicación	<b>ALPHA TESTING</b> Configurar el testing alpha de tu aplicación
--	---	--

### CONFIGURACIÓN DE PRODUCCIÓN

[Subir nuevo APK de producción](#)

APK ACTUAL Fecha y hora de publicación: **18/3/2016 14:44:46**

<b>Dispositivos compatibles</b> <b>11327</b> <a href="#">Ver lista</a>	<b>Dispositivos excluidos</b> <b>0</b> <a href="#">Administrar dispositivos excluidos</a>
--	---

▼ VERSIÓN	FECHA DE SUBIDA	ESTADO	ACCIONES
5 (1.0.3.1)	18/3/2016	en producción	

### OTROS APK [Ocultar](#)

▼ VERSIÓN	FECHA DE SUBIDA	ESTADO	
4 (1.0.3)	17/3/2016	No publicado	<a href="#">Mostrar detalles</a>
3 (1.0.2)	14/3/2016	No publicado	<a href="#">Mostrar detalles</a>
2 (1.0.1)	11/3/2016	No publicado	<a href="#">Mostrar detalles</a>
1 (1.0)	10/3/2016	No publicado	<a href="#">Mostrar detalles</a>

*Gráfico 20 - Configuración de APK de producción  
Fuente (Google Play Developer Console, 2016)*

## RESULTADOS DE ESTADÍSTICAS DE GOOGLE ANALYTICS

Al utilizar un dispositivo Android los usuarios están obligados a usar una cuenta de Google, en esta cuenta se ingresan datos personales como fecha de nacimiento, sexo, idioma, ubicación, preferencias y otros datos que sin ser ingresados manualmente Google los detecta y almacena de acuerdo a las actividades y el uso de diferentes aplicaciones.

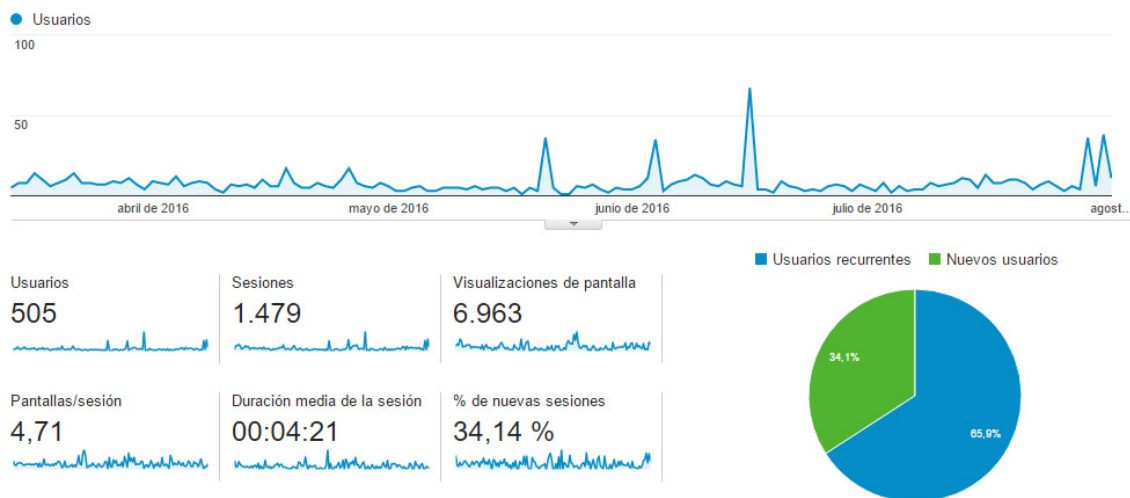
Al utilizar la herramienta Google Analytics se accede a ciertos datos de usuarios que Google brinda para realizar estadísticas en diferentes áreas.

Los siguientes cuadros estadísticos muestran resultados de instalaciones y uso entre el 14 de marzo y el 1 de agosto del 2016.

### Audiencia

La herramienta para desarrolladores Google Analytics presenta diferentes estadísticas e información agrupada del tráfico, audiencia, adquisición, comportamiento y conversiones en un sitio web o en una aplicación móvil.

Las instalaciones a nivel mundial han alcanzado las 505 en total y la duración media de la sesión es de 4:21 minutos.

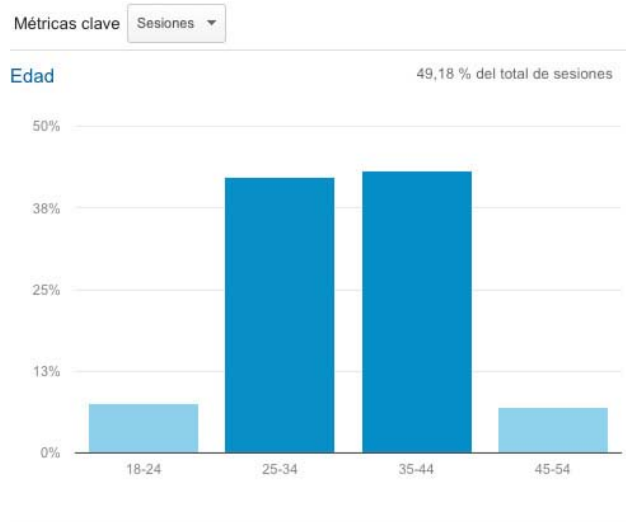


**Gráfico 21** – Usuarios totales  
Fuente (Google Analytics, 2016)

## Datos demográficos

### Edad

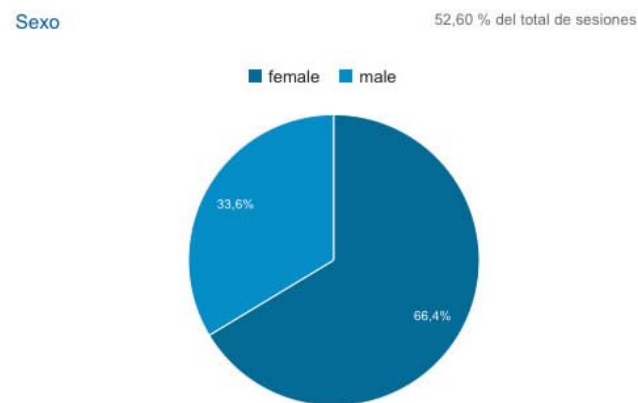
El siguiente gráfico muestra que la aplicación la utilizan más personas que comprenden las edades entre 25 y 44 años, un rango donde normalmente entran padres, tutores o profesionales de la educación.



**Gráfico 22 – Edad**  
Fuente (Google Analytics, 2016)

### Sexo

El siguiente gráfico muestra que el 66.4% de los usuarios de la aplicación son mujeres y un 33.6% son hombres.



**Gráfico 23 - Sexo**  
Fuente (Google Analytics, 2016)

## Idioma

En los dispositivos que por defecto tengan otros idiomas que no sean inglés o español la aplicación se abrirá en idioma inglés.

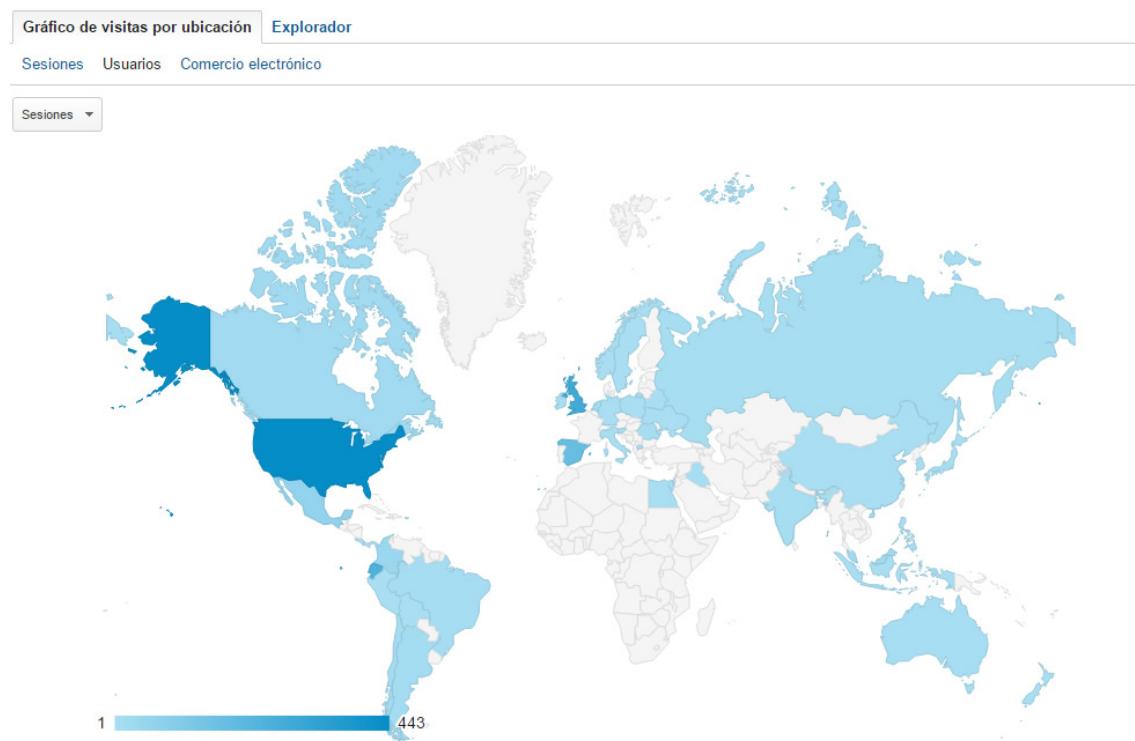
Idioma ?	Sesiones ? ↓	Visualizaciones de pantalla ?	Pantallas/sesión ?	Duración media de la sesión ?
	1.479 <small>% del total: 100,00 % (1.479)</small>	6.963 <small>% del total: 100,00 % (6.963)</small>	4,71 <small>Media de la vista: 4,71 (0,00 %)</small>	00:04:21 <small>Media de la vista: 00:04:21 (0,00 %)</small>
1. en-us	471 (31,85 %)	2.829 (40,63 %)	6,01	00:05:53
2. es-us	236 (15,96 %)	997 (14,32 %)	4,22	00:04:13
3. es-es	204 (13,79 %)	1.140 (16,37 %)	5,59	00:04:36
4. (not set)	194 (13,12 %)	0 (0,00 %)	0,00	00:00:00
5. en-gb	108 (7,30 %)	759 (10,90 %)	7,03	00:06:08
6. en	40 (2,70 %)	0 (0,00 %)	0,00	00:00:00
7. es-ec	39 (2,64 %)	126 (1,81 %)	3,23	00:03:58
8. en-sg	24 (1,62 %)	58 (0,83 %)	2,42	00:01:08
9. en-ca	22 (1,49 %)	86 (1,24 %)	3,91	00:04:19
10. es-co	21 (1,42 %)	88 (1,26 %)	4,19	00:03:25
11. es-mx	21 (1,42 %)	133 (1,91 %)	6,33	00:04:06
12. es-cl	15 (1,01 %)	82 (1,18 %)	5,47	00:05:06
13. nl-nl	14 (0,95 %)	112 (1,61 %)	8,00	00:03:33
14. es-ar	11 (0,74 %)	68 (0,98 %)	6,18	00:05:03
15. pl-pl	10 (0,68 %)	110 (1,58 %)	11,00	00:06:51
16. pt-br	10 (0,68 %)	63 (0,90 %)	6,30	00:11:27
17. iw-il	7 (0,47 %)	52 (0,75 %)	7,43	00:10:12
18. ar-eg	6 (0,41 %)	125 (1,80 %)	20,83	00:09:03

**Gráfico 24 – Idioma**  
Fuente (Google Analytics, 2016)

## País

La aplicación puede ser utilizada en idioma inglés como lenguaje principal y traducción al español para los dispositivos en los que se encuentre este idioma como predeterminado, gracias a esto la aplicación ha sido descargada en los siguientes países: Reino Unido, Estados Unidos, España, Ecuador, México, Colombia, Argentina, Brasil, India, entre otras.

A continuación se muestran dos gráficos que evidencian lo antes mencionado.



**Gráfico 25** - Mapa países  
Fuente (Google Analytics, 2016)

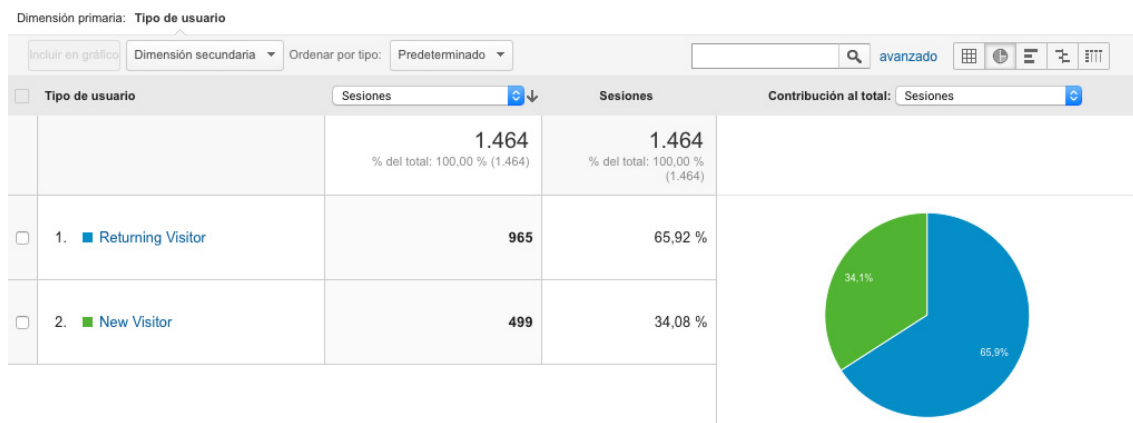
Pais ?	Usuarios ? ↓	Nuevos usuarios ?	Sesiones ?	Visualizaciones de pantalla ?	Pantallas/sesión ?	Duración media de la sesión ?
	<b>505</b> % del total: 100,00 % (505)	<b>505</b> % del total: 100,00 % (505)	<b>1.479</b> % del total: 100,00 % (1.479)	<b>6.963</b> % del total: 100,00 % (6.963)	<b>4,71</b> Media de la vista: 4,71 (0,00 %)	<b>00:04:21</b> Media de la vista: 00:04:21 (0,00 %)
1.  United Kingdom	<b>186</b> (36,40 %)	<b>186</b> (36,83 %)	<b>279</b> (18,86 %)	<b>947</b> (13,60 %)	3,39	00:02:44
2.  United States	<b>101</b> (19,77 %)	<b>99</b> (19,60 %)	<b>443</b> (29,95 %)	<b>2.647</b> (38,02 %)	5,98	00:05:37
3.  Spain	<b>49</b> (9,59 %)	<b>49</b> (9,70 %)	<b>180</b> (12,17 %)	<b>1.014</b> (14,56 %)	5,63	00:04:54
4.  Ecuador	<b>42</b> (8,22 %)	<b>42</b> (8,32 %)	<b>245</b> (16,57 %)	<b>953</b> (13,69 %)	3,89	00:04:03
5. (not set)	<b>20</b> (3,91 %)	<b>18</b> (3,56 %)	<b>26</b> (1,76 %)	<b>51</b> (0,73 %)	1,96	00:00:57
6.  Mexico	<b>15</b> (2,94 %)	<b>14</b> (2,77 %)	<b>63</b> (4,26 %)	<b>312</b> (4,48 %)	4,95	00:04:45
7.  Colombia	<b>9</b> (1,76 %)	<b>9</b> (1,78 %)	<b>40</b> (2,70 %)	<b>154</b> (2,21 %)	3,85	00:03:25
8.  Argentina	<b>6</b> (1,17 %)	<b>6</b> (1,19 %)	<b>17</b> (1,15 %)	<b>125</b> (1,80 %)	7,35	00:04:22
9.  Brazil	<b>6</b> (1,17 %)	<b>6</b> (1,19 %)	<b>11</b> (0,74 %)	<b>63</b> (0,90 %)	5,73	00:10:24
10.  India	<b>6</b> (1,17 %)	<b>6</b> (1,19 %)	<b>8</b> (0,54 %)	<b>14</b> (0,20 %)	1,75	00:03:22
11.  Iraq	<b>6</b> (1,17 %)	<b>6</b> (1,19 %)	<b>6</b> (0,41 %)	<b>0</b> (0,00 %)	0,00	00:00:00
12.  Netherlands	<b>6</b> (1,17 %)	<b>6</b> (1,19 %)	<b>14</b> (0,95 %)	<b>112</b> (1,61 %)	8,00	00:03:33
13.  Australia	<b>5</b> (0,98 %)	<b>5</b> (0,99 %)	<b>6</b> (0,41 %)	<b>27</b> (0,39 %)	4,50	00:01:45
14.  China	<b>4</b> (0,78 %)	<b>4</b> (0,79 %)	<b>4</b> (0,27 %)	<b>0</b> (0,00 %)	0,00	00:00:00
15.  Germany	<b>4</b> (0,78 %)	<b>4</b> (0,79 %)	<b>5</b> (0,34 %)	<b>10</b> (0,14 %)	2,00	00:05:56
16.  Japan	<b>4</b> (0,78 %)	<b>4</b> (0,79 %)	<b>4</b> (0,27 %)	<b>0</b> (0,00 %)	0,00	00:00:00
17.  Hong Kong	<b>3</b> (0,59 %)	<b>3</b> (0,59 %)	<b>3</b> (0,20 %)	<b>15</b> (0,22 %)	5,00	00:03:52
18.  Russia	<b>3</b> (0,59 %)	<b>3</b> (0,59 %)	<b>9</b> (0,61 %)	<b>2</b> (0,03 %)	0,22	00:00:23
19.  Canada	<b>2</b> (0,39 %)	<b>2</b> (0,40 %)	<b>22</b> (1,49 %)	<b>86</b> (1,24 %)	3,91	00:04:19

**Gráfico 26 - Lista países**  
Fuente (Google Analytics, 2016)



## Usuarios recurrentes vs usuarios nuevos

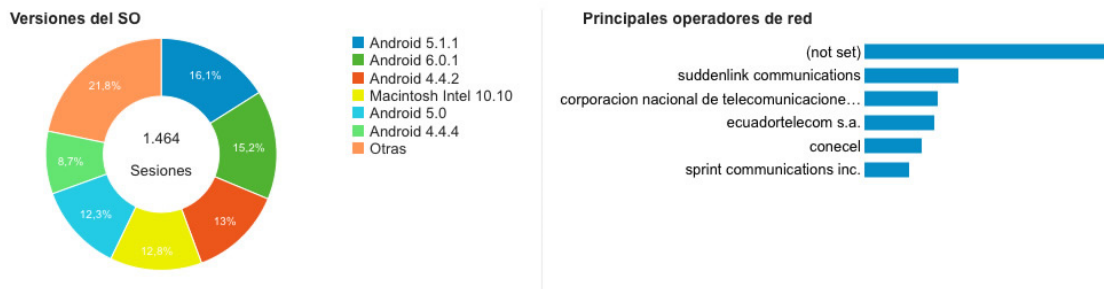
El siguiente gráfico muestra el porcentaje de usuarios recurrentes, es decir, que han usado anteriormente la aplicación versus el porcentaje de usuarios nuevos.



**Gráfico 27** - Usuarios recurrentes vs usuarios nuevos  
Fuente (Google Analytics, 2016)

## Versiones del sistema operativo y operadores de red

En el siguiente gráfico se tiene datos de los diferentes sistemas operativos en los que corre la aplicación y las operadoras celulares principales que usan los dispositivos, cabe indicar que cuando no existe una operadora de red celular “(not set)” se trata de dispositivos que usan la aplicación mientras están conectados a una red local inalámbrica.



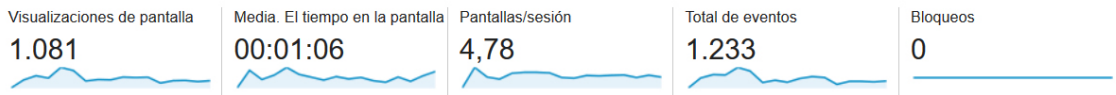
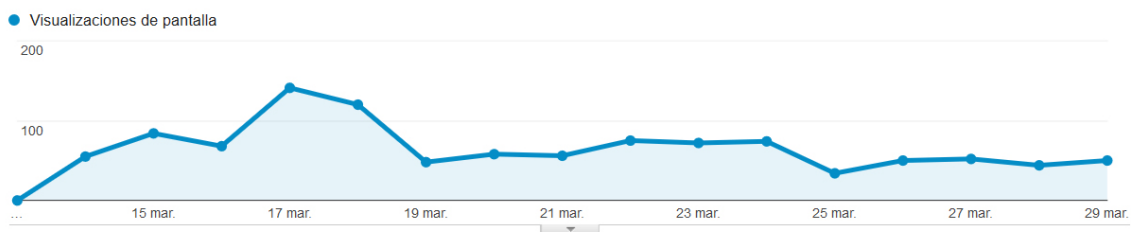
**Gráfico 28** - Versiones del SO y operadores de red  
Fuente (Google Analytics, 2016)

## Visión general del comportamiento

Los informes de comportamiento permiten saber de qué forma los usuarios utilizan la aplicación, el número total de pantallas vistas, el orden en que se ven y la duración media de una sesión.

### Comportamiento en pantallas

El siguiente gráfico nos muestra el número de pantallas vistas y el orden en que estas se ven.

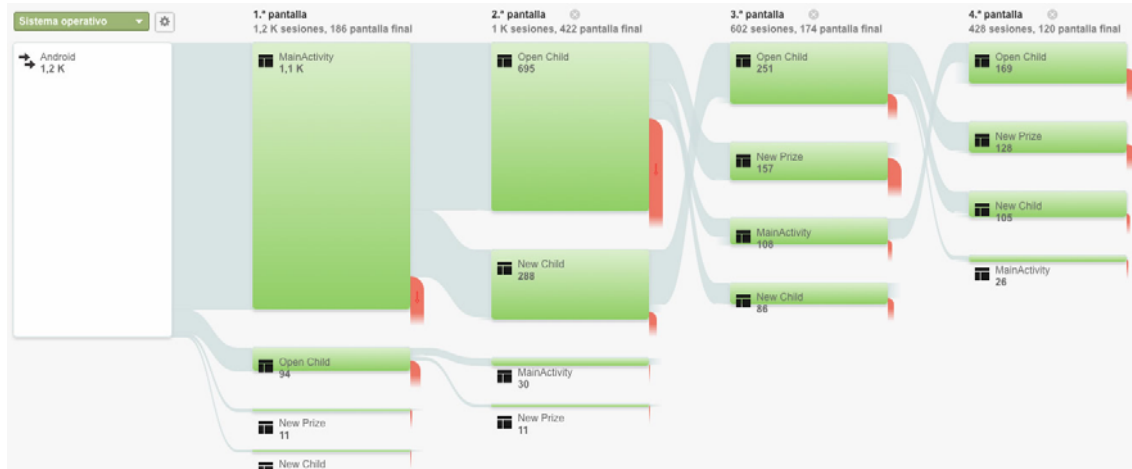


Nombre en pantalla	Visualizaciones de pantalla	Nombre en pantalla	Total de eventos	Categoría de evento	Total de eventos
Open Child	427	Open Child	380	MainActivity	427
MainActivity	277	New Prize	333	OpenChild	345
New Prize	194	New Child	317	NewPrize	219
New Child	183	MainActivity	185	NewChild	217
	<a href="#">ver todo el informe</a>		<a href="#">ver todo el informe</a>	Action	21

**Gráfico 29 – Interacción**  
Fuente (Google Analytics, 2016)

## Flujo del comportamiento

El siguiente gráfico muestra la estadística del flujo de los usuarios al utilizar la aplicación, desde la pantalla inicial hasta las siguientes pantallas y también en que actividad cierran la aplicación.



**Gráfico 30 - Flujo**

Fuente (Google Analytics, 2016)



## CONCLUSIONES

El análisis de requerimientos en el desarrollo de software permite obtener un producto de calidad, optimizando tiempo y recursos, suministrando los medios necesarios para el cumplimiento de resultados, procedimientos y datos.

La integración entre la interfaz de usuario y el diseño gráfico contribuye a construir un producto vistoso y sencillo de utilizar, lo que ayuda al usuario a simplificar el manejo de un determinado software.

El uso de lenguaje nativo de cada plataforma (Android, iOS, Windows Phone, BlackBerry, etc.) es la mejor elección al momento del desarrollo de una aplicación. Cada plataforma utiliza un lenguaje de programación diferente, librerías y herramientas particulares. Aunque al momento de desarrollar una aplicación multiplataforma se debe conocer muchos lenguajes y herramientas, se obtiene una máxima flexibilidad, adaptación y rendimiento para la aplicación a desarrollar.

La consola de desarrolladores de Google Play ofrece las herramientas necesarias para publicar aplicaciones de forma sencilla, otorgando facilidad para realizar pruebas y manteniendo un control óptimo sobre las versiones de las aplicaciones.

Google Analytics es una herramienta de uso sencillo. Ofrece información valiosa sobre audiencia, adquisición, comportamiento y conversiones de una página web o de una aplicación móvil; información que puede ser utilizada para el mejoramiento o actualización de un software.

Las Tecnologías de la Información y la comunicación (TICs), han provocado, y van a seguir provocando un cambio radical no solo en los procedimientos de todos los sectores de la economía y las sociedades humanas, cada vez más globalizadas, sino también en el pensamiento y la expresión en los procesos mentales, la apreciación y la sensibilidad de sus usuarios, cuyo número no deja de crecer. La población más joven, siempre más abierta y flexible que los adultos a los cambios, ha acogido con entusiasmo las nuevas tecnologías.

Manejando con prudencia y sentido común los avances tecnológicos como soporte de aprendizajes y cambios conductuales en los niños, los adultos futuros pueden llegar a ser personas cada vez más equilibradas, positivas y con una alta autoestima y por lo mismo ser, además, productivas y solidarias.

## RECOMENDACIONES

Desarrollar la aplicación en otras plataformas (iOs, Whindows Phone, etc) para que su acceso sea más amplio.

Continuar con el desarrollo de la aplicación para replicar los datos locales en la nube y no perder los perfiles al cambiar un dispositivo.

Implementar en el flujo de la aplicación objetivos por equipo, para obtener más puntos y premios especiales.

Analizar los resultados reflejados en la herramienta Analytics for Android con el fin de mejorar la experiencia de usuario.

## BIBLIOGRAFÍA

- Android Developer Guide.* (2016). Obtenido de [http://developer.android.com/intl/es/guide/practices/screens\\_support.html](http://developer.android.com/intl/es/guide/practices/screens_support.html)
- Android Developers. (2016). *Consola para desarrolladores.* Obtenido de <https://developer.android.com/distribute/googleplay/developer-console.html?hl=es>.
- Ayuda de Developer Console. (2016). *Calificaciones del contenido para aplicaciones y juegos.* Obtenido de <https://support.google.com/googleplay/android-developer/answer/188189?hl=es-419>.
- Canós, J. H., Letelier, P., & Penadés, M. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia.*
- Craig, G., & Woolfolk, A. (1990). *Manual de Psicología y desarrollo educativo.*
- Curso Java para Android y Programación Orientada a Objetos.* (2010). Obtenido de <http://www.digitalllearning.es/curso-java-para-android.html>
- Dashboards – Android Developers.* (Febrero de 2016). Obtenido de <http://developer.android.com/intl/es/about/dashboards/index.html>
- Dodenhof, H. (2016). *CircleImageView.* Obtenido de <https://github.com/hdodenhof/CircleImageView>.
- Española, R. A. (2014). *Diccionario de la lengua española.*
- Figueroa, R. G., Solís, C. J., & Cabrera, A. A. (2008). *Metodologías Tradicionales vs. Metodologías Ágiles.* Obtenido de Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>
- Future, I. A. (s.f.). *IDC.* Obtenido de <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>



- Gironés, J. T. (2012). *El gran libro de Android*. Marcombo.
- Good, T. L., & Brophy, J. (1995). *Psicología educativa contemporánea*.
- Google Analytics*. (2016). Obtenido de <https://analytics.google.com/analytics/>
- Google Inc. (2016). *Consola para desarrolladores | Android Developers*. Obtenido de Consola para desarrolladores: <https://developer.android.com/distribute/googleplay/developer-console.html?hl=es>
- Google Play Developer Console*. (2016). Obtenido de <https://play.google.com/apps/publish/>
- Graig, G., Mehrens, & Clarizio, W. (1988). *Psicología Educativa Contemporánea, Tomo 3*.
- Groussard, T. (2012). *JAVA 7: Los fundamentos del lenguaje Java*. Ediciones Eni.
- ISO/IEC. (2010). *9241-210*.
- Muschko, B. (2014). *Gradle in action*. Manning.
- The Statistics Portal*. (2016). Obtenido de <http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- Yalantis. (2016). *UCrop*. Obtenido de <https://github.com/Yalantis/uCrop>.

# **ANEXOS**

## **Anexo 1**

Código fuente

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\DbHelper.java

```
1 package com.mushoq.tinpoint;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7 /**
8  * Created by santiago on 2/11/16.
9  */
10 public class DbHelper extends SQLiteOpenHelper {
11
12     private static final String DB_NAME = "tinpoint.sqlite";
13     private static final int DB_SCHEME_VERSION = 1;
14
15     public DbHelper(Context context) {
16         super(context, DB_NAME, null, DB_SCHEME_VERSION);
17     }
18
19     @Override
20     public void onCreate(SQLiteDatabase db) {
21         db.execSQL(DbManager.CREATE_CHILDREN);
22         db.execSQL(DbManager.CREATE_PRIZES);
23     }
24
25     @Override
26     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
27
28     }
29 }
30
```

```

1  package com.mushoq.tinpoint;
2
3  import android.content.ContentValues;
4  import android.content.Context;
5  import android.database.Cursor;
6  import android.database.sqlite.SQLiteDatabase;
7  import android.os.Vibrator;
8
9  /**
10 * Created by santiagoe on 2/11/16.
11 */
12 public class DbManager {
13     /*TABLE CHILDREN*/
14     public static final String TABLE_CHILDREN = "children";
15     public static final String CHILDREN_ID = "_id";
16     public static final String CHILDREN_NAME = "name";
17     public static final String CHILDREN_BIRTHDATE = "birthdate";
18     public static final String CHILDREN_PHOTO = "photo";
19     public static final String CHILDREN_GENDER = "gender";
20     public static final String CHILDREN_POINTS = "points";
21
22     public static final String CREATE_CHILDREN = "create table " + TABLE_CHILDREN + " ("
23         + CHILDREN_ID + " integer primary key autoincrement, "
24         + CHILDREN_NAME + " text not null, "
25         + CHILDREN_BIRTHDATE + " text not null, "
26         + CHILDREN_PHOTO + " text, "
27         + CHILDREN_POINTS + " integer, "
28         + CHILDREN_GENDER + " text not null);";
29
30     /*TABLE PRIZES*/
31     public static final String TABLE_PRIZES = "prizes";
32     public static final String PRIZES_ID = "_id";
33     public static final String PRIZES_CHILD_ID = "prize_child_id";
34     public static final String PRIZES_NAME = "prize_name";
35     public static final String PRIZES_PHOTO = "prize_photo";
36     public static final String PRIZES_POINTS = "prize_points";
37     public static final String PRIZES_GIVEN = "prize_given";
38
39     public static final String CREATE_PRIZES = "create table " + TABLE_PRIZES + " ("
40         + PRIZES_ID + " integer primary key autoincrement, "
41         + PRIZES_CHILD_ID + " integer not null, "
42         + PRIZES_NAME + " text not null, "
43         + PRIZES_PHOTO + " text, "
44         + PRIZES_GIVEN + " text, "
45         + PRIZES_POINTS + " integer);";
46
47     private DbHelper helper;
48
49
50     public static final String[] COLUMNS_CHILDS = new String[]{CHILDREN_ID, CHILDREN_NAME,
51         CHILDREN_BIRTHDATE, CHILDREN_PHOTO, CHILDREN_POINTS, CHILDREN_GENDER};
52     public static final String[] COLUMNS_PRIZES = new String[]{PRIZES_ID, PRIZES_CHILD_ID,
53         PRIZES_NAME, PRIZES_PHOTO, PRIZES_GIVEN, PRIZES_POINTS};
54
55     public DbManager(Context context) {
56         helper = new DbHelper(context);
57     }
58
59     public ContentValues genChildValues(String name, String birthdate, String photo, Integer points,
60         String gender){
61         ContentValues values = new ContentValues();
62         values.put(CHILDREN_NAME, name);
63         values.put(CHILDREN_BIRTHDATE, birthdate);
64         values.put(CHILDREN_PHOTO, photo);
65         values.put(CHILDREN_POINTS, points);
66         values.put(CHILDREN_GENDER, gender);
67
68         return values;

```

```
66     }
67
68     public ContentValues genPrizeValues(Integer prizechildid, String prizename, String prizephoto,
String prizegiven, Integer prizepoints){
69         ContentValues values = new ContentValues();
70         values.put(PRIZES_CHILD_ID, prizechildid);
71         values.put(PRIZES_NAME, prizename);
72         values.put(PRIZES_PHOTO, prizephoto);
73         values.put(PRIZES_GIVEN, prizegiven);
74         values.put(PRIZES_POINTS, prizepoints);
75         return values;
76     }
77
78     /*public void removeChild(String id){
79         db.delete(TABLE_CHILDREN, CHILDREN_ID + "=?", new String[]{id});
80
81     }
82
83     public void truncateChild(){
84         db.execSQL("delete from " + TABLE_CHILDREN);
85     }/i
86 }
87
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\SecurityPin.java

```
1 package com.mushoq.tinpoint;
2
3 import android.Manifest;
4
5 import android.content.Intent;
6 import android.content.pm.PackageManager;
7 import android.support.v4.app.ActivityCompat;
8
9 import android.support.v7.app.AppCompatActivity;
10 import android.os.Bundle;
11 import android.widget.Toast;
12
13 import com.guardanis.applock.CreateLockActivity;
14 import com.guardanis.applock.UnlockActivity;
15 import com.guardanis.applock.locking.LockingHelper;
16
17 import java.util.ArrayList;
18 import java.util.HashMap;
19 import java.util.List;
20 import java.util.Map;
21
22
23 public class SecurityPin extends AppCompatActivity {
24
25     final private int REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS = 124;
26
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_security_pin);
32
33         int sdk = android.os.Build.VERSION.SDK_INT;
34         if(sdk < android.os.Build.VERSION_CODES.M) {
35             securityPin();
36         }else{
37             myPermissions();
38         }
39
40
41     }
42
43     @Override
44     public void onActivityResult(int requestCode, int resultCode, Intent data){
45         if(requestCode == LockingHelper.REQUEST_CODE_UNLOCK){
46             Intent mainIntent = new Intent().setClass(SecurityPin.this, MainActivity.class);
47             startActivity(mainIntent);
48             finish();
49         }
50         if(requestCode == LockingHelper.REQUEST_CODE_CREATE_LOCK){
51             Intent mainIntent = new Intent().setClass(SecurityPin.this, MainActivity.class);
52             startActivity(mainIntent);
53             finish();
54         }
55     }
56 }
57
58 @Override
59 public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
60     switch (requestCode) {
61         case REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS:
62             {
63                 Map<String, Integer> perms = new HashMap<String, Integer>();
64                 // Initial
65
66                 perms.put(Manifest.permission.READ_EXTERNAL_STORAGE, PackageManager.
67 PERMISSION_GRANTED);
68                 perms.put(Manifest.permission.WRITE_EXTERNAL_STORAGE, PackageManager.
```

```

67 PERMISSION_GRANTED);
68     // Fill with results
69     for (int i = 0; i < permissions.length; i++)
70         perms.put(permissions[i], grantResults[i]);
71     // Check for CAMERA
72     if (perms.get(Manifest.permission.READ_EXTERNAL_STORAGE) == PackageManager.
PERMISSION_GRANTED
73         && perms.get(Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
74         // All Permissions Granted
75         securityPin();
76     } else {
77         // Permission Denied
78         Toast.makeText(SecurityPin.this, "Some Permission is Denied", Toast.LENGTH_SHORT
)
79         .show();
80         myPermissions();
81     }
82 }
83 break;
84 default:
85     super.onRequestPermissionsResult(requestCode, permissions, grantResults);
86 }
87 }
88
89 public void securityPin(){
90     if(LockingHelper.hasSavedPIN(this)){
91         Intent intent = new Intent(SecurityPin.this, UnlockActivity.class);
92         startActivityForResult(intent, LockingHelper.REQUEST_CODE_UNLOCK);
93     }else{
94         Intent intent = new Intent(SecurityPin.this, CreateLockActivity.class);
95         startActivityForResult(intent, LockingHelper.REQUEST_CODE_CREATE_LOCK);
96     }
97 }
98
99 public void myPermissions(){ List<String> permissionsNeeded = new ArrayList<String>();
100
101     final List<String> permissionsList = new ArrayList<String>();
102
103     if (!addPermission(permissionsList, Manifest.permission.READ_EXTERNAL_STORAGE))
104         permissionsNeeded.add("READ_EXTERNAL_STORAGE");
105     if (!addPermission(permissionsList, Manifest.permission.WRITE_EXTERNAL_STORAGE))
106         permissionsNeeded.add("WRITE_EXTERNAL_STORAGE");
107
108     if (permissionsList.size() > 0) {
109         if (permissionsNeeded.size() > 0) {
110             // Need Rationale
111             ActivityCompat.requestPermissions(SecurityPin.this, permissionsList.toArray(new String[
permissionsList.size()]),
112                 REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS);
113             return;
114         }
115         ActivityCompat.requestPermissions(SecurityPin.this, permissionsList.toArray(new String[
permissionsList.size()]),
116             REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS);
117         return;
118     }
119     securityPin();
120 }
121 }
122
123 private boolean addPermission(List<String> permissionsList, String permission) {
124     if (ActivityCompat.checkSelfPermission(SecurityPin.this, permission) != PackageManager.
PERMISSION_GRANTED){
125         permissionsList.add(permission);
126         // Check for Rationale Option
127         if (!ActivityCompat.shouldShowRequestPermissionRationale(SecurityPin.this, permission))
128             return false;

```



File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\SecurityPin.java

```
129     }  
130     return true;  
131 }  
132  
133  
134 }  
135
```

```

1  package com.mushoq.tinpoint;
2
3
4
5  import android.app.Activity;
6  import android.app.AlertDialog;
7  import android.app.LoaderManager;
8  import android.app.SearchManager;
9  import android.content.Context;
10 import android.content.CursorLoader;
11 import android.content.DialogInterface;
12 import android.content.Intent;
13 import android.database.Cursor;
14
15 import android.support.v7.app.AppCompatActivity;
16 import android.os.Bundle;
17 import android.support.v7.widget.SearchView;
18 import android.view.ContextMenu;
19 import android.view.Menu;
20 import android.view.MenuInflater;
21 import android.view.MenuItem;
22 import android.view.View;
23 import android.widget.AdapterView;
24 import android.widget.Button;
25 import android.widget.FrameLayout;
26 import android.widget.ListView;
27 import com.google.android.gms.ads.AdRequest;
28 import com.google.android.gms.ads.AdSize;
29 import com.google.android.gms.ads.AdView;
30 import com.google.android.gms.analytics.HitBuilders;
31 import com.google.android.gms.analytics.Tracker;
32
33 public class MainActivity extends AppCompatActivity implements LoaderManager.LoaderCallbacks<
Cursor>{
34
35     private DbManager manager;
36     private ListView childList;
37     private ChildCursorAdapter adapter;
38     private Button newchild;
39     private LoaderManager.LoaderCallbacks<Cursor> mCallbacks;
40     private FrameLayout mAdFrameLayout;
41     private static final int LOADER_ID = 1, SET_CHILD = 151;
42     private Tracker mTracker;
43     private AnalyticsApplication application;
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_main);
48         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
49         getSupportActionBar().setIcon(R.drawable.header_icon);
50         getSupportActionBar().setTitle("TinPoint");
51         //getSupportActionBar().setSubtitle("Lista de niños");
52
53         manager = new DbManager(this);
54
55         application = (AnalyticsApplication) getApplication();
56         mTracker = application.getDefaultTracker();
57
58         mTracker.setScreenName("MainActivity");
59         mTracker.send(new HitBuilders.ScreenViewBuilder().build());
60
61         mTracker.enableAdvertisingIdCollection(true);
62         //ADS BY GOOGLE
63         mAdFrameLayout = (FrameLayout)findViewById(R.id.addsFrameLayout);
64
65         AdView mAdView = new AdView(this);
66         mAdView.setAdUnitId(getString(R.string.banner_ad_unit_id));
67         mAdView.setAdSize(AdSize.SMART_BANNER);

```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\MainActivity.java

```
68     mAdFrameLayout.addView(mAdView);
69     mAdView.loadAd(new AdRequest.Builder().build());
70
71
72     mCallbacks = this;
73     LoaderManager lm = getLoaderManager();
74     lm.initLoader(LOADER_ID, null, mCallbacks);
75
76     childList = (ListView)findViewById(R.id.childList);
77
78
79     adapter = new ChildCursorAdapter(this, null, 0);
80     childList.setAdapter(adapter);
81
82     childList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
83         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
84             mTracker.send(new HitBuilders.EventBuilder()
85                 .setCategory("MainActivity")
86                 .setAction("Open Child")
87                 .build());
88             Intent intent = new Intent(MainActivity.this, OpenChildActivity.class);
89             intent.putExtra("id", String.valueOf(id));
90             startActivity(intent);
91             overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
92         }
93     });
94
95     registerForContextMenu(childList);
96
97     newchild = (Button) findViewById(R.id.newchild);
98     newchild.setOnClickListener(new View.OnClickListener() {
99         @Override
100        public void onClick(View v) {
101            Intent intent = new Intent(MainActivity.this, NewChildActivity.class);
102            startActivityForResult(intent, SET_CHILD);
103            overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
104        }
105    });
106 }
107
108 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
109
110     if (requestCode == SET_CHILD) {
111         if (resultCode == Activity.RESULT_OK) {
112             adapter = new ChildCursorAdapter(this, null, 0);
113             adapter.changeCursor(loadCursorChild());
114
115             childList.setAdapter(adapter);
116         }
117         if (resultCode == Activity.RESULT_CANCELED) {
118             //Write your code if there's no result
119         }
120     }
121 }
122
123 @Override
124 public boolean onCreateOptionsMenu(Menu menu) {
125     MenuInflater inflater = getMenuInflater();
126     inflater.inflate(R.menu.main_menu, menu);
127
128     SearchManager searchManager = (SearchManager) getSystemService(Context.
SEARCH_SERVICE);
129     SearchView searchView = (SearchView) menu.findItem(R.id.action_search).getActionView();
130     // Assumes current activity is the searchable activity
131     searchView.setSearchableInfo(searchManager.getSearchableInfo(getComponentName()));
132     searchView.setIconifiedByDefault(true);
133     searchView.setQueryHint(getText(R.string.search));
134 }
```

File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\MainActivity.java

```
135     searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
136         @Override
137         public boolean onQueryTextSubmit(String query) {
138             Cursor searchChild = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMN_COLUMNS_CHILDS, manager.CHILDREN_NAME + " like ? ", new String[]{"%" + query + "%"}, null);
139             adapter.changeCursor(searchChild);
140             return false;
141         }
142
143         @Override
144         public boolean onQueryTextChange(String newText) {
145
146             Cursor searchChild = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMN_COLUMNS_CHILDS, manager.CHILDREN_NAME + " like ? ", new String[]{"%" + newText + "%"},
null);
147             adapter.changeCursor(searchChild);
148             return false;
149         }
150     });
151
152
153     return super.onCreateOptionsMenu(menu);
154 }
155
156 @Override
157 public void onCreateContextMenu(ContextMenu menu, View v,
158     ContextMenu.ContextMenuInfo menuInfo) {
159     if (v.getId()==R.id.childList) {
160         String[] menuItems = {getResources().getString(R.string.edit), getResources().getString(R.string.
remove)};
161         for (int i = 0; i<menuItems.length; i++) {
162             menu.add(Menu.NONE, i, i, menuItems[i]);
163         }
164     }
165 }
166
167 @Override
168 public boolean onContextItemSelected(Menu.Item item) {
169     AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)item.
getMenuInfo();
170     int itemPosition = info.position;
171     Cursor cursor = (adapter).getCursor();
172     cursor.moveToPosition(itemPosition);
173     final String mid = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_ID));
174     final String mName = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_NAME));
175     int menuItemIndex = item.getItemId();
176
177     if(menuItemIndex == 0){
178         Intent intent = new Intent(MainActivity.this, NewChildActivity.class);
179         intent.putExtra("id", mid);
180         startActivity(intent);
181         overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
182     }else if(menuItemIndex == 1){
183         new AlertDialog.Builder(this)
184             .setIcon(android.R.drawable.ic_menu_delete)
185             .setTitle(getResources().getString(R.string.remove) + " " + mName)
186             .setMessage(getResources().getString(R.string.arwYouSureDeleteProfile))
187             .setPositiveButton(getResources().getString(R.string.yes), new DialogInterface.
OnClickListener() {
188                 @Override
189                 public void onClick(DialogInterface dialog, int which) {
190                     mTracker.send(new HitBuilders.EventBuilder()
191                         .setCategory("MainActivity")
192                         .setAction("Remove Child")
193                         .build());
194                     getContentResolver().delete(ChildProvider.CHILDS_URI, manager.CHILDREN_ID +
"=?", new String[]{mid});
195                     adapter.changeCursor(loadCursorChild());
```

File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\MainActivity.java

```
196         }
197
198     })
199     .setNegativeButton(getResources().getString(R.string.no), null)
200     .show();
201 }
202 return true;
203 }
204
205 private Cursor loadCursorChild(){
206     Cursor cursor = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMNS_CHILDS, null, null, manager.CHILDREN_NAME);
207     return cursor;
208 }
209
210 @Override
211 public android.content.Loader<Cursor> onCreateLoader(int id, Bundle args) {
212     if(id == LOADER_ID) {
213         CursorLoader cursorloader = new CursorLoader(MainActivity.this, ChildProvider.CHILDS_URI,
manager.COLUMNS_CHILDS, null, null, manager.CHILDREN_NAME);
214         return cursorloader;
215     }else{
216         return null;
217     }
218 }
219
220 @Override
221 public void onLoadFinished(android.content.Loader<Cursor> loader, Cursor data) {
222     switch (loader.getId()) {
223         case LOADER_ID:
224             // The asynchronous load is complete and the data
225             // is now available for use. Only now can we associate
226             // the queried Cursor with the SimpleCursorAdapter.
227             adapter.changeCursor(data);
228             break;
229     }
230 }
231
232 @Override
233 public void onLoaderReset(android.content.Loader<Cursor> loader) {
234     adapter.swapCursor(null);
235 }
236 }
```

File - D:\work\Tinpoint\TinpointAndroid\app\src\main\java\com\mushoq\tinpoint\ChildProvider.java

```
1 package com.mushoq.tinpoint;
2
3 import android.content.ContentProvider;
4 import android.content.ContentValues;
5 import android.content.UriMatcher;
6 import android.database.Cursor;
7 import android.database.sqlite.SQLiteDatabase;
8 import android.database.sqlite.SQLiteQueryBuilder;
9 import android.net.Uri;
10 import android.support.annotation.Nullable;
11
12 /**
13  * Created by santiagoe on 2/18/16.
14  */
15 public class ChildProvider extends ContentProvider
16 {
17
18     private static final String AUTH = "com.mushoq.tinpoint.ChildProvider";
19     public static final Uri CHILDS_URI = Uri.parse("content://" + AUTH + "/" + DbManager.
TABLE_CHILDREN);
20     public static final Uri PRIZES_URI = Uri.parse("content://" + AUTH + "/" + DbManager.
TABLE_PRIZES);
21
22     final static int CHILDS = 1;
23     final static int PRIZES = 2;
24
25     private DbHelper helper;
26     private DbManager manager;
27     private SQLiteDatabase db;
28
29     private final static UriMatcher uriMatcher;
30     static{
31         uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
32         uriMatcher.addURI(AUTH, DbManager.TABLE_CHILDREN, CHILDS);
33         uriMatcher.addURI(AUTH, DbManager.TABLE_PRIZES, PRIZES);
34     }
35     @Override
36     public boolean onCreate() {
37         helper = new DbHelper(getContext());
38
39         return true;
40     }
41
42     @Nullable
43     @Override
44     public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder
) {
45         Cursor cursor;
46         db = helper.getReadableDatabase();
47         if(uriMatcher.match(uri) == CHILDS){
48             cursor = db.query(manager.TABLE_CHILDREN, projection, selection, selectionArgs, null, null,
sortOrder);
49             cursor.setNotificationUri(getContext().getContentResolver(), uri);
50             return cursor;
51         }else if(uriMatcher.match(uri) == PRIZES){
52             cursor = db.query(manager.TABLE_PRIZES, projection, selection, selectionArgs, null, null,
sortOrder);
53
54             cursor.setNotificationUri(getContext().getContentResolver(), uri);
55             return cursor;
56         }else{
57             return null;
58         }
59     }
60
61     @Nullable
62     @Override
63     public String getType(Uri uri) {
```

```
64     return null;
65 }
66
67 @Nullable
68 @Override
69 public Uri insert(Uri uri, ContentValues values) {
70     db = helper.getWritableDatabase();
71     switch (uriMatcher.match(uri)){
72         case CHILDS:
73             db.insert(manager.TABLE_CHILDREN, null, values);
74             break;
75         case PRIZES:
76             db.insert(manager.TABLE_PRIZES, null, values);
77             break;
78     }
79     db.close();
80     getContext().getContentResolver().notifyChange(uri, null);
81     return null;
82 }
83
84 @Override
85 public int delete(Uri uri, String selection, String[] selectionArgs) {
86     db = helper.getWritableDatabase();
87     switch (uriMatcher.match(uri)){
88         case CHILDS:
89             db.delete(manager.TABLE_CHILDREN, selection, selectionArgs);
90             break;
91         case PRIZES:
92             db.delete(manager.TABLE_PRIZES, selection, selectionArgs);
93             break;
94     }
95     db.close();
96     return 0;
97 }
98
99 @Override
100 public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
101     db = helper.getWritableDatabase();
102     switch (uriMatcher.match(uri)){
103         case CHILDS:
104             db.update(manager.TABLE_CHILDREN, values, selection, selectionArgs);
105             break;
106         case PRIZES:
107             db.update(manager.TABLE_PRIZES, values, selection, selectionArgs);
108             break;
109     }
110     db.close();
111     getContext().getContentResolver().notifyChange(uri, null);
112     return 0;
113 }
114 }
115
```

File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\SplashActivity.java

```
1 package com.mushoq.inpoint;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.animation.Animation;
7 import android.view.animation.AnimationSet;
8 import android.view.animation.AnimationUtils;
9 import android.view.animation.LinearInterpolator;
10 import android.view.animation.RotateAnimation;
11 import android.view.animation.ScaleAnimation;
12 import android.view.animation.TranslateAnimation;
13 import android.widget.ImageView;
14 import android.widget.TextView;
15
16 import java.util.Timer;
17 import java.util.TimerTask;
18
19 public class SplashActivity extends AppCompatActivity {
20
21     private ImageView logo;
22     private long splashDelay = 1500; //1.5 segundos
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         getSupportActionBar().hide();
27         setContentView(R.layout.activity_splash);
28
29         Animation scale = new ScaleAnimation(0, 1, 0, 1, Animation.RELATIVE_TO_SELF, 0.5f, Animation
.RELATIVE_TO_SELF, 0.5f);
30         // 1 second duration
31         scale.setDuration(700);
32         // Moving up
33         Animation slideUp = new TranslateAnimation(0, 1, 0, 1);
34         // 1 second duration
35         slideUp.setDuration(700);
36         // Animation set to join both scaling and moving
37         AnimationSet animSet = new AnimationSet(true);
38         animSet.setFillEnabled(true);
39         animSet.addAnimation(scale);
40         animSet.addAnimation(slideUp);
41
42         logo = (ImageView) findViewById(R.id.logo);
43
44         logo.startAnimation(animSet);
45
46         TimerTask task = new TimerTask() {
47             @Override
48
49             public void run() {
50
51                 Intent mainIntent = new Intent().setClass(SplashActivity.this, SecurityPin.class);
52                 startActivity(mainIntent);
53                 finish();//Destruimos esta actividad para prevenir que el usuario retorne aqui presionando el boton
Atras.
54             }
55         };
56
57         Timer timer = new Timer();
58         timer.schedule(task, splashDelay);//Pasado los 6 segundos dispara la tarea
59     }
60 }
61
```



```
1 package com.mushoq.tinpoint;
2
3 import android.app.Activity;
4 import android.app.Dialog;
5 import android.content.DialogInterface;
6 import android.content.Intent;
7 import android.content.res.Resources;
8 import android.database.Cursor;
9 import android.database.MatrixCursor;
10 import android.graphics.Bitmap;
11 import android.net.Uri;
12 import android.os.Environment;
13 import android.support.annotation.NonNull;
14 import android.support.v7.app.AlertDialog;
15 import android.support.v7.app.AppCompatActivity;
16 import android.os.Bundle;
17 import android.util.Log;
18 import android.view.ContextMenu;
19 import android.view.LayoutInflater;
20 import android.view.Menu;
21 import android.view.MenuItem;
22 import android.view.MotionEvent;
23 import android.view.View;
24 import android.view.ViewGroup;
25 import android.view.Window;
26 import android.view.inputmethod.InputMethodManager;
27 import android.widget.AdapterView;
28 import android.widget.Button;
29 import android.widget.CheckBox;
30 import android.widget.EditText;
31 import android.widget.FrameLayout;
32 import android.widget.LinearLayout;
33 import android.widget.ListView;
34 import android.widget.TextView;
35 import android.widget.Toast;
36
37 import com.google.android.gms.ads.AdRequest;
38 import com.google.android.gms.ads.AdSize;
39 import com.google.android.gms.ads.AdView;
40 import com.google.android.gms.analytics.HitBuilders;
41 import com.google.android.gms.analytics.Tracker;
42 import com.yalantis.ucrop.UCrop;
43
44 import java.io.File;
45 import java.text.SimpleDateFormat;
46 import java.util.Arrays;
47 import java.util.Date;
48 import java.util.List;
49 import java.util.Locale;
50
51 import de.hdodenhof.circleimageview.CircleImageView;
52
53 public class NewAwardActivity extends AppCompatActivity {
54
55     private String id, prizeid = "", photoName, oprizeid, oprizechildid, oprizename, oprizegiven,
oprizephoto, oprizepoints;
56     private CheckBox prizeForAll;
57     private DbManager manager;
58     private Cursor openChild, openPrize;
59     private TextView name, prizefor;
60     private LinearLayout layoutGender;
61     private GenericPrizeCursorAdapter adapter;
62     private ExpandableHeightGridView listIcons;
63     private CircleImageView photo;
64     final int PIC_CROP = 5;
65     private EditText prizename, prizepoints;
66     private Button takephoto, galleryphoto, save, icons;
67     private static final int CAMERA_REQUEST = 1888, RESULT_LOAD_IMG = 2;
```

```

68 private Uri resultUri;
69 private int flag = 0;
70 private FrameLayout mAdFrameLayout;
71 private Tracker mTracker;
72 private AnalyticsApplication application;
73
74 @Override
75 protected void onCreate(Bundle savedInstanceState) {
76     super.onCreate(savedInstanceState);
77     setContentView(R.layout.activity_new_award);
78     Bundle extras = getIntent().getExtras();
79     id = extras.getString("id");
80     if(getIntent().hasExtra("prizeid")){
81         prizeid = extras.getString("prizeid");
82     }
83
84
85
86     getSupportActionBar().setTitle("TinPoint");
87     getSupportActionBar().setDisplayHomeAsUpEnabled(true);
88
89     setupUI(findViewById(R.id.parent));
90
91     manager = new DbManager(this);
92
93     application = (AnalyticsApplication) getApplication();
94     mTracker = application.getDefaultTracker();
95
96     mTracker.setScreenName("New Prize");
97     mTracker.send(new HitBuilders.ScreenViewBuilder().build());
98
99     /**ADS BY GOOGLE
100     mAdFrameLayout = (FrameLayout)findViewById(R.id.addsFrameLayout);
101
102     AdView mAdView = new AdView(this;
103     mAdView.setAdUnitId(getString(R.string.banner_ad_unit_id));
104     mAdView.setAdSize(AdSize.SMART_BANNER);
105     mAdFrameLayout.addView(mAdView);
106     mAdView.loadAd(new AdRequest.Builder().build());
107     */
108
109     openChild = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMN_COLUMNS_CHILDS, manager.CHILDREN_ID + "=?", new String[]{id}, null);
110     openChild.moveToFirst();
111     String ogender = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_GENDER));
112     String oname = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_NAME));
113     openChild.close();
114     name = (TextView)findViewById(R.id.name);
115     name.setText(oname);
116
117     layoutGender = (LinearLayout)findViewById(R.id.layoutGender);
118
119     if(ogender.equals("0")) {
120         //layoutGender.setBackgroundColor(Color.parseColor("#5500238C"));
121     }
122     else {
123
124         int sdk = android.os.Build.VERSION.SDK_INT;
125
126         if(sdk < android.os.Build.VERSION_CODES.JELLY_BEAN) {
127             layoutGender.setBackgroundDrawable(getResources().getDrawable(R.drawable.
pink_name_title));
128         } else {
129             layoutGender.setBackground(getResources().getDrawable(R.drawable.pink_name_title));
130         }
131         //childLayout.setBackground(v.getResources().getDrawable(R.drawable.pink_title));
132     }
133

```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\NewAwardActivity.java

```
134     prizename = (EditText)findViewById(R.id.prizename);
135     prizepoints = (EditText)findViewById(R.id.prizepoints);
136     photo = (CircleImageView)findViewById(R.id.photo);
137     takephoto = (Button)findViewById(R.id.takephoto);
138     prizeForAll = (CheckBox)findViewById(R.id.prizeForAll);
139     takephoto.setOnClickListener(new View.OnClickListener() {
140         @Override
141         public void onClick(View v) {
142             Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
143             startActivityForResult(cameraIntent, CAMERA_REQUEST);
144         }
145     });
146     if(!prizeid.equals("")) {
147         openPrize = getContentResolver().query(ChildProvider.PRIZES_URI, manager.
COLUMNNS_PRIZES, manager.PRIZES_ID + "=?", new String[]{prizeid}, null);
148         openPrize.moveToFirst();
149         byte[] image = openPrize.getBlob(openPrize.getColumnIndex(manager.PRIZES_PHOTO));
150         oprizeid = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_ID));
151         oprizechildid = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_CHILD_ID));
152         if(oprizechildid.equals("0")){
153             prizeForAll.setChecked(true);
154         }
155
156         oprizename = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_NAME));
157         oprizegiven = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_GIVEN));
158         oprizephoto = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_PHOTO));
159         oprizepoints = openPrize.getString(openPrize.getColumnIndex(manager.PRIZES_POINTS));
160         prizename.setText(oprizename);
161         prizepoints.setText(oprizepoints);
162         if(image.length > 3) {
163             File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
164                 Environment.DIRECTORY_PICTURES),
165                 "TinPoint");
166
167             final String photoStorePath = imagesFolder.getAbsolutePath();
168
169             final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
170                 + java.io.File.separator + oprizephoto));
171             photo.setImageURI(null);
172             photo.setImageURI(fileUri);
173             photoName = oprizephoto;
174             flag = 1;
175         }else if(!oprizephoto.equals("NA")){
176             photo.setImageResource(getResources().getIdentifier(oprizephoto, "drawable", this.
getPackageName()));
177             photoName = oprizephoto;
178             flag = 1;
179         }else{
180             photo.setImageResource(R.drawable.ic_gift);
181         }
182
183         prizefor = (TextView)findViewById(R.id.prizefor);
184         prizefor.setText(getResources().getString(R.string.EditPrizeFor));
185
186     }
187     galleryphoto = (Button)findViewById(R.id.galleryphoto);
188     galleryphoto.setOnClickListener(new View.OnClickListener() {
189         @Override
190         public void onClick(View v) {
191             Intent galleryIntent = new Intent(Intent.ACTION_PICK,
192                 android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
193             startActivityForResult(galleryIntent, RESULT_LOAD_IMG);
194         }
195     });
196
197     registerContextMenu(photo);
198     photo.setOnClickListener(new View.OnClickListener() {
199         @Override
```

```

200     public void onClick(View v) {
201         openContextMenu(photo);
202     }
203 };
204
205 save = (Button)findViewById(R.id.save);
206 save.setOnClickListener(new View.OnClickListener() {
207     @Override
208     public void onClick(View v) {
209         if(!prizename.getText().toString().equals(""))
210             && !prizepoints.getText().toString().equals("")
211         ) {
212             if(prizepoints.getText().toString().length()>0 && !prizepoints.getText().toString().equals("0"))
213         {
214             String saveprizename = prizename.getText().toString();
215             String saveprizepoints = prizepoints.getText().toString();
216
217             String savephoto = "ig";
218             if (flag == 1) {
219                 savephoto = photoName;
220             }
221             oprizechildid = id;
222             if(prizeForAll.isChecked()){
223                 oprizechildid = "0";
224             }
225             if(!prizeid.equals("")) {
226                 mTracker.send(new HitBuilders.EventBuilder()
227                     .setCategory("NewPrize")
228                     .setAction("Edit Prize")
229                     .build());
230                 getContentResolver().update(ChildProvider.PRIZES_URI, manager.genPrizeValues(
231                     Integer.valueOf(oprizechildid), saveprizename, savephoto, oprizegiven, Integer.valueOf(saveprizepoints)
232                     ), manager.PRIZES_ID + "=?", new String[]{oprizeid});
233                 Intent openIntent = new Intent().setClass(NewAwardActivity.this, OpenChildActivity.
234                     class);
235                 openIntent.putExtra("id", id);
236                 setResult(Activity.RESULT_OK, openIntent);
237                 finish();
238                 overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
239             } else {
240                 mTracker.send(new HitBuilders.EventBuilder()
241                     .setCategory("NewPrize")
242                     .setAction("Create Prize")
243                     .build());
244                 getContentResolver().insert(ChildProvider.PRIZES_URI, manager.genPrizeValues(
245                     Integer.valueOf(oprizechildid), saveprizename, savephoto, "0", Integer.valueOf(saveprizepoints)));
246                 Intent mainIntent = new Intent();
247                 setResult(Activity.RESULT_OK, mainIntent);
248                 finish();
249                 overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
250             }
251         } else {
252             Toast toast1 = Toast.makeText(getApplicationContext(),
253                 getResources().getString(R.string.intNumberOnly), Toast.LENGTH_SHORT);
254             toast1.show();
255         }
256     }
257 }
258 });
259 String[] columnNames = {"_id", "icon"};
260 MatrixCursor cursor = new MatrixCursor(columnNames);
261 String[] array = getResources().getStringArray(R.array.icons_array);
262 String[] temp = new String[2];

```

```

263     int id = 0;
264     for(String item : array){
265         temp[0] = Integer.toString(id++);
266         temp[1] = item;
267         cursor.addRow(temp);
268     }
269
270
271     adapter = new GenericPrizeCursorAdapter(this, cursor, 0);
272
273
274
275
276
277     icons = (Button)findViewById(R.id.icons);
278     icons.setOnClickListener(new View.OnClickListener() {
279         @Override
280         public void onClick(View v) {
281
282             callIcons();
283
284         }
285     });
286 }
287
288 @Override
289 public void onCreateContextMenu(ContextMenu menu, View v,
290                               ContextMenu.ContextMenuInfo menuInfo) {
291     super.onCreateContextMenu(menu, v, menuInfo);
292     if (v.getId()==R.id.photo) {
293
294         String[] menuItems = {getResources().getString(R.string.replaceWithPhoto), getResources().
getString(R.string.replaceWithGallery), getResources().getString(R.string.replaceWithIcon)};
295         for (int i = 0; i<menuItems.length; i++) {
296             menu.add(Menu.NONE, i, i, menuItems[i]);
297         }
298     }
299 }
300
301 @Override
302 public boolean onContextItemSelected(Menu.Item item) {
303     int menuItemIndex = item.getItemId();
304
305     if(menuItemIndex == 0){
306         Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
307         startActivityForResult(cameraIntent, CAMERA_REQUEST);
308     }else if(menuItemIndex == 1){
309         Intent galleryIntent = new Intent(Intent.ACTION_PICK,
310             android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
311         startActivityForResult(galleryIntent, RESULT_LOAD_IMG);
312     }else if(menuItemIndex == 2){
313         callIcons();
314     }
315     return true;
316 }
317 @Override
318 public boolean onOptionsItemSelected(Menu.Item item) {
319
320     switch (item.getItemId()) {
321         case android.R.id.home:
322             // API 5+ solution
323             onBackPressed();
324             return true;
325
326         default:
327             return super.onOptionsItemSelected(item);
328     }
329

```

```

330     }
331
332     @Override
333     public void onBackPressed() {
334         super.onBackPressed();
335         overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
336     }
337
338     private File getProductPhotoDirectory() {
339         //get directory where file should be stored
340         File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
341             Environment.DIRECTORY_PICTURES),
342             "TinPoint");
343         if (!imagesFolder.exists()){
344             if (!imagesFolder.mkdirs()){
345                 Log.d("TinPoint", "failed to create directory");
346                 return null;
347             }
348         }
349         return imagesFolder;
350     }
351
352     private Uri getPhotoFileUri(final String photoStorePath) {
353
354         //timestamp used in file name
355         final String timestamp = new SimpleDateFormat("yyyyMMdd_HHmms",
356             Locale.US).format(new Date());
357
358         // file uri with timestamp
359         final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
360             + java.io.File.separator + "IMG_" + timestamp + ".jpg"));
361
362         photoName = "IMG_" + timestamp + ".jpg";
363
364         return fileUri;
365     }
366
367     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
368         if (requestCode == CAMERA_REQUEST && resultCode == RESULT_OK) {
369             mTracker.send(new HitBuilders.EventBuilder()
370                 .setCategory("NewPrize")
371                 .setAction("Take Photo")
372                 .build());
373             Bitmap pic = (Bitmap) data.getExtras().get("data");
374             Uri selectedImage = data.getData();
375             final String photoStorePath = getProductPhotoDirectory().getAbsolutePath();
376
377             //create file uri
378             final Uri destinationUri = getPhotoFileUri(photoStorePath);
379             //dateView.setText(destinationUri.toString());
380             //dateView.setText(selectedImage.toString());
381             UCrop uCrop = UCrop.of(selectedImage, destinationUri);
382             uCrop = advancedConfig(uCrop);
383             uCrop.withAspectRatio(1, 1)
384                 .withMaxResultSize(600, 600)
385                 .start(this);
386         }
387         if (requestCode == RESULT_LOAD_IMG && resultCode == RESULT_OK && null != data) {
388             // Get the Image from data
389
390             mTracker.send(new HitBuilders.EventBuilder()
391                 .setCategory("NewPrize")
392                 .setAction("Gallery Pic")
393                 .build());
394
395             Uri selectedImage = data.getData();
396             final String photoStorePath = getProductPhotoDirectory().getAbsolutePath();
397

```

```

398     //create file uri
399     final Uri destinationUri = getPhotoFileUri(photoStorePath);
400     //dateView.setText(destinationUri.toString());
401     //dateView.setText(selectedImage.toString());
402     UCrop uCrop = UCrop.of(selectedImage, destinationUri);
403     uCrop = advancedConfig(uCrop);
404     uCrop.withAspectRatio(1, 1)
405         .withMaxResultSize(600, 600)
406         .start(this);
407
408 }
409 if (requestCode == PIC_CROP) {
410     if (data != null) {
411         // get the returned data
412         Bundle extras = data.getExtras();
413         // get the cropped bitmap
414         Bitmap selectedBitmap = extras.getParcelable("data");
415
416         photo.setImageBitmap(selectedBitmap);
417     }
418 }
419 if (resultCode == RESULT_OK && requestCode == UCrop.REQUEST_CROP) {
420     resultUri = UCrop.getOutput(data);
421     photo.setImageURI(resultUri);
422     flag = 1;
423     String path = resultUri.getPath();
424     String idStr = path.substring(path.lastIndexOf('/') + 1);
425     photoName = idStr;
426 } else if (resultCode == UCrop.RESULT_ERROR) {
427     final Throwable cropError = UCrop.getError(data);
428     Toast toast = Toast.makeText(this, cropError.toString(), Toast.LENGTH_SHORT);
429     toast.show();
430 }
431 }
432
433 private UCrop advancedConfig(@NonNull UCrop uCrop) {
434     UCrop.Options options = new UCrop.Options();
435     /*
436     Tune everything (ノ◕▽◕)ノ*:·° ✧
437     options.setMaxScaleMultiplier(5);
438     options.setImageToCropBoundsAnimDuration(666);
439     options.setDimmedLayerColor(Color.CYAN);
440     options.setOvalDimmedLayer(true);
441     options.setShowCropFrame(false);
442     options.setCropGridStrokeWidth(20);
443     options.setCropGridColor(Color.GREEN);
444     options.setCropGridColumnCount(2);
445     options.setCropGridRowCount(1);
446     */
447
448     return uCrop.withOptions(options);
449 }
450
451 private void callIcons(){
452     final AlertDialog.Builder alertDialog = new AlertDialog.Builder(NewAwardActivity.this);
453
454
455     LayoutInflater inflater = getLayoutInflater();
456     View convertView = inflater.inflate(R.layout.icon_prizes, null);
457     alertDialog.setView(convertView);
458     alertDialog.setTitle(getResources().getString(R.string.icons));
459
460     final AlertDialog alert = alertDialog.create();
461
462
463     listIcons = (ExpandableHeightGridView) convertView.findViewById(R.id.iconsprizesgridview);
464     listIcons.setExpanded(true);
465

```



File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\NewAwardActivity.java

```
466     listIcons.setOnItemClickListener(new AdapterView.OnItemClickListener() {
467         @Override
468         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
469             mTracker.send(new HitBuilders.EventBuilder()
470                 .setCategory("NewPrize")
471                 .setAction("Set Icon")
472                 .build());
473             Cursor cursor = (Cursor) parent.getItemAtPosition(position);
474             final String icon = cursor.getString(cursor.getColumnIndex("icon"));
475             photo.setImageResource(getResources().getIdentifier(icon, "drawable", NewAwardActivity.
this.getPackageName()));
476             photoName = icon;
477             flag = 1;
478             alert.dismiss();
479         }
480     });
481
482     listIcons.setAdapter(adapter);
483     alert.show();
484 }
485
486 public static void hideSoftKeyboard(Activity activity) {
487     InputMethodManager inputMethodManager = (InputMethodManager) activity.getSystemService(
Activity.INPUT_METHOD_SERVICE);
488     inputMethodManager.hideSoftInputFromWindow(activity.getCurrentFocus().getWindowToken(), 0);
489 }
490 public void setupUI(View view) {
491
492     //Set up touch listener for non-text box views to hide keyboard.
493     if (!(view instanceof EditText)) {
494
495         view.setOnTouchListener(new View.OnTouchListener() {
496
497             public boolean onTouch(View v, MotionEvent event) {
498                 hideSoftKeyboard(NewAwardActivity.this);
499                 return false;
500             }
501
502         });
503     }
504
505     //If a layout container, iterate over children and seed recursion.
506     if (view instanceof ViewGroup) {
507
508         for (int i = 0; i < ((ViewGroup) view).getChildCount(); i++) {
509
510             View innerView = ((ViewGroup) view).getChildAt(i);
511
512             setupUI(innerView);
513         }
514     }
515 }
516 }
517
```



```
1 package com.mushoq.inpoint;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.app.DatePickerDialog;
6 import android.app.Dialog;
7 import android.app.LoaderManager;
8 import android.content.ActivityNotFoundException;
9 import android.content.ContentResolver;
10 import android.content.CursorLoader;
11 import android.content.DialogInterface;
12 import android.content.Intent;
13 import android.content.Loader;
14 import android.database.Cursor;
15 import android.graphics.Bitmap;
16 import android.graphics.BitmapFactory;
17 import android.graphics.Color;
18 import android.net.Uri;
19 import android.os.Environment;
20 import android.provider.MediaStore;
21 import android.support.annotation.NonNull;
22 import android.support.v4.app.NavUtils;
23 import android.support.v7.app.AppCompatActivity;
24 import android.os.Bundle;
25 import android.util.Log;
26 import android.view.ContextMenu;
27 import android.view.Menu;
28 import android.view.MenuItem;
29 import android.view.MotionEvent;
30 import android.view.View;
31 import android.view.ViewGroup;
32 import android.view.inputmethod.InputMethodManager;
33 import android.widget.AdapterView;
34 import android.widget.AdapterView.Adapter;
35 import android.widget.Button;
36 import android.widget.DatePicker;
37 import android.widget.EditText;
38 import android.widget.FrameLayout;
39 import android.widget.ImageView;
40 import android.widget.Spinner;
41 import android.widget.TextView;
42 import android.widget.Toast;
43
44 import com.google.android.gms.ads.AdRequest;
45 import com.google.android.gms.ads.AdSize;
46 import com.google.android.gms.ads.AdView;
47 import com.google.android.gms.analytics.HitBuilders;
48 import com.google.android.gms.analytics.Tracker;
49 import com.yalantis.ucrop.UCrop;
50
51 import java.io.File;
52 import java.text.SimpleDateFormat;
53 import java.util.Calendar;
54 import java.util.Date;
55 import java.util.Locale;
56
57 import de.hdodenhof.circleimageview.CircleImageView;
58
59 public class NewChildActivity extends AppCompatActivity{
60
61     private Calendar calendar;
62     private TextView dateView, greentitle;
63     private EditText name;
64     private int year, month, day, flag = 0;
65     private Integer opoints = 0;
66     private Spinner gender;
67     private static final int CAMERA_REQUEST = 1888, RESULT_LOAD_IMG = 2;
68     private CircleImageView photo;
```

```

69  private Button takephoto, galleryphoto, save, getdate;
70  final int PIC_CROP = 5;
71  private DbManager manager;
72  private Uri resultUri;
73  private String photoName, id;
74  private Cursor openChild;
75  private static final int LOADER_ID = 1;
76  private FrameLayout mAdFrameLayout;
77  private Tracker mTracker;
78  private AnalyticsApplication application;
79
80  @Override
81  protected void onCreate(Bundle savedInstanceState) {
82      super.onCreate(savedInstanceState);
83      setContentView(R.layout.activity_new_child);
84
85      Bundle extras = getIntent().getExtras();
86      if (extras != null) {
87          id = extras.getString("id");
88      }else{
89          id = "";
90      }
91      getSupportActionBar().setTitle("TinPoint");
92      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
93
94      setupUI(findViewById(R.id.parent));
95
96      application = (AnalyticsApplication) getApplication();
97      mTracker = application.getDefaultTracker();
98
99      mTracker.setScreenName("New Child");
100     mTracker.send(new HitBuilders.ScreenViewBuilder().build());
101
102     manager = new DbManager(this);
103
104     /**ADS BY GOOGLE
105     mAdFrameLayout = (FrameLayout)findViewById(R.id.addsFrameLayout);
106
107     AdView mAdView = new AdView(this);
108     mAdView.setAdUnitId(getString(R.string.banner_ad_unit_id));
109     mAdView.setAdSize(AdSize.SMART_BANNER);
110     mAdFrameLayout.addView(mAdView);
111     mAdView.loadAd(new AdRequest.Builder().build());*/
112
113     photo = (CircleImageView)findViewById(R.id.photo);
114     name = (EditText)findViewById(R.id.name);
115
116     dateView = (TextView) findViewById(R.id.textView3);
117     greentitle = (TextView)findViewById(R.id.greentitle);
118     calendar = Calendar.getInstance();
119     year = calendar.get(Calendar.YEAR) - 4;
120
121     month = calendar.get(Calendar.MONTH);
122     day = calendar.get(Calendar.DAY_OF_MONTH);
123     //showDate(year, month + 1, day);
124     getdate = (Button)findViewById(R.id.getdate);
125
126     gender = (Spinner)findViewById(R.id.gender);
127
128     String []opciones={getString(R.string.male),getString(R.string.female)};
129     ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.my_spinner, opciones);
130     adapter.setDropDownViewResource(R.layout.my_spinner_item);
131     gender.setAdapter(adapter);
132
133     gender.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener) {
134         @Override
135         public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position,
long id) {

```

```

136     if (flag == 0) {
137         if (position == 0) {
138             photo.setImageResource(R.drawable.boy);
139         } else {
140             photo.setImageResource(R.drawable.girl);
141         }
142     }
143 }
144
145 @Override
146 public void onNothingSelected(AdapterView<?> parentView) {
147     // your code here
148 }
149
150 });
151
152 takephoto = (Button)findViewById(R.id.takephoto);
153 takephoto.setOnClickListener(new View.OnClickListener() {
154     @Override
155     public void onClick(View v) {
156         Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
157         startActivityForResult(cameraIntent, CAMERA_REQUEST);
158     }
159 });
160
161 galleryphoto = (Button)findViewById(R.id.galleryphoto);
162 galleryphoto.setOnClickListener(new View.OnClickListener() {
163     @Override
164     public void onClick(View v) {
165         Intent galleryIntent = new Intent(Intent.ACTION_PICK,
166             android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
167         startActivityForResult(galleryIntent, RESULT_LOAD_IMG);
168     }
169 });
170
171 if(!lid.equals("")) {
172     openChild = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMN_COLUMNS_CHILDS, manager.CHILDREN_ID + "=?", new String[]{id}, null); //manager.
loadOpenChilds(id);
173     openChild.moveToFirst();
174     byte[] image = openChild.getBlob(openChild.getColumnIndex(manager.CHILDREN_PHOTO));
175     String ogender = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_GENDER
));
176     String birthdate = openChild.getString(openChild.getColumnIndex(manager.
CHILDREN_BIRTHDATE));
177     String oname = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_NAME));
178     String ophoto = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_PHOTO));
179     opoints = openChild.getInt(openChild.getColumnIndex(manager.CHILDREN_POINTS));
180     if(ogender.equals("0"))
181         gender.setSelection(0);
182     else
183         gender.setSelection(1);
184     if(image.length > 3) {
185         File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
Environment.DIRECTORY_PICTURES),
186             "TinPoint");
187
188         final String photoStorePath = imagesFolder.getAbsolutePath();
189
190         final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
+ java.io.File.separator + ophoto));
191         photo.setImageURI(null);
192         photo.setImageURI(fileUri);
193         flag = 1;
194         photoName = ophoto;
195     }else{
196         if(ogender.equals("0"))
197             photo.setImageResource(R.drawable.boy);

```

```

200         else
201             photo.setImageResource(R.drawable.girl);
202     }
203
204     name.setText(oname);
205     dateView.setText(birthdate);
206     greentitle.setText(getResources().getString(R.string.Editfields));
207     getdate.setText(getResources().getString(R.string.changeDate));
208     openChild.close();
209
210     String[] separated = birthdate.split("-");
211
212     year = Integer.valueOf(separated[0]);
213
214     month = Integer.valueOf(separated[1]) - 1;
215     day = Integer.valueOf(separated[2]);
216 }
217
218 registerForContextMenu(photo);
219 photo.setOnClickListener(new View.OnClickListener() {
220     @Override
221     public void onClick(View v) {
222         openContextMenu(photo);
223     }
224 });
225
226 save = (Button)findViewById(R.id.save);
227 save.setOnClickListener(new View.OnClickListener() {
228     @Override
229     public void onClick(View v) {
230         if(!name.getText().toString().equals("")
231             && !dateView.getText().toString().equals("")
232             ) {
233             String savename = name.getText().toString();
234             String savebirthdate = dateView.getText().toString();
235             String savephoto = "NA";
236             if (flag == 1) {
237                 savephoto = photoName;
238             }
239             String savegender = String.valueOf(gender.getSelectedItemPosition());
240             if(!id.equals("")) {
241                 //manager.updateChild(id, savename, savebirthdate, savephoto, opoints, savegender);
242                 mTracker.send(new HitBuilders.EventBuilder()
243                     .setCategory("NewChild")
244                     .setAction("Edit Child")
245                     .build());
246                 getContentResolver().update(ChildProvider.CHILDS_URI, manager.genChildValues(
savename, savebirthdate, savephoto, opoints, savegender), manager.CHILDREN_ID + "=?", new
String[]{id});
247                 Intent openIntent = new Intent().setClass(NewChildActivity.this, OpenChildActivity.class)
;
248                 openIntent.putExtra("id", id);
249                 setResult(RESULT_OK, openIntent);
250                 finish();
251                 setResult(Activity.RESULT_OK, openIntent);
252                 overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
253             }else{
254                 //manager.insertChild(savename, savebirthdate, savephoto, opoints, savegender);
255                 mTracker.send(new HitBuilders.EventBuilder()
256                     .setCategory("NewChild")
257                     .setAction("Create Child")
258                     .build());
259                 getContentResolver().insert(ChildProvider.CHILDS_URI, manager.genChildValues(
savename, savebirthdate, savephoto, opoints, savegender));
260                 Intent mainIntent = new Intent().setClass(NewChildActivity.this, MainActivity.class);
261                 finish();
262                 setResult(Activity.RESULT_OK, mainIntent);
263                 overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);

```

```

264         }
265     }
266
267     } else {
268         Toast toast1 = Toast.makeText(getApplicationContext(),
269             getResources().getString(R.string.allFields), Toast.LENGTH_SHORT);
270         toast1.show();
271     }
272 }
273 });
274 }
275
276 @Override
277 public void onCreateContextMenu(ContextMenu menu, View v,
278     ContextMenu.ContextMenuInfo menuInfo) {
279     super.onCreateContextMenu(menu, v, menuInfo);
280     if (v.getId()==R.id.photo) {
281
282         String[] menuItems = {getResources().getString(R.string.replaceWithPhoto), getResources().
283             getString(R.string.replaceWithGallery)};
284         for (int i = 0; i<menuItems.length; i++) {
285             menu.add(Menu.NONE, i, i, menuItems[i]);
286         }
287     }
288
289 @Override
290 public boolean onOptionsItemSelected(MenuItem item) {
291     int menuItemIndex = item.getItemId();
292
293     if(menuItemIndex == 0){
294         Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
295         startActivityForResult(cameraIntent, CAMERA_REQUEST);
296     }else if(menuItemIndex == 1){
297         Intent galleryIntent = new Intent(Intent.ACTION_PICK,
298             android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
299         startActivityForResult(galleryIntent, RESULT_LOAD_IMG);
300     }
301     return true;
302 }
303
304 @Override
305 public boolean onOptionsItemSelected(MenuItem item) {
306
307     switch (item.getItemId()) {
308         case android.R.id.home:
309             // API 5+ solution
310             onBackPressed();
311             return true;
312
313         default:
314             return super.onOptionsItemSelected(item);
315     }
316 }
317
318 @Override
319 public void onBackPressed() {
320     super.onBackPressed();
321     overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
322 }
323
324 private File getProductPhotoDirectory() {
325     //get directory where file should be stored
326     File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
327         Environment.DIRECTORY_PICTURES),
328         "TinPoint");
329     if (! imagesFolder.exists()){

```

File - D:\work\Tinpoint\TinpointAndroid\app\src\main\java\com\mushoq\Tinpoint\NewChildActivity.java

```
331     if (! imagesFolder.mkdirs()){
332         Log.d("TinPoint", "failed to create directory");
333         return null;
334     }
335 }
336 return imagesFolder;
337 }
338
339 private Uri getPhotoFileUri(final String photoStorePath) {
340
341     //timestamp used in file name
342     final String timestamp = new SimpleDateFormat("yyyyMMdd_HHmms",
343         Locale.US).format(new Date());
344
345     // file uri with timestamp
346     final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
347         + java.io.File.separator + "IMG_" + timestamp + ".jpg"));
348
349     photoName = "IMG_" + timestamp + ".jpg";
350
351     return fileUri;
352 }
353
354 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
355     if (requestCode == CAMERA_REQUEST && resultCode == RESULT_OK) {
356
357         mTracker.send(new HitBuilders.EventBuilder()
358             .setCategory("NewChild")
359             .setAction("Take Photo")
360             .build());
361         Uri selectedImage = data.getData();
362         final String photoStorePath = getProductPhotoDirectory().getAbsolutePath();
363
364         //create file uri
365         final Uri destinationUri = getPhotoFileUri(photoStorePath);
366         UCrop uCrop = UCrop.of(selectedImage, destinationUri);
367         uCrop = advancedConfig(uCrop);
368         uCrop.withAspectRatio(1, 1)
369             .withMaxResultSize(800, 800)
370             .start(this);
371     }
372     if (requestCode == RESULT_LOAD_IMG && resultCode == RESULT_OK && null != data) {
373         // Get the Image from data
374         mTracker.send(new HitBuilders.EventBuilder()
375             .setCategory("NewChild")
376             .setAction("Gallery Pic")
377             .build());
378         Uri selectedImage = data.getData();
379         final String photoStorePath = getProductPhotoDirectory().getAbsolutePath();
380
381         final Uri destinationUri = getPhotoFileUri(photoStorePath);
382
383         UCrop uCrop = UCrop.of(selectedImage, destinationUri);
384         uCrop = advancedConfig(uCrop);
385         uCrop.withAspectRatio(1, 1)
386             .withMaxResultSize(800, 800)
387             .start(this);
388     }
389 }
390 if (requestCode == PIC_CROP) {
391     if (data != null) {
392         // get the returned data
393         Bundle extras = data.getExtras();
394         // get the cropped bitmap
395         Bitmap selectedBitmap = extras.getParcelable("data");
396
397         photo.setImageBitmap(selectedBitmap);
398     }
399 }
```

```

399     }
400     if (resultCode == RESULT_OK && requestCode == UCrop.REQUEST_CROP) {
401         resultUri = UCrop.getOutput(data);
402         photo.setImageURI(resultUri);
403         flag = 1;
404         String path = resultUri.getPath();
405         String idStr = path.substring(path.lastIndexOf('/') + 1);
406         photoName = idStr;
407     } else if (resultCode == UCrop.RESULT_ERROR) {
408         final Throwable cropError = UCrop.getError(data);
409         Toast toast = Toast.makeText(this, cropError.toString(), Toast.LENGTH_SHORT);
410         toast.show();
411     }
412 }
413
414 private UCrop advancedConfig(@NonNull UCrop uCrop) {
415     UCrop.Options options = new UCrop.Options();
416     /*
417      * Tune everything (ノ◕▽◕)ノ*:·° ☆
418      * options.setMaxScaleMultiplier(5);
419      * options.setImageToCropBoundsAnimDuration(666);
420      * options.setDimmedLayerColor(Color.CYAN);
421      * options.setOvalDimmedLayer(true);
422      * options.setShowCropFrame(false);
423      * options.setCropGridStrokeWidth(20);
424      * options.setCropGridColor(Color.GREEN);
425      * options.setCropGridColumnCount(2);
426      * options.setCropGridRowCount(1);
427      */
428
429     return uCrop.withOptions(options);
430 }
431
432 @SuppressWarnings("deprecation")
433 public void setDate(View view) {
434     showDialog(999);
435     //Toast.makeText(getApplicationContext(), "ca", Toast.LENGTH_SHORT).show();
436 }
437 @SuppressWarnings("deprecation")
438 @Override
439 protected Dialog onCreateDialog(int id) {
440     // TODO Auto-generated method stub
441     if (id == 999) {
442         return new DatePickerDialog(this, myDateListener, year, month, day);
443     }
444     return null;
445 }
446
447 private DatePickerDialog.OnDateSetListener myDateListener = new DatePickerDialog.
OnDateSetListener() {
448     @Override
449     public void onDateSet(DatePicker arg0, int arg1, int arg2, int arg3) {
450         // TODO Auto-generated method stub
451         // arg1 = year
452         // arg2 = month
453         // arg3 = day
454         showDate(arg1, arg2+1, arg3);
455     }
456 };
457
458 private void showDate(int year, int month, int day) {
459     //dateView.setText(new StringBuilder().append(day).append("/").append(month).append("/").append(
year));
460     String dia = String.valueOf(day);
461     String mes = String.valueOf(month);
462     String anio = String.valueOf(year);
463     if(month < 10){
464

```



```
465     mes = "0" + mes;
466     }
467     if(day < 10){
468
469         dia = "0" + dia ;
470     }
471     dateView.setText(anio + "-" + mes + "-" + dia);
472     getdate.setText(getResources().getString(R.string.changeDate));
473 }
474
475 public static void hideSoftKeyboard(Activity activity) {
476     InputMethodManager inputMethodManager = (InputMethodManager) activity.getSystemService(
Activity.INPUT_METHOD_SERVICE);
477     inputMethodManager.hideSoftInputFromWindow(activity.getCurrentFocus().getWindowToken(), 0);
478 }
479 public void setupUI(View view) {
480
481     //Set up touch listener for non-text box views to hide keyboard.
482     if (!(view instanceof EditText)) {
483
484         view.setOnTouchListener(new View.OnTouchListener() {
485
486             public boolean onTouch(View v, MotionEvent event) {
487                 hideSoftKeyboard(NewChildActivity.this);
488                 return false;
489             }
490
491         });
492     }
493
494     //If a layout container, iterate over children and seed recursion.
495     if (view instanceof ViewGroup) {
496
497         for (int i = 0; i < ((ViewGroup) view).getChildCount(); i++) {
498
499             View innerView = ((ViewGroup) view).getChildAt(i);
500
501             setupUI(innerView);
502         }
503     }
504 }
505 }
506
```



```
1  package com.mushoq.tinpoint;
2
3  import android.animation.Animator;
4  import android.animation.AnimatorListenerAdapter;
5  import android.animation.AnimatorSet;
6  import android.animation.ObjectAnimator;
7  import android.app.Activity;
8  import android.app.AlertDialog;
9  import android.app.LoaderManager;
10 import android.content.CursorLoader;
11 import android.content.DialogInterface;
12 import android.content.Intent;
13 import android.database.Cursor;
14 import android.database.MatrixCursor;
15 import android.graphics.Point;
16 import android.graphics.Rect;
17 import android.media.MediaPlayer;
18 import android.net.Uri;
19 import android.os.Environment;
20 import android.support.v7.app.AppCompatActivity;
21 import android.os.Bundle;
22 import android.view.ContextMenu;
23 import android.view.LayoutInflater;
24 import android.view.Menu;
25 import android.view.MenuItem;
26 import android.view.View;
27 import android.view.animation.DecelerateInterpolator;
28 import android.widget.AdapterView;
29 import android.widget.Button;
30 import android.widget.FrameLayout;
31 import android.widget.LinearLayout;
32 import android.widget.TextView;
33 import android.widget.Toast;
34
35 import com.google.android.gms.ads.AdRequest;
36 import com.google.android.gms.ads.AdSize;
37 import com.google.android.gms.ads.AdView;
38 import com.google.android.gms.analytics.HitBuilders;
39 import com.google.android.gms.analytics.Tracker;
40 import com.guardanis.applock.UnlockDialogBuilder;
41 import com.guardanis.applock.locking.ActionLockingHelper;
42
43 import java.io.File;
44 import java.util.Calendar;
45 import java.util.HashSet;
46 import java.util.Set;
47
48 import de.hdodenhof.circleimageview.CircleImageView;
49
50 public class OpenChildActivity extends AppCompatActivity implements LoaderManager.
LoaderCallbacks<Cursor> {
51
52     private DbManager manager;
53     private String id, childid, gender, birthdate, oname, ophoto;
54     public Integer opoints, numeroPuntos;
55     private Cursor openChild;
56     private PrizeCursorAdapter adapter;
57     private PrizeGivenCursorAdapter adaptergiven;
58     private PointsCursorAdapter adapterPoints;
59     private ExpandableHeightGridView listPrizes, prizesawarded;
60     private ExpandableHeightGridView listPoints;
61     private TextView name, tage, points, notprizes, textprizesawarded, aviablePrizes;
62     private CircleImageView photo;
63     private LinearLayout layoutGender;
64     private Button upPoint, editProfile, setAward;
65     private static final int SET_PRIZE = 154, EDIT_PROFILE = 155;
66     private LoaderManager.LoaderCallbacks<Cursor> mCallbacks;
67     private static final int LOADER_PRIZES = 1;
```

```

68  private static final int LOADER_PRIZES_GIVEN = 2;
69  private byte[] image;
70  private FrameLayout mAdFrameLayout;
71  private Animator mCurrentAnimator;
72  private int mShortAnimationDuration;
73  private static Set<MediaPlayer> activePlayers = new HashSet<MediaPlayer>();
74  private Tracker mTracker;
75  private AnalyticsApplication application;
76
77  @Override
78  protected void onCreate(Bundle savedInstanceState) {
79      super.onCreate(savedInstanceState);
80      setContentView(R.layout.activity_open_child);
81
82      getSupportActionBar().setTitle("TinPoint");
83
84      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
85
86      manager = new DbManager(this);
87
88      application = (AnalyticsApplication) getApplication();
89      mTracker = application.getDefaultTracker();
90
91      mTracker.enableAdvertisingIdCollection(true);
92
93      mTracker.setScreenName("Open Child");
94      mTracker.send(new HitBuilders.ScreenViewBuilder().build());
95
96      //ADS BY GOOGLE
97      mAdFrameLayout = (FrameLayout)findViewById(R.id.addsFrameLayout);
98
99      AdView mAdView = new AdView(this);
100     mAdView.setAdUnitId(getString(R.string.banner_ad_unit_id));
101     mAdView.setAdSize(AdSize.SMART_BANNER);
102     mAdFrameLayout.addView(mAdView);
103     mAdView.loadAd(new AdRequest.Builder().build());
104
105
106     mCallbacks = this;
107
108     Bundle extras = getIntent().getExtras();
109     id = extras.getString("id");
110     childid = id;
111     Bundle bundle = new Bundle();
112     bundle.putString("id", id);
113
114     LoaderManager lm = getLoaderManager();
115     lm.initLoader(LOADER_PRIZES, bundle, mCallbacks);
116     lm.initLoader(LOADER_PRIZES_GIVEN, bundle, mCallbacks);
117
118     loadChild();
119
120     listPrizes = (ExpandableHeightGridView)findViewById(R.id.listPrizes);
121     listPrizes.setExpanded(true);
122
123
124     adapter = new PrizeCursorAdapter(this, null, 0, opoints);
125     listPrizes.setAdapter(adapter);
126
127
128     listPrizes.setOnItemClickListener(new AdapterView.OnItemClickListener() {
129         @Override
130         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
131             Cursor cursor = (Cursor) parent.getItemAtPosition(position);
132             final String prizeid = cursor.getString(cursor.getColumnIndex(manager.PRIZES_ID));
133             final Integer prizechildid = cursor.getInt(cursor.getColumnIndex(manager.
PRIZES_CHILD_ID));
134             final Integer prizepoints = cursor.getInt(cursor.getColumnIndex(manager.PRIZES_POINTS)

```

```

134 );
135         final String prizename = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME)
);
136         final String prizephoto = cursor.getString(cursor.getColumnIndex(manager.
PRIZES_PHOTO));
137         if (prizepoints <= opoints) {
138             new AlertDialog.Builder(view.getContext())
139                 .setIcon(R.drawable.alert_gift)
140                 .setTitle(prizename)
141                 .setMessage(getResources().getString(R.string.areYouSurePrize))
142                 .setPositiveButton(getResources().getString(R.string.yes), new DialogInterface.
OnClickListener() {
143                     @Override
144                     public void onClick(DialogInterface dialog, int which) {
145
146                         mTracker.send(new HitBuilders.EventBuilder()
147                             .setCategory("OpenChild")
148                             .setAction("Give Prize")
149                             .build());
150                         if(prizechildid != 0) {
151                             getContentResolver().update(ChildProvider.PRIZES_URI, manager.
genPrizeValues(prizechildid, prizename, prizephoto, "1", prizepoints), manager.PRIZES_ID + "=",
new String[]{prizeid});
152                         }else {
153                             getContentResolver().insert(ChildProvider.PRIZES_URI, manager.
genPrizeValues(Integer.valueOf(childid), prizename, prizephoto, "1", prizepoints));
154                         }
155                         getContentResolver().update(ChildProvider.CHILDS_URI, manager.
genChildValues(oname, birthdate, ophoto, opoints - prizepoints, gender), manager.CHILDREN_ID +
"=?", new String[]{childid});
156
157                         upPointFunction(prizepoints, prizename, prizephoto);
158
159                     }
160
161                 })
162                 .setNegativeButton(getResources().getString(R.string.no), null)
163                 .show();
164             } else
165                 Toast.makeText(getApplicationContext(), getResources().getString(R.string.
ContinuesToAddPointsToRedeemThisPrize), Toast.LENGTH_LONG).show();
166         }
167     });
168
169     listPrizes.setLongClickable(true);
170     registerContextMenu(listPrizes);
171
172     String[] columnNames = {"_id", "points"};
173     MatrixCursor cursor = new MatrixCursor(columnNames);
174     String[] array = getResources().getStringArray(R.array.points_array);
175     String[] temp = new String[2];
176     int id = 0;
177     for(String item : array){
178         temp[0] = Integer.toString(id++);
179         temp[1] = item;
180         cursor.addRow(temp);
181     }
182
183
184     adapterPoints = new PointsCursorAdapter(this, cursor, 0);
185
186     upPoint = (Button)findViewById(R.id.upPoint);
187     upPoint.setOnClickListener(new View.OnClickListener() {
188         @Override
189         public void onClick(View v) {
190             new AlertDialog.Builder(v.getContext())
191                 .setIcon(R.drawable.ic_star)
192                 .setTitle(getResources().getString(R.string.scoreapoint))

```

File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\OpenChildActivity.java

```
193         .setMessage(getResources().getString(R.string.areYouSurePoint))
194         .setPositiveButton(getResources().getString(R.string.yes), new DialogInterface.
OnClickListener() {
195             @Override
196             public void onClick(DialogInterface dialog, int which) {
197                 callPoints();
198             }
199         })
200     .setNegativeButton(getResources().getString(R.string.no), null)
201     .show();
202 }
203 });
204
205
206
207
208 editProfile = (Button)findViewById(R.id.editProfile);
209 editProfile.setOnClickListener(new View.OnClickListener() {
210     @Override
211     public void onClick(View v) {
212         Intent intent = new Intent(OpenChildActivity.this, NewChildActivity.class);
213         intent.putExtra("id", String.valueOf(childid));
214         startActivityForResult(intent, EDIT_PROFILE);
215         overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
216     }
217 });
218
219 setAward = (Button)findViewById(R.id.setAward);
220 setAward.setOnClickListener(new View.OnClickListener() {
221     @Override
222     public void onClick(View v) {
223         Intent intent = new Intent(OpenChildActivity.this, NewAwardActivity.class);
224         intent.putExtra("id", String.valueOf(childid));
225         startActivityForResult(intent, SET_PRIZE);
226         overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
227     }
228 });
229
230 //cursorPrizesGiven = manager.loadGivenPrizes(id);
231 prizesawarded = (ExpandableHeightGridView)findViewById(R.id.prizesawarded);
232 prizesawarded.setExpanded(true);
233
234 adaptergiven = new PrizeGivenCursorAdapter(this, null, 0);
235 prizesawarded.setAdapter(adaptergiven);
236 prizesawarded.setLongClickable(true);
237 prizesawarded.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
238     @Override
239     public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
240         Cursor cursor = (Cursor) parent.getItemAtPosition(position);
241         final String prizeid = cursor.getString(cursor.getColumnIndex(manager.PRIZES_ID));
242         final String prizename = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME));
243
244         new AlertDialog.Builder(view.getContext())
245             .setIcon(android.R.drawable.ic_menu_delete)
246             .setTitle(getResources().getString(R.string.remove) + " " + prizename)
247             .setMessage(getResources().getString(R.string.arwYouSureDeletePrize))
248             .setPositiveButton(getResources().getString(R.string.yes), new DialogInterface.
OnClickListener() {
249                 @Override
250                 public void onClick(DialogInterface dialog, int which) {
251                     mTracker.send(new HitBuilders.EventBuilder()
252                         .setCategory("OpenChild")
253                         .setAction("Remove Given Prize")
254                         .build());
255                     getContentResolver().delete(ChildProvider.PRIZES_URI, manager.PRIZES_ID +
"=?", new String[] {prizeid});
256
257                     adaptergiven.changeCursor(loadCursorGiven());
```

```

258         }
259
260     })
261     .setNegativeButton(getResources().getString(R.string.no), null)
262     .show();
263     return true;
264 }
265 });
266
267 prizesawarded.setOnItemClickListener(new AdapterView.OnItemClickListener() {
268     @Override
269     public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
270         Cursor cursor = (Cursor) parent.getItemAtPosition(position);
271         final String prizephoto = cursor.getString(cursor.getColumnIndex(manager.PRIZES_PHOTO));
272         final String prizename = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME));
273         final String prizepoints = cursor.getString(cursor.getColumnIndex(manager.PRIZES_POINTS));
274
275         prizeAlert(prizename, prizephoto, prizepoints, true);
276     }
277 });
278
279 }
280
281 private void callPoints(){
282     final android.support.v7.app.AlertDialog.Builder alertDialog = new android.support.v7.app.
AlertDialog.Builder(OpenChildActivity.this);
283
284     numeroPuntos = 0;
285
286     LayoutInflater inflater = getLayoutInflater();
287     View convertView = inflater.inflate(R.layout.give_points, null);
288     alertDialog.setView(convertView);
289     alertDialog.setTitle(getResources().getString(R.string.howManyPoints));
290
291     final android.support.v7.app.AlertDialog alert = alertDialog.create();
292
293
294     listPoints = (ExpandableHeightGridView) convertView.findViewById(R.id.pointsgridview);
295     listPoints.setExpanded(true);
296
297     listPoints.setOnItemClickListener(new AdapterView.OnItemClickListener() {
298         @Override
299         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
300
301             final Integer darpuntos = position + 1;
302
303             ActionLockingHelper.unlockIfRequired(OpenChildActivity.this, new UnlockDialogBuilder.
UnlockEventListener() {
304                 public void onCancel() {
305                     } // Dialog was closed without entry
306
307                 public void onUnlockFailed(String reason) {
308                     } // Not called with default Dialog, instead is handled internally
309
310                 public void onUnlockSuccessful() {
311                     numeroPuntos = darpuntos;
312
313                     mTracker.send(new HitBuilders.EventBuilder()
314                         .setCategory("OpenChild")
315                         .setAction("Score a Point")
316                         .build());
317                     getContentResolver().update(ChildProvider.CHILDS_URI, manager.genChildValues(
oname, birthdate, ophoto, opoints + numeroPuntos, gender), manager.CHILDREN_ID + "=?", new
String[][]{childid});
318                     loadChild();
319                     adapter = new PrizeCursorAdapter(OpenChildActivity.this, null, 0, opoints);
320                     adapter.changeCursor(loadCursorPrizes());

```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\OpenChildActivity.java

```
321         adaptergiven.changeCursor(loadCursorGiven());
322         listPrizes.setAdapter(adapter);
323     }
324     });
325     alert.dismiss();
326 }
327 });
328 listPoints.setAdapter(adapterPoints);
329 alert.show();
330 }
331
332 private Cursor loadCursorGiven(){
333     Cursor cursor = getContentResolver().query(ChildProvider.PRIZES_URI, manager.
COLUMNS_PRIZES, manager.PRIZES_CHILD_ID + "=? AND " + manager.PRIZES_GIVEN + " = ?",
new String[]{childid, "1"}, manager.PRIZES_POINTS + " DESC");
334     if(cursor.getCount() > 0) {
335         textprizesawarded.setVisibility(View.VISIBLE);
336     }else{
337         textprizesawarded.setVisibility(View.GONE);
338     }
339     return cursor;
340 }
341 private Cursor loadCursorPrizes(){
342     Cursor cursor = getContentResolver().query(ChildProvider.PRIZES_URI, manager.
COLUMNS_PRIZES, "(" + manager.PRIZES_CHILD_ID + "=? OR "+manager.PRIZES_CHILD_ID+
" = ? ) AND " + manager.PRIZES_GIVEN + " = ?", new String[]{childid, "0", "0"}, manager.
PRIZES_POINTS);
343     if(cursor.getCount() > 0) {
344         notprizes.setVisibility(View.GONE);
345         aviablePrizes.setVisibility(View.VISIBLE);
346     }else{
347         notprizes.setVisibility(View.VISIBLE);
348         aviablePrizes.setVisibility(View.GONE);
349     }
350     return cursor;
351 }
352 private void loadChild(){
353     openChild = getContentResolver().query(ChildProvider.CHILDS_URI, manager.
COLUMNS_CHILDS, manager.CHILDREN_ID + "=?", new String[]{id}, null);
354     openChild.moveToFirst();
355
356     name = (TextView)findViewById(R.id.name);
357     photo = (CircleImageView)findViewById(R.id.photo);
358     points = (TextView)findViewById(R.id.points);
359     notprizes = (TextView)findViewById(R.id.notprizes);
360     textprizesawarded = (TextView)findViewById(R.id.textprizesawarded);
361     textprizesawarded.setVisibility(View.GONE);
362     aviablePrizes = (TextView)findViewById(R.id.aviablePrizes);
363     aviablePrizes.setVisibility(View.GONE);
364     image = openChild.getBlob(openChild.getColumnIndex(manager.CHILDREN_PHOTO));
365     gender = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_GENDER));
366     birthdate = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_BIRTHDATE));
367     oname = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_NAME));
368     ophoto = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_PHOTO));
369     opoints = openChild.getInt(openChild.getColumnIndex(manager.CHILDREN_POINTS));
370
371     name.setText(oname);
372     points.setText(String.valueOf(opoints));
373     if(image.length > 3) {
374         File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
Environment.DIRECTORY_PICTURES),
375             "TinPoint");
376         String pic = openChild.getString(openChild.getColumnIndex(manager.CHILDREN_PHOTO));
377         final String photoStorePath = imagesFolder.getAbsolutePath();
378
379
380         final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
381             + java.io.File.separator + pic));
382         photo.setImageURI(null);
```

```

383     photo.setImageURI(fileUri);
384
385     mShortAnimationDuration = getResources().getInteger(android.R.integer.config_shortAnimTime);
386     photo.setOnClickListener(new View.OnClickListener() {
387         @Override
388         public void onClick(View v) {
389             zoomImageFromThumb(photo, fileUri);
390         }
391     });
392
393
394     }else{
395         if(gender.equals("0"))
396             photo.setImageResource(R.drawable.boy);
397         else
398             photo.setImageResource(R.drawable.girl);
399     }
400
401     openChild.close();
402
403     String[] separated = birthdate.split("-");
404     String formatAge = getAge(Integer.parseInt(separated[0]), Integer.parseInt(separated[1]), Integer.
parseInt(separated[2]));
405
406     tage = (TextView)findViewById(R.id.age);
407     tage.setText(getResources().getString(R.string.age) + ": " + formatAge + " " + getResources().
getString(R.string.yearsold));
408
409     layoutGender = (LinearLayout)findViewById(R.id.layoutGender);
410
411     if(gender.equals("0")) {
412         //layoutGender.setBackgroundColor(Color.parseColor("#5500238C"));
413     }
414     else {
415
416         int sdk = android.os.Build.VERSION.SDK_INT;
417
418         if(sdk < android.os.Build.VERSION_CODES.JELLY_BEAN) {
419             layoutGender.setBackgroundDrawable(getResources().getDrawable(R.drawable.
pink_name_title));
420         } else {
421             layoutGender.setBackground(getResources().getDrawable(R.drawable.pink_name_title));
422         }
423         //childLayout.setBackground(v.getResources().getDrawable(R.drawable.pink_title));
424     }
425 }
426
427 @Override
428 public boolean onOptionsItemSelected(MenuItem item) {
429
430     switch (item.getItemId()) {
431         case android.R.id.home:
432             // API 5+ solution
433             onBackPressed();
434             return true;
435
436         default:
437             return super.onOptionsItemSelected(item);
438     }
439 }
440 }
441
442 @Override
443 public void onBackPressed() {
444     super.onBackPressed();
445     overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
446 }
447

```



```

448     @Override
449     public void onCreateContextMenu(ContextMenu menu, View v,
450         ContextMenu.ContextMenuInfo menuInfo) {
451         if (v.getId()==R.id.listPrizes) {
452             AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)menuInfo;
453             String[] menuItems = {getResources().getString(R.string.edit), getResources().getString(R.string.
remove)});
454             for (int i = 0; i<menuItems.length; i++) {
455                 menu.add(Menu.NONE, i, i, menuItems[i]);
456             }
457         }
458     }
459
460     @Override
461     public boolean onOptionsItemSelected(MenuItem item) {
462         AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)item.
getMenuInfo();
463         int itemPosition = info.position;
464         Cursor cursor = (adapter).getCursor();
465         cursor.moveToPosition(itemPosition);
466         final String prizeid = cursor.getString(cursor.getColumnIndex(manager.PRIZES_ID));
467         final String prizename = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME));
468         int menuItemIndex = item.getItemId();
469
470         if(menuItemIndex == 0){
471             Intent intent = new Intent(OpenChildActivity.this, NewAwardActivity.class);
472             intent.putExtra("id", id);
473             intent.putExtra("prizeid", String.valueOf(prizeid));
474             startActivityForResult(intent, SET_PRIZE);
475             overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
476         }else if(menuItemIndex == 1){
477             new AlertDialog.Builder(this)
478                 .setIcon(android.R.drawable.ic_menu_delete)
479                 .setTitle(getResources().getString(R.string.remove) + " " + prizename)
480                 .setMessage(getResources().getString(R.string.arwYouSureDeletePrize))
481                 .setPositiveButton(getResources().getString(R.string.yes), new DialogInterface.
OnClickListener() {
482                     @Override
483                     public void onClick(DialogInterface dialog, int which) {
484                         mTracker.send(new HitBuilders.EventBuilder()
485                             .setCategory("OpenChild")
486                             .setAction("Remove Prize")
487                             .build());
488                         getContentResolver().delete(ChildProvider.PRIZES_URI, manager.PRIZES_ID + "=?",
new String[] {prizeid});
489                         adapter.changeCursor(loadCursorPrizes());
490                     }
491                 })
492             .setNegativeButton(getResources().getString(R.string.no), null)
493             .show();
494         }
495     }
496     return true;
497 }
498
499 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
500
501     if (requestCode == SET_PRIZE) {
502         if(resultCode == Activity.RESULT_OK){
503
504             adapter.changeCursor(loadCursorPrizes());
505             //adapter = new PrizeCursorAdapter(this, null, 0, opoints);
506             listPrizes.setAdapter(adapter);
507         }
508         if (resultCode == Activity.RESULT_CANCELED) {
509             //Write your code if there's no result
510         }
511     }else if (requestCode == EDIT_PROFILE) {

```



File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\OpenChildActivity.java

```
512     loadChild();
513     }
514 }
515
516 private void upPointFunction(int prizepoints, String prizename, String prizephoto){
517
518     prizeAlert(prizename, prizephoto, "", false);
519
520     MediaPlayer mPlayer2;
521     mPlayer2= MediaPlayer.create(OpenChildActivity.this, R.raw.applause);
522     activePlayers.add(mPlayer2);
523     mPlayer2.setOnCompletionListener(releaseOnFinishListener);
524     mPlayer2.start();
525     opoints = opoints - prizepoints;
526     adapter = new PrizeCursorAdapter(OpenChildActivity.this, null, 0, opoints);
527     adapter.changeCursor(loadCursorPrizes());
528     adaptergiven.changeCursor(loadCursorGiven());
529     listPrizes.setAdapter(adapter);
530
531     points.setText(String.valueOf(opoints));
532 }
533
534 MediaPlayer.OnCompletionListener releaseOnFinishListener = new MediaPlayer.
OnCompletionListener() {
535     public void onCompletion(MediaPlayer mp) {
536         mp.release();
537         activePlayers.remove(mp);
538     }
539 };
540
541 private void prizeAlert(String prizename, String prizepic, String prizepoints, Boolean entregado){
542     final android.support.v7.app.AlertDialog.Builder alertDialog = new android.support.v7.app.
AlertDialog.Builder(OpenChildActivity.this);
543
544
545     LayoutInflater inflater = getLayoutInflater();
546     View convertView = inflater.inflate(R.layout.congratulations, null);
547     alertDialog.setView(convertView);
548     if(entregado)
549         alertDialog.setTitle(getResources().getString(R.string.GivenPrize));
550     else
551         alertDialog.setTitle(getResources().getString(R.string.Congratulations));
552
553     final android.support.v7.app.AlertDialog alert = alertDialog.create();
554     alert.setButton(AlertDialog.BUTTON_NEUTRAL, "OK",
555         new DialogInterface.OnClickListener() {
556             public void onClick(DialogInterface dialog, int which) {
557                 dialog.dismiss();
558             }
559         });
560
561
562     CircleImageView prizephoto = (CircleImageView) convertView.findViewById(R.id.prizephoto);
563     File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
564         Environment.DIRECTORY_PICTURES),
565         "TinPoint");
566     if(prizepic.length() > 3){
567         String photoStorePath = imagesFolder.getAbsolutePath();
568         final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
569             + java.io.File.separator + prizepic));
570         prizephoto.setImageURI(null);
571         prizephoto.setImageURI(fileUri);
572     }else{
573         prizephoto.setImageResource(convertView.getResources().getIdentifier(prizepic, "drawable",
getPackageName()));
574     }
575
576
```

File - D:\work\inpoint\inpointAndroid\app\src\main\java\com\mushoq\inpoint\OpenChildActivity.java

```
577     TextView alertprizename = (TextView)convertView.findViewById(R.id.prizename);
578     alertprizename.setText(prizename);
579
580     TextView alertGivePrize = (TextView)convertView.findViewById(R.id.givePrize);
581     if(entregado)
582         alertGivePrize.setText(prizepoints + "" + getResources().getString(R.string.points));
583     else
584         alertGivePrize.setText(getResources().getString(R.string.YouHaveGivenThisPrize));
585
586     alert.show();
587 }
588
589 private String getAge(int year, int month, int day){
590     Calendar dob = Calendar.getInstance();
591     Calendar today = Calendar.getInstance();
592
593     dob.set(year, month, day);
594
595     int age = today.get(Calendar.YEAR) - dob.get(Calendar.YEAR);
596
597     if (today.get(Calendar.DAY_OF_YEAR) < dob.get(Calendar.DAY_OF_YEAR)){
598         age--;
599     }
600
601     Integer ageInt = new Integer(age);
602     String ageS = ageInt.toString();
603
604     return ageS;
605 }
606
607 @Override
608 public android.content.Loader<Cursor> onCreateLoader(int id, Bundle args) {
609     if(id == LOADER_PRIZES) {
610         CursorLoader cursorloader = new CursorLoader(OpenChildActivity.this, ChildProvider.
PRIZES_URI, manager.COLUMNS_PRIZES, "(" + manager.PRIZES_CHILD_ID + "=? OR " + manager.
PRIZES_CHILD_ID+"=? ) AND " + manager.PRIZES_GIVEN + "=? ", new String[]{args.getString("id
"), "0", "0"}, manager.PRIZES_POINTS);
611         return cursorloader;
612     }else if(id == LOADER_PRIZES_GIVEN) {
613         CursorLoader cursorloader = new CursorLoader(OpenChildActivity.this, ChildProvider.
PRIZES_URI, manager.COLUMNS_PRIZES, manager.PRIZES_CHILD_ID + "=? AND " + manager.
PRIZES_GIVEN + "=? ", new String[]{args.getString("id"), "1"}, manager.PRIZES_POINTS + " DESC
");
614         return cursorloader;
615     }else{
616         return null;
617     }
618 }
619
620 @Override
621 public void onLoadFinished(android.content.Loader<Cursor> loader, Cursor data) {
622     switch (loader.getId()) {
623         case LOADER_PRIZES:
624             // The asynchronous load is complete and the data
625             // is now available for use. Only now can we associate
626             // the queried Cursor with the SimpleCursorAdapter.
627             adapter.changeCursor(data);
628             if(data.getCount() > 0) {
629                 notprizes.setVisibility(View.GONE);
630                 aviablePrizes.setVisibility(View.VISIBLE);
631             }
632             break;
633         case LOADER_PRIZES_GIVEN:
634             // The asynchronous load is complete and the data
635             // is now available for use. Only now can we associate
636             // the queried Cursor with the SimpleCursorAdapter.
637             adaptergiven.changeCursor(data);
638             if(data.getCount() > 0)
```

```

639         textprizesawarded.setVisibility(View.VISIBLE);
640         break;
641     }
642 }
643
644 @Override
645 public void onLoaderReset(android.content.Loader<Cursor> loader) {
646     adapter.swapCursor(null);
647     adaptergiven.swapCursor(null);
648 }
649
650 private void zoomImageFromThumb(final View thumbView, Uri fileUri) {
651     // If there's an animation in progress, cancel it
652     // immediately and proceed with this one.
653     if (mCurrentAnimator != null) {
654         mCurrentAnimator.cancel();
655     }
656
657     // Load the high-resolution "zoomed-in" image.
658     final CircleImageView expandedImageView = (CircleImageView) findViewById(
659         R.id.expanded_image);
660     expandedImageView.setImageURI(fileUri);
661
662     // Calculate the starting and ending bounds for the zoomed-in image.
663     // This step involves lots of math. Yay, math.
664     final Rect startBounds = new Rect();
665     final Rect finalBounds = new Rect();
666     final Point globalOffset = new Point();
667
668     // The start bounds are the global visible rectangle of the thumbnail,
669     // and the final bounds are the global visible rectangle of the container
670     // view. Also set the container view's offset as the origin for the
671     // bounds, since that's the origin for the positioning animation
672     // properties (X, Y).
673     thumbView.getGlobalVisibleRect(startBounds);
674     findViewById(R.id.container)
675         .getGlobalVisibleRect(finalBounds, globalOffset);
676     startBounds.offset(-globalOffset.x, -globalOffset.y);
677     finalBounds.offset(-globalOffset.x, -globalOffset.y);
678
679     // Adjust the start bounds to be the same aspect ratio as the final
680     // bounds using the "center crop" technique. This prevents undesirable
681     // stretching during the animation. Also calculate the start scaling
682     // factor (the end scaling factor is always 1.0).
683     float startScale;
684     if ((float) finalBounds.width() / finalBounds.height()
685         > (float) startBounds.width() / startBounds.height()) {
686         // Extend start bounds horizontally
687         startScale = (float) startBounds.height() / finalBounds.height();
688         float startWidth = startScale * finalBounds.width();
689         float deltaWidth = (startWidth - startBounds.width()) / 2;
690         startBounds.left -= deltaWidth;
691         startBounds.right += deltaWidth;
692     } else {
693         // Extend start bounds vertically
694         startScale = (float) startBounds.width() / finalBounds.width();
695         float startHeight = startScale * finalBounds.height();
696         float deltaHeight = (startHeight - startBounds.height()) / 2;
697         startBounds.top -= deltaHeight;
698         startBounds.bottom += deltaHeight;
699     }
700
701     // Hide the thumbnail and show the zoomed-in view. When the animation
702     // begins, it will position the zoomed-in view in the place of the
703     // thumbnail.
704     thumbView.setAlpha(0f);
705     expandedImageView.setVisibility(View.VISIBLE);
706

```

```

707 // Set the pivot point for SCALE_X and SCALE_Y transformations
708 // to the top-left corner of the zoomed-in view (the default
709 // is the center of the view).
710 expandedImageView.setPivotX(0f);
711 expandedImageView.setPivotY(0f);
712
713 // Construct and run the parallel animation of the four translation and
714 // scale properties (X, Y, SCALE_X, and SCALE_Y).
715 AnimatorSet set = new AnimatorSet();
716 set
717     .play(ObjectAnimator.ofFloat(expandedImageView, View.X,
718         startBounds.left, finalBounds.left))
719     .with(ObjectAnimator.ofFloat(expandedImageView, View.Y,
720         startBounds.top, finalBounds.top))
721     .with(ObjectAnimator.ofFloat(expandedImageView, View.SCALE_X,
722         startScale, 1f)).with(ObjectAnimator.ofFloat(expandedImageView,
723         View.SCALE_Y, startScale, 1f));
724 set.setDuration(mShortAnimationDuration);
725 set.setInterpolator(new DecelerateInterpolator());
726 set.addListener(new AnimatorListenerAdapter() {
727     @Override
728     public void onAnimationEnd(Animator animation) {
729         mCurrentAnimator = null;
730     }
731
732     @Override
733     public void onAnimationCancel(Animator animation) {
734         mCurrentAnimator = null;
735     }
736 });
737 set.start();
738 mCurrentAnimator = set;
739
740 // Upon clicking the zoomed-in image, it should zoom back down
741 // to the original bounds and show the thumbnail instead of
742 // the expanded image.
743 final float startScaleFinal = startScale;
744 expandedImageView.setOnClickListener(new View.OnClickListener() {
745     @Override
746     public void onClick(View view) {
747         if (mCurrentAnimator != null) {
748             mCurrentAnimator.cancel();
749         }
750
751         // Animate the four positioning/sizing properties in parallel,
752         // back to their original values.
753         AnimatorSet set = new AnimatorSet();
754         set.play(ObjectAnimator
755             .ofFloat(expandedImageView, View.X, startBounds.left))
756             .with(ObjectAnimator
757                 .ofFloat(expandedImageView,
758                     View.Y, startBounds.top))
759             .with(ObjectAnimator
760                 .ofFloat(expandedImageView,
761                     View.SCALE_X, startScaleFinal))
762             .with(ObjectAnimator
763                 .ofFloat(expandedImageView,
764                     View.SCALE_Y, startScaleFinal));
765         set.setDuration(mShortAnimationDuration);
766         set.setInterpolator(new DecelerateInterpolator());
767         set.addListener(new AnimatorListenerAdapter() {
768             @Override
769             public void onAnimationEnd(Animator animation) {
770                 thumbView.setAlpha(1f);
771                 expandedImageView.setVisibility(View.GONE);
772                 mCurrentAnimator = null;
773             }
774

```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\OpenChildActivity.java

```
775         @Override
776         public void onAnimationCancel(Animator animation) {
777             thumbView.setAlpha(1f);
778             expandedImageView.setVisibility(View.GONE);
779             mCurrentAnimator = null;
780         }
781     });
782     set.start();
783     mCurrentAnimator = set;
784 }
785 });
786 }
787
788 }
789
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\ChildCursorAdapter.java

```
1 package com.mushoq.tinpoint;
2
3 import android.content.Context;
4 import android.database.Cursor;
5
6 import android.net.Uri;
7 import android.os.Environment;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11 import android.widget.CursorAdapter;
12 import android.widget.LinearLayout;
13 import android.widget.TextView;
14 import java.io.File;
15 import java.util.Calendar;
16 import de.hdodenhof.circleimageview.CircleImageView;
17
18 /**
19  * Created by santiagoe on 2/11/16.
20  */
21 public class ChildCursorAdapter extends CursorAdapter {
22
23
24     private Context context;
25     private DbManager manager;
26
27     public ChildCursorAdapter(Context context, Cursor c, int flags) {
28         super(context, c, flags);
29         this.context = context;
30     }
31
32     @Override
33     public View newView(Context context, Cursor cursor, ViewGroup parent) {
34         LayoutInflater inflater = (LayoutInflater) context.getSystemService(context.
LAYOUT_INFLATER_SERVICE);
35         View view = inflater.inflate(R.layout.row_layout_child, parent, false);
36         return view;
37     }
38
39     @Override
40     public void bindView(final View view, final Context context, final Cursor cursor) {
41         manager = new DbManager(this.context);
42         View v = view;
43
44
45         String name = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_NAME));
46         String points = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_POINTS));
47         String birthdate = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_BIRTHDATE));
48         String gender = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_GENDER));
49         String photo = cursor.getString(cursor.getColumnIndex(manager.CHILDREN_PHOTO));
50         byte[] image = cursor.getBlob(cursor.getColumnIndex(manager.CHILDREN_PHOTO));
51
52         String[] separated = birthdate.split("-");
53         String formatAge = getAge(Integer.parseInt(separated[0]), Integer.parseInt(separated[1]), Integer.
parseInt(separated[2]));
54
55         TextView tage = (TextView) v.findViewById(R.id.age);
56         tage.setText(v.getResources().getString(R.string.age) + ": " + formatAge + " " + v.getResources().
getString(R.string.yearsold));
57
58         CircleImageView iv = (CircleImageView) v.findViewById(R.id.photo);
59
60         if (image != null) {
61             if (image.length > 3)
62                 {
63                 File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
64                     Environment.DIRECTORY_PICTURES),
65                     "TinPoint");
```

```

66
67     final String photoStorePath = imagesFolder.getAbsolutePath();
68     final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
69         + java.io.File.separator + photo));
70     iv.setImageURI(null);
71     iv.setImageURI(fileUri);
72 }
73 else
74 {
75     if(gender.equals("0"))
76         iv.setImageResource(R.drawable.boy);
77     else
78         iv.setImageResource(R.drawable.girl);
79 }
80 }
81 LinearLayout childLayout = (LinearLayout) v.findViewById(R.id.childlayout);
82
83 TextView tname = (TextView) v.findViewById(R.id.name);
84 tname.setText(name);
85
86 TextView tpoints = (TextView) v.findViewById(R.id.points);
87 tpoints.setText(points);
88 int sdk = android.os.Build.VERSION.SDK_INT;
89
90 if(gender.equals("0")) {
91     if(sdk < android.os.Build.VERSION_CODES.JELLY_BEAN) {
92         childLayout.setBackgroundDrawable(v.getResources().getDrawable(R.drawable.blue_title));
93     } else {
94         childLayout.setBackground(v.getResources().getDrawable(R.drawable.blue_title));
95     }
96 }
97 else {
98     if(sdk < android.os.Build.VERSION_CODES.JELLY_BEAN) {
99         childLayout.setBackgroundDrawable(v.getResources().getDrawable(R.drawable.pink_title));
100    } else {
101        childLayout.setBackground(v.getResources().getDrawable(R.drawable.pink_title));
102    }
103 }
104 }
105 }
106
107 private String getAge(int year, int month, int day){
108     Calendar dob = Calendar.getInstance();
109     Calendar today = Calendar.getInstance();
110
111     dob.set(year, month, day);
112
113     int age = today.get(Calendar.YEAR) - dob.get(Calendar.YEAR);
114
115     if (today.get(Calendar.DAY_OF_YEAR) < dob.get(Calendar.DAY_OF_YEAR)){
116         age--;
117     }
118
119     Integer ageInt = new Integer(age);
120     String ageS = ageInt.toString();
121
122     return ageS;
123 }
124 }
125

```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\PrizeCursorAdapter.java

```
1  package com.mushoq.tinpoint;
2
3  import android.content.Context;
4  import android.database.Cursor;
5  import android.net.Uri;
6  import android.os.Environment;
7  import android.view.LayoutInflater;
8  import android.view.View;
9  import android.view.ViewGroup;
10 import android.widget.CursorAdapter;
11 import android.widget.LinearLayout;
12 import android.widget.SimpleCursorAdapter;
13 import android.widget.TextView;
14
15 import java.io.File;
16
17 import de.hdodenhof.circleimageview.CircleImageView;
18
19 /**
20  * Created by santiago on 2/16/16.
21  */
22 public class PrizeCursorAdapter extends CursorAdapter {
23
24     private Context context;
25     private Integer opoints;
26     private DbManager manager;
27     private LinearLayout layoutPrize;
28
29     public PrizeCursorAdapter(Context context, Cursor c, int flags, Integer opoints) {
30         super(context, c, flags);
31         this.context = context;
32         this.opoints = opoints;
33     }
34
35     @Override
36     public View newView(Context context, Cursor cursor, ViewGroup parent) {
37         LayoutInflater inflater = (LayoutInflater) context.getSystemService(context.
38 LAYOUT_INFLATER_SERVICE);
39         View view = inflater.inflate(R.layout.row_layout_prize, parent, false);
40         return view;
41     }
42
43     @Override
44     public void bindView(final View view, final Context context, final Cursor cursor) {
45         manager = new DbManager(this.context);
46         View v = view;
47         String name = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME));
48         String points = cursor.getString(cursor.getColumnIndex(manager.PRIZES_POINTS));
49         Integer pointsto = cursor.getInt(cursor.getColumnIndex(manager.PRIZES_POINTS));
50         String photo = cursor.getString(cursor.getColumnIndex(manager.PRIZES_PHOTO));
51         byte[] image = cursor.getBlob(cursor.getColumnIndex(manager.PRIZES_PHOTO));
52         CircleImageView iv = (CircleImageView) v.findViewById(R.id.prizephoto);
53
54         if(this.opoints < pointsto){
55             layoutPrize = (LinearLayout)v.findViewById(R.id.layoutPrize);
56             layoutPrize.setAlpha((float)0.5);
57         }
58
59         if (image != null) {
60             // If there is no image in the database "NA" is stored instead of a blob
61             // test if there more than 3 chars "NA" + a terminating char if more than
62             // there is an image otherwise load the default
63             if(image.length > 4)
64             {
65                 File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
66                     Environment.DIRECTORY_PICTURES),
67                     "TinPoint");
68                 final String photoStorePath = imagesFolder.getAbsolutePath();
```



```
68
69     final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
70         + java.io.File.separator + photo));
71     iv.setImageURI(null);
72     iv.setImageURI(fileUri);
73 }else{
74     iv.setImageResource(v.getResources().getIdentifier(photo, "drawable", context.
getPackageName()));
75 }
76 }
77 TextView prizename = (TextView) v.findViewById(R.id.prizename);
78 prizename.setText(name);
79
80 TextView prizepoints = (TextView) v.findViewById(R.id.prizepoints);
81 prizepoints.setText(points + " " + v.getResources().getString(R.string.points));
82 }
83
84
85 }
86
```

File - D:\work\pinpoint\pinpointAndroid\app\src\main\java\com\mushoq\pinpoint\PointsCursorAdapter.java

```
1 package com.mushoq.pinpoint;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.CursorAdapter;
9 import android.widget.TextView;
10
11 /**
12  * Created by santiagoe on 3/17/16.
13  */
14 public class PointsCursorAdapter extends CursorAdapter {
15
16     public PointsCursorAdapter(Context context, Cursor c, int flags) {
17         super(context, c, flags);
18     }
19
20     public View newView(Context context, Cursor cursor, ViewGroup parent) {
21         LayoutInflater inflater = (LayoutInflater) context.getSystemService(context.
LAYOUT_INFLATER_SERVICE);
22         View view = inflater.inflate(R.layout.row_layout_points, parent, false);
23         return view;
24     }
25     @Override
26     public void bindView(final View view, final Context context, final Cursor cursor) {
27
28         View v = view;
29
30         String points = cursor.getString(cursor.getColumnIndex("points"));
31         TextView givepoints = (TextView) v.findViewById(R.id.givepoints);
32         givepoints.setText(points);
33     }
34 }
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\AnalyticsApplication.java

```
1 package com.mushoq.tinpoint;
2
3 import android.app.Application;
4
5 import com.google.android.gms.analytics.GoogleAnalytics;
6 import com.google.android.gms.analytics.Logger;
7 import com.google.android.gms.analytics.Tracker;
8
9 /**
10  * This is a subclass of {@link Application} used to provide shared objects for this app, such as
11  * the {@link Tracker}.
12  */
13 public class AnalyticsApplication extends Application {
14     private Tracker mTracker;
15
16     /**
17      * Gets the default {@link Tracker} for this {@link Application}.
18      * @return tracker
19      */
20     synchronized public Tracker getDefaultTracker() {
21         if (mTracker == null) {
22             GoogleAnalytics analytics = GoogleAnalytics.getInstance(this);
23             // To enable debug logging use: adb shell setprop log.tag.GAv4 DEBUG
24             mTracker = analytics.newTracker(R.xml.global_tracker);
25         }
26         return mTracker;
27     }
28 }
29
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\PrizeGivenCursorAdapter.java

```
1  package com.mushoq.tinpoint;
2
3  import android.content.Context;
4  import android.database.Cursor;
5  import android.net.Uri;
6  import android.os.Environment;
7  import android.view.LayoutInflater;
8  import android.view.View;
9  import android.view.ViewGroup;
10 import android.widget.CursorAdapter;
11 import android.widget.LinearLayout;
12 import android.widget.SimpleCursorAdapter;
13 import android.widget.TextView;
14
15 import java.io.File;
16
17 import de.hdodenhof.circleimageview.CircleImageView;
18
19 /**
20  * Created by santiagoe on 2/19/16.
21  */
22 public class PrizeGivenCursorAdapter extends CursorAdapter {
23     private Context context;
24     private DbManager manager;
25     public PrizeGivenCursorAdapter(Context context, Cursor c, int flags) {
26         super(context, c, flags);
27         this.context = context;
28     }
29
30     public View newView(Context context, Cursor cursor, ViewGroup parent) {
31         LayoutInflater inflater = (LayoutInflater) context.getSystemService(context.
LAYOUT_INFLATER_SERVICE);
32         View view = inflater.inflate(R.layout.row_layout_prize_given, parent, false);
33         return view;
34     }
35     @Override
36     public void bindView(final View view, final Context context, final Cursor cursor) {
37         manager = new DbManager(this.context);
38         View v = view;
39         String name = cursor.getString(cursor.getColumnIndex(manager.PRIZES_NAME));
40
41
42         String photo = cursor.getString(cursor.getColumnIndex(manager.PRIZES_PHOTO));
43         byte[] image = cursor.getBlob(cursor.getColumnIndex(manager.PRIZES_PHOTO));
44         CircleImageView iv = (CircleImageView) v.findViewById(R.id.prizephoto);
45
46
47         if (image != null) {
48             // If there is no image in the database "NA" is stored instead of a blob
49             // test if there more than 3 chars "NA" + a terminating char if more than
50             // there is an image otherwise load the default
51             if(image.length > 4)
52             {
53                 File imagesFolder = new File(Environment.getExternalStoragePublicDirectory(
54                     Environment.DIRECTORY_PICTURES),
55                     "TinPoint");
56                 final String photoStorePath = imagesFolder.getAbsolutePath();
57
58                 final Uri fileUri = Uri.fromFile(new java.io.File(photoStorePath
59                     + java.io.File.separator + photo));
60                 iv.setImageURI(null);
61                 iv.setImageURI(fileUri);
62             }else{
63                 iv.setImageResource(v.getResources().getIdentifier(photo, "drawable", context.
64                     getPackageName()));
65             }
66             TextView prizename = (TextView) v.findViewById(R.id.prizenam);
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\PrizeGivenCursorAdapter.java

```
67     prizename.setText(name);
68 }
69 }
70
```

```
1 package com.mushoq.tinpoint;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.view.ViewGroup;
6 import android.widget.GridView;
7
8 /**
9  * Created by santiago on 2/16/16.
10 */
11 public class ExpandableHeightGridView extends GridView {
12     boolean expanded = false;
13
14     public ExpandableHeightGridView(Context context)
15     {
16         super(context);
17     }
18
19     public ExpandableHeightGridView(Context context, AttributeSet attrs)
20     {
21         super(context, attrs);
22     }
23
24     public ExpandableHeightGridView(Context context, AttributeSet attrs,
25                                     int defStyle)
26     {
27         super(context, attrs, defStyle);
28     }
29
30     public boolean isExpanded()
31     {
32         return expanded;
33     }
34
35     @Override
36     public void onMeasure(int widthMeasureSpec, int heightMeasureSpec)
37     {
38         // HACK! TAKE THAT ANDROID!
39         if (isExpanded())
40         {
41             // Calculate entire height by providing a very large height hint.
42             // View.MEASURED_SIZE_MASK represents the largest height possible.
43             int expandSpec = MeasureSpec.makeMeasureSpec(MEASURED_SIZE_MASK,
44                                                         MeasureSpec.AT_MOST);
45             super.onMeasure(widthMeasureSpec, expandSpec);
46
47             ViewGroup.LayoutParams params = getLayoutParams();
48             params.height = getMeasuredHeight();
49         }
50         else
51         {
52             super.onMeasure(widthMeasureSpec, heightMeasureSpec);
53         }
54     }
55
56     public void setExpanded(boolean expanded)
57     {
58         this.expanded = expanded;
59     }
60 }
61
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\java\com\mushoq\tinpoint\GenericPrizeCursorAdapter.java

```
1 package com.mushoq.tinpoint;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.net.Uri;
6 import android.os.Environment;
7 import android.view.LayoutInflater;
8 import android.view.View;
9 import android.view.ViewGroup;
10 import android.widget.CursorAdapter;
11 import android.widget.TextView;
12
13 import java.io.File;
14
15 import de.hdodenhof.circleimageview.CircleImageView;
16
17 /**
18  * Created by santiago on 2/25/2016.
19  */
20 public class GenericPrizeCursorAdapter extends CursorAdapter {
21
22
23     public GenericPrizeCursorAdapter(Context context, Cursor c, int flags) {
24         super(context, c, flags);
25     }
26
27     public View newView(Context context, Cursor cursor, ViewGroup parent) {
28         LayoutInflater inflater = (LayoutInflater) context.getSystemService(context.
29 LAYOUT_INFLATER_SERVICE);
30         View view = inflater.inflate(R.layout.row_layout_generic_prize, parent, false);
31         return view;
32     }
33     @Override
34     public void bindView(final View view, final Context context, final Cursor cursor) {
35         View v = view;
36
37         String photo = cursor.getString(cursor.getColumnIndex("icon"));
38         CircleImageView iv = (CircleImageView) v.findViewById(R.id.icon_photo);
39         iv.setImageResource(v.getResources().getIdentifier(photo, "drawable", context.getPackageName())
40 );
41     }
42 }
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\my\_spinner.xml

```
1 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
2   android:id="@android:id/text1"
3   style="@style/spinner_style"
4   android:singleLine="true"
5   android:layout_width="match_parent"
6   android:layout_height="wrap_content"
7   android:ellipsize="marquee"
8   android:textAlignment="inherit"/>
```



File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\give\_points.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical" android:layout_width="match_parent"
4   android:paddingLeft="15dp"
5   android:paddingRight="15dp"
6   android:layout_height="match_parent">
7
8
9
10 <ScrollView
11   android:layout_width="match_parent"
12   android:layout_height="fill_parent"
13   android:id="@+id/scrollView3" >
14   <com.mushoq.tinpoint.ExpandableHeightGridView
15     xmlns:android="http://schemas.android.com/apk/res/android"
16     android:id="@+id/pointsgridview"
17     style="@style/grid_icons"
18     android:numColumns="auto_fit"
19     android:gravity="center"
20     android:stretchMode="columnWidth"
21     android:layout_width="wrap_content"
22     android:layout_height="wrap_content"
23     android:isScrollContainer="false">
24
25   </com.mushoq.tinpoint.ExpandableHeightGridView>
26 </ScrollView>
27 </LinearLayout>
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\icon\_prizes.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical" android:layout_width="match_parent"
4   android:paddingLeft="15dp"
5   android:paddingRight="15dp"
6   android:layout_height="match_parent">
7
8
9
10 <ScrollView
11   android:layout_width="match_parent"
12   android:layout_height="fill_parent"
13   android:id="@+id/scrollView3" >
14   <com.mushoq.tinpoint.ExpandableHeightGridView
15     xmlns:android="http://schemas.android.com/apk/res/android"
16     android:id="@+id/iconsprizesgridview"
17     style="@style/grid_icons"
18     android:numColumns="auto_fit"
19     android:gravity="center"
20     android:stretchMode="columnWidth"
21     android:layout_width="wrap_content"
22     android:layout_height="wrap_content"
23     android:isScrollContainer="false">
24
25   </com.mushoq.tinpoint.ExpandableHeightGridView>
26 </ScrollView>
27 </LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context="com.mushoq.tinpoint.MainActivity"
7   style="@style/general_layout">
8
9   <TextView
10     android:layout_width="match_parent"
11     android:layout_height="wrap_content"
12     style="@style/green_title"
13     android:text="@string/TheChildren"
14     android:id="@+id/textView6" />
15
16   <ListView
17     android:layout_width="match_parent"
18     android:layout_height="wrap_content"
19     android:id="@+id/childList"
20     android:layout_below="@+id/textView6"
21     android:layout_above="@+id/layoutButton"
22     android:layout_alignParentLeft="true"
23     android:layout_alignParentStart="true"
24     android:divider="#00000000"/>
25
26
27   <LinearLayout
28     android:orientation="horizontal"
29     android:layout_width="match_parent"
30     android:layout_height="wrap_content"
31     android:id="@+id/layoutButton"
32     android:layout_above="@+id/addsFrameLayout"
33     android:gravity="center"
34     android:padding="5dp"
35     android:background="@color/transparentBlack">
36     <Button
37       android:layout_width="wrap_content"
38       style="@style/green_button"
39       android:text="@string/addChild"
40       android:id="@+id/newchild"
41     />
42   </LinearLayout>
43   <FrameLayout
44     android:id="@+id/addsFrameLayout"
45     android:layout_width="wrap_content"
46     android:layout_height="wrap_content"
47     android:layout_alignParentBottom="true"
48     android:layout_centerHorizontal="true"></FrameLayout>
49
50 </RelativeLayout>
51
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\activity\_splash.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  android:background="@drawable/splash_bg"
11  tools:context="com.mushoq.tinpoint.SplashActivity">
12  <ImageView
13    android:layout_width="wrap_content"
14    android:layout_height="wrap_content"
15    android:src="@drawable/logo"
16    android:id="@+id/logo"
17    android:layout_above="@+id/textView"
18    android:layout_centerHorizontal="true" />
19  <TextView
20    android:layout_width="wrap_content"
21    android:layout_height="wrap_content"
22    android:textAppearance="?android:attr/textAppearanceLarge"
23    android:text="TinPoint"
24    android:id="@+id/textView"
25    android:textSize="18dp"
26    android:textColor="@color/white"
27    android:layout_centerVertical="true"
28    android:layout_centerHorizontal="true" />
29
30
31 </RelativeLayout>
32
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="horizontal"
6   android:paddingBottom="5dp"
7   >
8
9   <LinearLayout
10    android:id="@+id/linearLayout0"
11    android:layout_width="wrap_content"
12    android:layout_height="wrap_content"
13    style="@style/row_child_linear_name"
14    android:orientation="vertical" >
15
16    <LinearLayout
17      android:orientation="vertical"
18      android:gravity="right"
19      style="@style/congrats_background"
20      android:layout_width="wrap_content"
21      android:layout_height="wrap_content"
22      android:id="@+id/childlayout">
23
24      <TextView
25        android:id="@+id/prizename"
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        style="@style/row_child_name"
29        android:text="Santiago">
30      </TextView>
31
32      <TextView
33        android:id="@+id/givePrize"
34        android:layout_width="wrap_content"
35        android:layout_height="wrap_content"
36        style="@style/row_child_age"
37        android:text="7 años">
38      </TextView>
39
40    </LinearLayout>
41  </LinearLayout>
42  <de.hdodenhof.circleimageview.CircleImageView
43    xmlns:app="http://schemas.android.com/apk/res-auto"
44    android:id="@+id/prizephoto"
45    style="@style/row_child_image"
46    android:src="@drawable/boy"
47    app:civ_border_width="2dp"
48    app:civ_border_color="#FFFFFF"/>
49
50
51
52 </RelativeLayout >
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\my\_spinner\_item.xml

```
1 <CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"  
2   android:id="@android:id/text1"  
3   style="@style/spinnerDropDownItemStyle"  
4   android:singleLine="true"  
5   android:layout_width="match_parent"  
6   android:layout_height="40dp"  
7   android:ellipsize="marquee" />
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="horizontal"
6     android:paddingBottom="5dp"
7     android:background="@drawable/green_child_bottom"
8 >
9
10 <LinearLayout
11     android:id="@+id/linearLayout0"
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     style="@style/row_child_linear_name"
15     android:orientation="vertical" >
16
17     <LinearLayout
18         android:orientation="vertical"
19         android:gravity="right"
20         style="@style/child_background"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:id="@+id/childlayout">
24
25         <TextView
26             android:id="@+id/name"
27             android:layout_width="wrap_content"
28             android:layout_height="wrap_content"
29             style="@style/row_child_name"
30             android:text="Santiago">
31         </TextView>
32
33         <TextView
34             android:id="@+id/age"
35             android:layout_width="wrap_content"
36             android:layout_height="wrap_content"
37             style="@style/row_child_age"
38             android:text="7 años">
39         </TextView>
40
41     </LinearLayout>
42 </LinearLayout>
43 <de.hdodenhof.circleimageview.CircleImageView
44     xmlns:app="http://schemas.android.com/apk/res-auto"
45     android:id="@+id/photo"
46     style="@style/row_child_image"
47     android:src="@drawable/boy"
48     app:civ_border_width="2dp"
49     app:civ_border_color="#FFFFFF"/>
50
51 <LinearLayout
52     android:orientation="vertical"
53     android:gravity="right"
54     android:layout_width="wrap_content"
55     android:layout_height="wrap_content"
56     android:layout_alignParentRight="true"
57     android:layout_alignParentEnd="true"
58     android:layout_alignParentBottom="true"
59 >
60 <TextView
61     android:layout_gravity="right"
62     android:textColor="@color/white"
63     android:text="800"
64     style="@style/circle_points_on_list"
65     android:id="@+id/points"
66 />
67
68 <TextView
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\row\_layout\_child.xml

```
69     style="@style/row_child_points"
70     android:text="@string/points"
71     android:id="@+id/textPoints"
72     />
73 </LinearLayout>
74
75 </RelativeLayout >
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   android:gravity="center"
7   android:padding="5dp"
8   android:id="@+id/layoutPrize">
9
10  <de.hdodenhof.circleimageview.CircleImageView
11    xmlns:app="http://schemas.android.com/apk/res-auto"
12    android:id="@+id/prizephoto"
13    style="@style/grid_prizes"
14    android:src="@drawable/ic_gift"
15    app:civ_border_width="1dp"
16    app:civ_border_color="#FFFFFF"/>
17
18  <LinearLayout
19    android:orientation="vertical"
20    android:layout_width="match_parent"
21    android:gravity="center"
22    android:layout_height="wrap_content">
23    <TextView
24      android:id="@+id/prizename"
25      android:layout_width="wrap_content"
26      android:layout_height="wrap_content"
27      style="@style/row_child_age"
28      android:gravity="center"
29      android:text="Regalo">
30    </TextView>
31    <TextView
32      android:id="@+id/prizepoints"
33      android:layout_width="wrap_content"
34      android:layout_height="wrap_content"
35      style="@style/row_prize_points"
36      android:gravity="center"
37      android:text="10 puntos">
38    </TextView>
39  </LinearLayout>
40
41 </LinearLayout>
42
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\row\_layout\_points.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical" android:layout_width="match_parent"
4   android:gravity="center"
5   android:layout_height="match_parent">
6
7   <TextView
8     android:layout_gravity="center"
9     android:textColor="@color/white"
10    android:text="800"
11    style="@style/circle_points_to_give"
12    android:id="@+id/givepoints"
13  />
14 </LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:id="@+id/parent"
7   tools:context="com.mushoq.tinpoint.NewAwardActivity">
8   <TextView
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    style="@style/green_title"
12    android:text="@string/assignPrizes"
13    android:id="@+id/greentitle"
14    android:layout_alignParentTop="true"
15    android:layout_alignParentLeft="true"
16    android:layout_alignParentStart="true" />
17
18  <ScrollView
19    android:layout_width="match_parent"
20    android:layout_height="fill_parent"
21    android:id="@+id/scrollView"
22    android:layout_below="@+id/greentitle"
23    android:layout_alignParentLeft="true"
24    android:layout_alignParentStart="true"
25    android:layout_above="@+id/addsFrameLayout">
26
27    <LinearLayout
28      android:orientation="vertical"
29      android:layout_width="fill_parent"
30      android:layout_height="wrap_content"
31      android:focusable="true"
32      android:focusableInTouchMode="true"
33      android:layout_margin="@dimen/activity_horizontal_margin"
34    >
35
36      <LinearLayout
37        android:orientation="horizontal"
38        android:layout_width="match_parent"
39        android:layout_height="match_parent"
40        android:id="@+id/layoutGender"
41        style="@style/blue_name_title">
42        <TextView
43          android:layout_width="wrap_content"
44          android:layout_height="wrap_content"
45          style="@style/row_child_age"
46          android:text="@string/NewPrizeFor"
47          android:layout_marginRight="5dp"
48          android:id="@+id/prizefor"
49        />
50        <TextView
51          android:layout_width="wrap_content"
52          android:layout_height="wrap_content"
53          style="@style/row_child_name"
54          android:text="Martín"
55          android:id="@+id/name" />
56
57      </LinearLayout>
58    <TextView
59      android:layout_width="wrap_content"
60      android:layout_height="wrap_content"
61      style="@style/row_child_age"
62      android:text="@string/name" />
63
64    <EditText
65      android:layout_width="match_parent"
66      android:layout_height="wrap_content"
67      android:inputType="text"
68
```

```
69     style="@style/input_text"
70     android:id="@+id/prizename"
71     android:singleLine="true"/>
72 <TextView
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     style="@style/row_child_age"
76     android:text="@string/points" />
77
78 <EditText
79     android:layout_width="match_parent"
80     android:layout_height="wrap_content"
81     android:inputType="number"
82     android:numeric="integer"
83     android:ems="10"
84     style="@style/input_text"
85     android:id="@+id/prizepoints"
86     android:singleLine="true"/>
87
88 <CheckBox
89     android:layout_width="wrap_content"
90     android:layout_height="wrap_content"
91     android:ellipsize="marquee"
92     android:text="@string/AssignPrizeForAllProfiles"
93     style="@style/row_child_age"
94     android:id="@+id/prizeForAll" />
95
96 <LinearLayout
97     android:orientation="horizontal"
98     android:layout_width="match_parent"
99     android:layout_height="match_parent"
100    android:gravity="center"
101    android:id="@+id/background_image_linear_layout">
102
103    <Button
104        android:layout_width="wrap_content"
105
106        android:text="@string/TakePhoto"
107        style="@style/orange_button"
108        android:id="@+id/takephoto" />
109
110    <Button
111        android:layout_width="wrap_content"
112
113        android:text="@string/GalleryPicture"
114        style="@style/orange_button"
115        android:id="@+id/galleryphoto" />
116    <Button
117        android:layout_width="wrap_content"
118
119        android:text="@string/icons"
120        style="@style/orange_button"
121        android:id="@+id/icons" />
122 </LinearLayout>
123 <de.hdodenhof.circleimageview.CircleImageView
124     xmlns:app="http://schemas.android.com/apk/res-auto"
125     android:id="@+id/photo"
126     style="@style/row_child_image"
127     android:layout_gravity="center_horizontal"
128     android:src="@drawable/ic_gift"
129     app:civ_border_width="2dp"
130     app:civ_border_color="#FFFFFF"/>
131
132 <Button
133     android:layout_width="wrap_content"
134     android:layout_gravity="center_horizontal"
135     style="@style/green_button"
136     android:text="@string/save"
```

File - D:\work\pinpoint\pinpointAndroid\app\src\main\res\layout\activity\_new\_award.xml

```
137         android:id="@+id/save" />
138
139     </LinearLayout>
140 </ScrollView>
141 <FrameLayout
142     android:id="@+id/addsFrameLayout"
143     android:layout_width="wrap_content"
144     android:layout_height="wrap_content"
145     android:layout_alignParentBottom="true"
146     android:layout_centerHorizontal="true"></FrameLayout>
147 </RelativeLayout>
148
149
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:id="@+id/parent"
7   tools:context="com.mushoq.pinpoint.NewChildActivity">
8
9   <ScrollView
10    android:layout_width="fill_parent"
11    android:layout_height="fill_parent"
12    android:id="@+id/scrollView2"
13    android:layout_alignParentTop="true"
14    android:layout_alignParentLeft="true"
15    android:layout_alignParentStart="true"
16    android:layout_above="@+id/addsFrameLayout">
17
18    <LinearLayout
19      android:orientation="vertical"
20      android:layout_width="fill_parent"
21      android:layout_height="wrap_content"
22    >
23      <TextView
24        android:layout_width="match_parent"
25        android:layout_height="wrap_content"
26        style="@style/green_title"
27        android:text="@string/Enterallfields"
28        android:id="@+id/greentitle" />
29      <LinearLayout
30        android:orientation="vertical"
31        android:layout_width="fill_parent"
32        android:layout_height="wrap_content"
33        android:focusable="true"
34        android:focusableInTouchMode="true"
35        android:layout_margin="@dimen/activity_horizontal_margin">
36
37        <TextView
38          android:layout_width="wrap_content"
39          android:layout_height="wrap_content"
40          style="@style/row_child_age"
41          android:textStyle="bold"
42          android:text="@string/name"
43          android:id="@+id/nom" />
44
45        <EditText
46          android:layout_width="match_parent"
47          android:layout_height="wrap_content"
48          android:inputType="textPersonName"
49          style="@style/input_text"
50          android:id="@+id/name"
51          android:singleLine="true"/>
52        <TextView
53          android:layout_width="wrap_content"
54          android:layout_height="wrap_content"
55          style="@style/row_child_age"
56          android:textStyle="bold"
57          android:text="@string/Birthdate"
58          android:id="@+id/textView4" />
59
60        <LinearLayout
61          android:orientation="horizontal"
62          android:layout_width="match_parent"
63          android:layout_height="match_parent">
64
65          <Button
66            android:id="@+id/getdate"
67            android:layout_width="wrap_content"
68
```

```
69         style="@style/orange_button"
70         android:onClick="setDate"
71         android:text="@string/setDate" />
72     <TextView
73         android:id="@+id/textView3"
74         style="@style/row_child_name"
75         android:layout_width="fill_parent"
76         android:layout_height="fill_parent"
77         android:gravity="center"
78         android:text=""
79         android:textAppearance="?android:attr/textAppearanceMedium" />
80
81 </LinearLayout>
82
83 <TextView
84     android:layout_width="wrap_content"
85     android:layout_height="wrap_content"
86     android:text="@string/Gender"
87     style="@style/row_child_age"
88     android:textStyle="bold"
89     android:id="@+id/textView5" />
90
91 <Spinner
92     android:layout_width="match_parent"
93     android:layout_height="wrap_content"
94     android:background="@color/transparent"
95     android:id="@+id/gender" />
96
97 <LinearLayout
98     android:orientation="horizontal"
99     android:layout_width="match_parent"
100    android:layout_height="match_parent"
101    android:gravity="center"
102    android:id="@+id/background_image_linear_layout">
103
104    <Button
105        android:layout_width="wrap_content"
106        android:text="@string/TakePhoto"
107        style="@style/orange_button"
108        android:id="@+id/takephoto" />
109
110    <Button
111        android:layout_width="wrap_content"
112        android:text="@string/GalleryPicture"
113        style="@style/orange_button"
114        android:id="@+id/galleryphoto" />
115 </LinearLayout>
116 <de.hdodenhof.circleimageview.CircleImageView
117     xmlns:app="http://schemas.android.com/apk/res-auto"
118     android:id="@+id/photo"
119     style="@style/open_child_image"
120     android:layout_gravity="center_horizontal"
121
122     android:layout_margin="10dp"
123     android:src="@drawable/boy"
124     app:civ_border_width="2dp"
125     app:civ_border_color="#FFFFFF"/>
126
127 <Button
128     android:layout_width="wrap_content"
129     android:layout_gravity="center_horizontal"
130     style="@style/green_button"
131     android:text="@string/save"
132     android:id="@+id/save" />
133
134 </LinearLayout>
135 </LinearLayout>
136 </ScrollView>
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\activity\_new\_child.xml

```
137 <FrameLayout
138     android:id="@+id/addsFrameLayout"
139     android:layout_width="wrap_content"
140     android:layout_height="wrap_content"
141     android:layout_alignParentBottom="true"
142     android:layout_centerHorizontal="true"></FrameLayout>
143 </RelativeLayout>
144
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:id="@+id/container"
7   tools:context="com.mushoq.tinpoint.OpenChildActivity">
8   <TextView
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    style="@style/green_title"
12    android:text="@string/pointsAndPrizes"
13    android:id="@+id/greentitle"
14    android:layout_alignParentTop="true"
15    android:layout_alignParentLeft="true"
16    android:layout_alignParentStart="true" />
17
18   <ScrollView
19     android:layout_width="match_parent"
20     android:layout_height="fill_parent"
21     android:id="@+id/scrollView"
22     android:layout_below="@+id/greentitle"
23     android:layout_alignParentLeft="true"
24     android:layout_alignParentStart="true"
25     android:layout_above="@+id/addsFrameLayout">
26
27
28     <LinearLayout
29       android:orientation="vertical"
30       android:layout_width="fill_parent"
31       android:layout_height="wrap_content"
32       android:layout_margin="@dimen/activity_horizontal_margin"
33     >
34
35
36       <LinearLayout
37         android:orientation="horizontal"
38         android:layout_width="match_parent"
39         android:layout_height="match_parent"
40         android:id="@+id/layoutGender"
41         style="@style/blue_name_title">
42
43         <TextView
44           android:layout_width="wrap_content"
45           android:layout_height="wrap_content"
46           style="@style/open_child_name"
47           android:text="Martín"
48           android:id="@+id/name" />
49
50         <TextView
51           android:layout_width="wrap_content"
52           android:layout_height="wrap_content"
53
54           android:text="Edad: 5 años"
55           style="@style/row_child_age"
56           android:layout_marginLeft="10dp"
57           android:id="@+id/age" />
58       </LinearLayout>
59     <LinearLayout
60       android:orientation="horizontal"
61       android:layout_width="match_parent"
62       android:layout_height="match_parent"
63     >
64     <de.hdodenhof.circleimageview.CircleImageView
65       xmlns:app="http://schemas.android.com/apk/res-auto"
66       android:id="@+id/photo"
67       style="@style/open_child_image"
68     >
```

```
69         android:src="@drawable/boy"
70         app:civ_border_width="2dp"
71         app:civ_border_color="#FFFFFF"/>
72     <LinearLayout
73         android:orientation="vertical"
74         android:layout_width="match_parent"
75         android:layout_height="wrap_content"
76         android:gravity="right"
77     >
78
79         <Button
80             android:text="@string/editProfile"
81             style="@style/green_button_small"
82             android:id="@+id/editProfile" />
83
84         <TextView
85             android:layout_width="wrap_content"
86             android:layout_height="wrap_content"
87             android:text="@string/lifetimePoints"
88             style="@style/row_child_name"
89             android:id="@+id/textView2"
90         />
91
92         <TextView
93             android:textColor="@color/white"
94             android:text="80"
95             style="@style/circle_points_on_list"
96             android:layout_marginLeft="10dp"
97             android:id="@+id/points"
98         />
99
100     </LinearLayout>
101
102 </LinearLayout>
103 <TextView
104     android:layout_gravity="center_horizontal"
105     android:layout_width="wrap_content"
106     android:layout_height="wrap_content"
107     android:text="@string/prizesAvailable"
108     style="@style/row_child_name"
109     android:id="@+id/availablePrizes"
110 />
111 <com.mushoq.tinpoint.ExpandableHeightGridView
112     xmlns:android="http://schemas.android.com/apk/res/android"
113     android:id="@+id/listPrizes"
114     style="@style/prizes_grid"
115     android:numColumns="auto_fit"
116     android:gravity="center"
117     android:stretchMode="columnWidth"
118     android:layout_width="wrap_content"
119     android:layout_height="wrap_content"
120     android:isScrollContainer="false">
121
122 </com.mushoq.tinpoint.ExpandableHeightGridView>
123 <TextView
124     android:layout_gravity="center_horizontal"
125     android:layout_width="wrap_content"
126     android:layout_height="wrap_content"
127     style="@style/row_child_name"
128     android:text="@string/unassignePrizes"
129     android:id="@+id/notprizes"
130 />
131 <LinearLayout
132     android:orientation="horizontal"
133     android:layout_width="match_parent"
134     android:layout_height="wrap_content"
135     android:layout_marginBottom="10dp"
136     android:gravity="center"
```

```
137     >
138     <Button
139         android:text="@string/scoreapoint"
140         style="@style/orange_button"
141         android:id="@+id/upPoint" />
142     <Button
143         android:text="@string/assignPrizes"
144         style="@style/green_button_small"
145         android:id="@+id/setAward" />
146 </LinearLayout>
147 <TextView
148     android:layout_gravity="center_horizontal"
149     android:layout_width="wrap_content"
150     android:layout_height="wrap_content"
151     android:text="@string/prizesawarded"
152     style="@style/row_child_age"
153     android:id="@+id/textprizesawarded"
154 />
155 <com.mushoq.tinpoint.ExpandableHeightGridView
156     xmlns:android="http://schemas.android.com/apk/res/android"
157     android:id="@+id/prizesawarded"
158     android:numColumns="auto_fit"
159     android:gravity="center"
160     style="@style/prizes_given_grid"
161     android:stretchMode="columnWidth"
162     android:layout_width="wrap_content"
163     android:layout_height="wrap_content"
164     android:layout_marginBottom="20dp"
165     android:isScrollContainer="false">
166
167     </com.mushoq.tinpoint.ExpandableHeightGridView>
168 </LinearLayout>
169 </ScrollView>
170
171 <de.hdodenhof.circleimageview.CircleImageView
172     xmlns:app="http://schemas.android.com/apk/res-auto"
173     android:id="@+id/expanded_image"
174     style="@style/expanded_image"
175     android:visibility="invisible"
176     app:civ_border_width="2dp"
177     app:civ_border_color="#FFFFFF"/>
178 <FrameLayout
179     android:id="@+id/addsFrameLayout"
180     android:layout_width="wrap_content"
181     android:layout_height="wrap_content"
182     android:layout_alignParentBottom="true"
183     android:layout_centerHorizontal="true"></FrameLayout>
184 </RelativeLayout>
185
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\activity\_security\_pin.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  tools:context="com.mushoq.tinpoint.SecurityPin">
11
12 </RelativeLayout>
13
```

File - D:\work\inpoint\inpointAndroid\app\src\main\res\layout\row\_layout\_prize\_given.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   android:gravity="center"
7   android:padding="5dp"
8   android:id="@+id/layoutPrize">
9
10  <de.hdodenhof.circleimageview.CircleImageView
11    xmlns:app="http://schemas.android.com/apk/res-auto"
12    android:id="@+id/prizephoto"
13    style="@style/grid_given_prizes"
14    android:src="@drawable/ic_gift"
15    app:civ_border_width="1dp"
16    app:civ_border_color="#FFFFFF"/>
17
18  <LinearLayout
19    android:orientation="vertical"
20    android:layout_width="match_parent"
21    android:gravity="center"
22    android:layout_height="wrap_content">
23    <TextView
24      android:id="@+id/prizename"
25      android:layout_width="wrap_content"
26      android:layout_height="wrap_content"
27      android:gravity="center"
28      android:text="Regalo"
29      android:textColor="@color/white"
30      android:textSize="11sp" >
31    </TextView>
32  </LinearLayout>
33
34 </LinearLayout>
35
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\layout\row\_layout\_generic\_prize.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical" android:layout_width="match_parent"
4   android:gravity="center"
5   android:layout_height="match_parent">
6
7   <de.hdodenhof.circleimageview.CircleImageView
8     xmlns:app="http://schemas.android.com/apk/res-auto"
9     android:id="@+id/icon_photo"
10    style="@style/icon_prizes"
11    android:src="@drawable/i1"
12    app:civ_border_width="1dp"
13    app:civ_border_color="#FFFFFF"/>
14 </LinearLayout>
```

File - D:\work\pinpoint\pinpointAndroid\app\src\main\res\values\attrs.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4 </resources>
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\values\colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#356679</color>
4   <color name="colorPrimaryDark">#222222</color>
5   <color name="colorAccent">#FF4081</color>
6   <color name="white">#FFFFFF</color>
7   <color name="yellow">#FFFF00</color>
8   <color name="transparent">#00000000</color>
9   <color name="transparentBlack">#33000000</color>
10  <color name="pink">#D962DA</color>
11  <color name="blue">#358EA8</color>
12  <color name="pin_default_item_background">#356679</color>
13  <color name="pin_default_item_text">#FFFFFF</color>
14 </resources>
15
```



File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\values\dimens.xml

```
1 <resources>
2   <!-- Default screen margins, per the Android Design guidelines. -->
3   <dimen name="activity_horizontal_margin">16dp</dimen>
4   <dimen name="activity_vertical_margin">16dp</dimen>
5   <dimen name="fab_margin">16dp</dimen>
6 </resources>
7
```

```

1 <resources>
2
3 <!-- Base application theme. -->
4 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5 <!-- Customize your theme here. -->
6 <item name="colorPrimary">@color/colorPrimary</item>
7 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8 <item name="colorAccent">@color/colorAccent</item>
9 <item name="android:actionBarStyle">@style/ActionBarStyle</item>
10 <item name="actionBarStyle">@style/ActionBarStyle</item>
11 <item name="android:windowActionBar">>true</item>
12 <item name="android:windowBackground">@drawable/background</item>
13 </style>
14
15 <style name="AppThemeCrop" parent="Theme.AppCompat.Light.DarkActionBar">
16 <!-- Customize your theme here. -->
17 <item name="colorPrimary">@color/colorPrimary</item>
18 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
19 <item name="colorAccent">@color/colorAccent</item>
20 <item name="windowNoTitle">>true</item>
21 <item name="windowActionBar">>false</item>
22 <item name="android:windowFullscreen">>true</item>
23 <item name="android:windowContentOverlay">@null</item>
24 </style>
25 <style name="Pin__AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
26 <!-- Customize your theme here. -->
27 <item name="colorPrimary">@color/colorPrimary</item>
28 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
29 <item name="colorAccent">@color/colorAccent</item>
30 <item name="android:actionBarStyle">@style/ActionBarStyle</item>
31 <item name="actionBarStyle">@style/ActionBarStyle</item>
32 <item name="android:windowActionBar">>true</item>
33 <item name="android:windowBackground">@drawable/background</item>
34 </style>
35 <style name="ActionBarStyle" parent="Widget.AppCompat.ActionBar">
36 <item name="android:background">@drawable/header_border</item>
37 <item name="background">@drawable/header_border</item>
38 </style>
39
40 <style name="spinner_style">
41 <item name="android:paddingLeft">10dp</item>
42 <item name="android:paddingTop">3dp</item>
43 <item name="android:paddingBottom">2dp</item>
44 <item name="android:dividerHeight">1dp</item>
45 <item name="android:textSize">17dp</item>
46
47 <item name="android:background">@drawable/campo_blanco_rounded</item>
48 <item name="android:drawableRight">@drawable/ic_spinner</item>
49 <item name="android:scrollbarAlwaysDrawVerticalTrack">>true</item>
50 <item name="android:dropDownSelector">@android:color/white</item>
51 </style>
52
53 <style name="spinnerDropDownItemStyle" parent="android:TextAppearance.Widget.TextView.
SpinnerItem">
54 <item name="android:textSize">17dp</item>
55 <item name="android:paddingLeft">6dp</item>
56 <item name="android:paddingTop">2dp</item>
57 <item name="android:background">@drawable/border_bottom_dashed_whitebg</item>
58 </style>
59
60 <style name="general_layout">
61 <item name="android:background">@drawable/border_top</item>
62 </style>
63
64 <style name="green_title">
65 <item name="android:textColor">@color/white</item>
66 <item name="android:background">@drawable/green_title</item>
67 <item name="android:layout_width">match_parent</item>

```

```
68     <item name="android:paddingLeft">10dp</item>
69     <item name="android:paddingTop">3dp</item>
70     <item name="android:paddingBottom">3dp</item>
71     <item name="android:textSize">15dp</item>
72 </style>
73 <style name="row_green_child_bottom">
74     <item name="android:textColor">@color/white</item>
75     <item name="android:background">@drawable/green_child_bottom</item>
76     <item name="android:layout_width">match_parent</item>
77     <item name="android:paddingLeft">10dp</item>
78     <item name="android:paddingTop">3dp</item>
79     <item name="android:paddingBottom">3dp</item>
80     <item name="android:textSize">12dp</item>
81 </style>
82 <style name="blue_name_title">
83     <item name="android:layout_marginBottom">12dp</item>
84     <item name="android:background">@drawable/blue_name_title</item>
85 </style>
86 <style name="green_button">
87     <item name="android:layout_width">wrap_content</item>
88     <item name="android:layout_height">35dp</item>
89     <item name="android:background">@drawable/button_green_custom</item>
90     <item name="android:textColor">@color/white</item>
91     <item name="android:paddingLeft">6dp</item>
92     <item name="android:paddingRight">6dp</item>
93     <item name="android:paddingTop">1dp</item>
94     <item name="android:paddingBottom">1dp</item>
95     <item name="android:textSize">16dp</item>
96     <item name="android:capitalize">none</item>
97     <item name="android:layout_margin">5dp</item>
98     <item name="android:textAllCaps">>false</item>
99 </style>
100
101 <style name="green_button_small">
102     <item name="android:layout_width">wrap_content</item>
103     <item name="android:layout_height">35dp</item>
104     <item name="android:background">@drawable/button_green_custom</item>
105     <item name="android:textColor">@color/white</item>
106     <item name="android:paddingLeft">6dp</item>
107     <item name="android:paddingRight">6dp</item>
108     <item name="android:paddingTop">1dp</item>
109     <item name="android:paddingBottom">1dp</item>
110     <item name="android:textSize">15dp</item>
111     <item name="android:capitalize">none</item>
112     <item name="android:layout_margin">3dp</item>
113     <item name="android:textAllCaps">>false</item>
114 </style>
115
116 <style name="orange_button">
117     <item name="android:background">@drawable/button_orange_custom</item>
118     <item name="android:layout_height">35dp</item>
119     <item name="android:layout_width">wrap_content</item>
120     <item name="android:textColor">@color/white</item>
121     <item name="android:paddingLeft">6dp</item>
122     <item name="android:paddingRight">6dp</item>
123     <item name="android:paddingTop">1dp</item>
124     <item name="android:paddingBottom">1dp</item>
125     <item name="android:textSize">15dp</item>
126     <item name="android:capitalize">none</item>
127     <item name="android:layout_margin">3dp</item>
128     <item name="android:textAllCaps">>false</item>
129 </style>
130
131 <style name="input_text">
132     <item name="android:background">@drawable/campo_blanco_rounded</item>
133     <item name="android:layout_height">20dp</item>
134     <item name="android:layout_marginLeft">5dp</item>
135     <item name="android:padding">8dp</item>
```

```

136     <item name="android:textSize">15sp</item>
137     <item name="android:layout_marginBottom">4dp</item>
138 </style>
139
140 <style name="circle_points">
141     <item name="android:background">@drawable/circle_points</item>
142     <item name="android:textColor">@color/white</item>
143     <item name="android:textStyle">bold</item>
144     <item name="android:textSize">33sp</item>
145     <item name="android:layout_width">70dp</item>
146     <item name="android:layout_height">70dp</item>
147     <item name="android:gravity">center</item>
148     <item name="android:layout_marginTop">4dp</item>
149 </style>
150
151 <style name="AppTheme.NoActionBar">
152     <item name="windowActionBar">false</item>
153     <item name="windowNoTitle">>true</item>
154 </style>
155
156 <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar"
157 />
158 <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
159 <!-- Row Layout Child -->
160 <style name="row_child_linear_name">
161     <item name="android:layout_marginTop">10dp</item>
162     <item name="android:layout_marginLeft">50dp</item>
163 </style>
164 <style name="child_background">
165     <item name="android:background">@drawable/blue_title</item>
166 </style>
167 <style name="congrats_background">
168     <item name="android:background">@drawable/green_price_title</item>
169 </style>
170 <style name="row_child_name">
171     <item name="android:textStyle">bold</item>
172     <item name="android:textColor">@color/white</item>
173     <item name="android:textSize">14sp</item>
174 </style>
175 <style name="row_child_age">
176     <item name="android:textStyle">normal</item>
177     <item name="android:textColor">@color/white</item>
178     <item name="android:textSize">12sp</item>
179 </style>
180 <style name="row_prize_points">
181     <item name="android:textStyle">bold</item>
182     <item name="android:textColor">@color/yellow</item>
183     <item name="android:textSize">11sp</item>
184 </style>
185 <style name="row_child_image">
186     <item name="android:layout_width">85dp</item>
187     <item name="android:layout_height">85dp</item>
188     <item name="android:layout_margin">5dp</item>
189 </style>
190 <style name="open_child_name">
191     <item name="android:textStyle">bold</item>
192     <item name="android:textColor">@color/white</item>
193     <item name="android:textSize">16sp</item>
194 </style>
195 <style name="open_child_image">
196     <item name="android:layout_width">100dp</item>
197     <item name="android:layout_height">100dp</item>
198     <item name="android:layout_marginLeft">5dp</item>
199     <item name="android:layout_gravity">left</item>
200 </style>
201 <style name="circle_points_on_list">
202     <item name="android:background">@drawable/circle_points</item>

```

```
203     <item name="android:textColor">@color/white</item>
204     <item name="android:textStyle">bold</item>
205     <item name="android:textSize">20sp</item>
206     <item name="android:layout_width">50dp</item>
207     <item name="android:layout_height">50dp</item>
208     <item name="android:gravity">center</item>
209     <item name="android:layout_marginRight">5dp</item>
210 </style>
211 <style name="circle_points_to_give">
212     <item name="android:background">@drawable/circle_points</item>
213     <item name="android:textColor">@color/white</item>
214     <item name="android:textStyle">bold</item>
215     <item name="android:textSize">15sp</item>
216     <item name="android:layout_width">30dp</item>
217     <item name="android:layout_height">30dp</item>
218     <item name="android:gravity">center</item>
219     <item name="android:layout_margin">10dp</item>
220 </style>
221 <style name="row_child_points">
222     <item name="android:textColor">@color/white</item>
223     <item name="android:textStyle">normal</item>
224     <item name="android:textSize">11sp</item>
225     <item name="android:layout_width">50dp</item>
226     <item name="android:layout_height">wrap_content</item>
227     <item name="android:gravity">center</item>
228     <item name="android:layout_marginRight">5dp</item>
229 </style>
230 <style name="prizes_grid">
231     <item name="android:columnWidth">60dp</item>
232 </style>
233 <style name="prizes_given_grid">
234     <item name="android:columnWidth">50dp</item>
235 </style>
236 <style name="icon_prizes">
237     <item name="android:layout_width">50dp</item>
238     <item name="android:layout_height">50dp</item>
239     <item name="android:layout_margin">10dp</item>
240     <item name="android:layout_gravity">center_horizontal</item>
241 </style>
242 <style name="grid_icons">
243     <item name="android:layout_marginBottom">20dp</item>
244     <item name="android:columnWidth">60dp</item>
245 </style>
246 <style name="grid_prizes">
247     <item name="android:layout_width">50dp</item>
248     <item name="android:layout_height">50dp</item>
249     <item name="android:layout_marginBottom">2dp</item>
250     <item name="android:layout_gravity">center_horizontal</item>
251 </style>
252 <style name="grid_given_prizes">
253     <item name="android:layout_width">35dp</item>
254     <item name="android:layout_height">35dp</item>
255     <item name="android:layout_marginBottom">1dp</item>
256 </style>
257 <style name="expanded_image">
258     <item name="android:layout_width">240dp</item>
259     <item name="android:layout_height">240dp</item>
260 </style>
261 </resources>
262
```



```
1 <resources>
2 <string name="banner_ad_unit_id">ca-app-pub-2861342412299558/3271483429</string>
3 <string name="app_name">TinPoint</string>
4 <string name="TheChildren">Children</string>
5 <string name="male">male</string>
6 <string name="female">female</string>
7 <string name="age">age</string>
8 <string name="yearsold">years old</string>
9 <string name="Birthdate">Birthdate</string>
10 <string name="Gender">Gender</string>
11 <string name="setDate">Select date</string>
12 <string name="changeDate">Change date</string>
13 <string name="allFields">Enter all fields</string>
14 <string name="intNumberOnly">Points only integers</string>
15 <string name="pointsAndPrizes">Points and Prizes</string>
16 <string name="lifetimePoints">Accumulated points</string>
17 <string name="prizesAvailable">Prizes available</string>
18 <string name="scoreapoint">Score points</string>
19 <string name="unassignPrizes">No added prizes</string>
20 <string name="prizesawarded">Prizes awarded</string>
21 <string name="no">No</string>
22 <string name="yes">Yes</string>
23 <string name="Congratulations">Congratulations!</string>
24 <string name="GivenPrize">Given prize</string>
25 <string name="YouHaveGivenThisPrize">You have given this prize</string>
26 <string name="giveapize">Give a prize</string>
27 <string name="areYouSurePrize">Are you sure to submit this prize?</string>
28 <string name="arwYouSureDeletePrize">Are you sure to delete this prize?</string>
29 <string name="arwYouSureDeleteProfile">Are you sure to delete this profile?</string>
30 <string name="areYouSurePoint">Are you sure to score points?</string>
31 <string name="editProfile">Edit profile</string>
32 <string name="Enterallfields">Enter all fields</string>
33 <string name="Editfields">Edit fields</string>
34 <string name="assignPrizes">Add prize</string>
35 <string name="NewPrizeFor">New prize for </string>
36 <string name="TakePhoto">Take photo</string>
37 <string name="GalleryPicture">Photo gallery</string>
38 <string name="save">Save</string>
39 <string name="addChild">Add child</string>
40 <string name="points">Points</string>
41 <string name="name">Name</string>
42 <string name="remove">Remove</string>
43 <string name="edit">Edit</string>
44 <string name="EditPrizeFor">Edit prize for</string>
45 <string name="AssignPrizeForAllProfiles">Assign prize for all profiles</string>
46 <string name="search">Search</string>
47 <string name="icons">Icons</string>
48 <string name="howManyPoints">How many points?</string>
49 <string name="replaceWithPhoto">Replace with photo</string>
50 <string name="replaceWithGallery">Replace with gallery</string>
51 <string name="replaceWithIcon">Replace with icon</string>
52 <string name="ContinuesToAddPointsToRedeemThisPrize">Continues to add points to redeem
this prize</string>
53 <string name="permission">These permissions are required to use TinPoint</string>
54 <string name="cancel">Cancel</string>
55 <string name="accesto">You need to grant access to</string>
56
57 <string name="pin__description_create">Enter a %1$s digit code to lock %2$s</string>
58 <string name="pin__description_confirm">Re-enter your PIN to confirm</string>
59 <string name="pin__description_unlock">%1$s locked! \n Enter PIN to unlock</string>
60
61 <string name="pin__unlock_error_no_matching_pin_found">There is no PIN to unlock!</string>
62 <string name="pin__unlock_error_match_failed">The PIN code does not match!</string>
63 <string name="pin__unlock_error_insufficient_selection">You have not selected the required
number of digits! Please try again</string>
64 <string name="pin__unlock_error_retry_limit_exceeded">You have failed too many times. You can
try again in %1$s.</string>
65
```

```
66 <string name="pin_toast_lock_success">%1$s locked!</string>
67 <string name="pin_toast_unlock_success">%1$s unlocked correctly!</string>
68 <string name="pin_toast_unlock_required">You need to unlock to do that!</string>
69
70 <string-array name="icons_array">
71 <item>i1</item>
72 <item>i2</item>
73 <item>i3</item>
74 <item>i4</item>
75 <item>i5</item>
76 <item>i6</item>
77 <item>i7</item>
78 <item>i8</item>
79 <item>i9</item>
80 <item>i10</item>
81 <item>i11</item>
82 <item>i12</item>
83 <item>i13</item>
84 <item>i14</item>
85 <item>i15</item>
86 <item>i16</item>
87 <item>i17</item>
88 <item>i18</item>
89 <item>i19</item>
90 <item>i20</item>
91 </string-array>
92 <string-array name="points_array">
93 <item>1</item>
94 <item>2</item>
95 <item>3</item>
96 <item>4</item>
97 <item>5</item>
98 <item>6</item>
99 <item>7</item>
100 <item>8</item>
101 </string-array>
102 </resources>
103
```

File - D:\work\pinpoint\pinpointAndroid\app\src\main\res\values\attrs.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4 </resources>
```



File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\values\colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#356679</color>
4   <color name="colorPrimaryDark">#222222</color>
5   <color name="colorAccent">#FF4081</color>
6   <color name="white">#FFFFFF</color>
7   <color name="yellow">#FFFF00</color>
8   <color name="transparent">#00000000</color>
9   <color name="transparentBlack">#33000000</color>
10  <color name="pink">#D962DA</color>
11  <color name="blue">#358EA8</color>
12  <color name="pin_default_item_background">#356679</color>
13  <color name="pin_default_item_text">#FFFFFF</color>
14 </resources>
15
```

File - D:\work\tinpoint\tinpointAndroid\app\src\main\res\values\dimens.xml

```
1 <resources>
2   <!-- Default screen margins, per the Android Design guidelines. -->
3   <dimen name="activity_horizontal_margin">16dp</dimen>
4   <dimen name="activity_vertical_margin">16dp</dimen>
5   <dimen name="fab_margin">16dp</dimen>
6 </resources>
7
```

```

1 <resources>
2
3 <!-- Base application theme. -->
4 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5 <!-- Customize your theme here. -->
6 <item name="colorPrimary">@color/colorPrimary</item>
7 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8 <item name="colorAccent">@color/colorAccent</item>
9 <item name="android:actionBarStyle">@style/ActionBarStyle</item>
10 <item name="actionBarStyle">@style/ActionBarStyle</item>
11 <item name="android:windowActionBar">true</item>
12 <item name="android:windowBackground">@drawable/background</item>
13 </style>
14
15 <style name="AppThemeCrop" parent="Theme.AppCompat.Light.DarkActionBar">
16 <!-- Customize your theme here. -->
17 <item name="colorPrimary">@color/colorPrimary</item>
18 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
19 <item name="colorAccent">@color/colorAccent</item>
20 <item name="windowNoTitle">true</item>
21 <item name="windowActionBar">false</item>
22 <item name="android:windowFullscreen">true</item>
23 <item name="android:windowContentOverlay">@null</item>
24 </style>
25 <style name="Pin__AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
26 <!-- Customize your theme here. -->
27 <item name="colorPrimary">@color/colorPrimary</item>
28 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
29 <item name="colorAccent">@color/colorAccent</item>
30 <item name="android:actionBarStyle">@style/ActionBarStyle</item>
31 <item name="actionBarStyle">@style/ActionBarStyle</item>
32 <item name="android:windowActionBar">true</item>
33 <item name="android:windowBackground">@drawable/background</item>
34 </style>
35 <style name="ActionBarStyle" parent="Widget.AppCompat.ActionBar">
36 <item name="android:background">@drawable/header_border</item>
37 <item name="background">@drawable/header_border</item>
38 </style>
39
40 <style name="spinner_style">
41 <item name="android:paddingLeft">10dp</item>
42 <item name="android:paddingTop">3dp</item>
43 <item name="android:paddingBottom">2dp</item>
44 <item name="android:dividerHeight">1dp</item>
45 <item name="android:textSize">17dp</item>
46
47 <item name="android:background">@drawable/campo_blanco_rounded</item>
48 <item name="android:drawableRight">@drawable/ic_spinner</item>
49 <item name="android:scrollbarAlwaysDrawVerticalTrack">true</item>
50 <item name="android:dropDownSelector">@android:color/white</item>
51 </style>
52
53 <style name="spinnerDropDownItemStyle" parent="android:TextAppearance.Widget.TextView.
SpinnerItem">
54 <item name="android:textSize">17dp</item>
55 <item name="android:paddingLeft">6dp</item>
56 <item name="android:paddingTop">2dp</item>
57 <item name="android:background">@drawable/border_bottom_dashed_whitebg</item>
58 </style>
59
60 <style name="general_layout">
61 <item name="android:background">@drawable/border_top</item>
62 </style>
63
64 <style name="green_title">
65 <item name="android:textColor">@color/white</item>
66 <item name="android:background">@drawable/green_title</item>
67 <item name="android:layout_width">match_parent</item>

```

```
68     <item name="android:paddingLeft">10dp</item>
69     <item name="android:paddingTop">3dp</item>
70     <item name="android:paddingBottom">3dp</item>
71     <item name="android:textSize">15dp</item>
72 </style>
73 <style name="row_green_child_bottom">
74     <item name="android:textColor">@color/white</item>
75     <item name="android:background">@drawable/green_child_bottom</item>
76     <item name="android:layout_width">match_parent</item>
77     <item name="android:paddingLeft">10dp</item>
78     <item name="android:paddingTop">3dp</item>
79     <item name="android:paddingBottom">3dp</item>
80     <item name="android:textSize">12dp</item>
81 </style>
82 <style name="blue_name_title">
83     <item name="android:layout_marginBottom">12dp</item>
84     <item name="android:background">@drawable/blue_name_title</item>
85 </style>
86 <style name="green_button">
87     <item name="android:layout_width">wrap_content</item>
88     <item name="android:layout_height">35dp</item>
89     <item name="android:background">@drawable/button_green_custom</item>
90     <item name="android:textColor">@color/white</item>
91     <item name="android:paddingLeft">6dp</item>
92     <item name="android:paddingRight">6dp</item>
93     <item name="android:paddingTop">1dp</item>
94     <item name="android:paddingBottom">1dp</item>
95     <item name="android:textSize">16dp</item>
96     <item name="android:capitalize">none</item>
97     <item name="android:layout_margin">5dp</item>
98     <item name="android:textAllCaps">>false</item>
99 </style>
100
101 <style name="green_button_small">
102     <item name="android:layout_width">wrap_content</item>
103     <item name="android:layout_height">35dp</item>
104     <item name="android:background">@drawable/button_green_custom</item>
105     <item name="android:textColor">@color/white</item>
106     <item name="android:paddingLeft">6dp</item>
107     <item name="android:paddingRight">6dp</item>
108     <item name="android:paddingTop">1dp</item>
109     <item name="android:paddingBottom">1dp</item>
110     <item name="android:textSize">15dp</item>
111     <item name="android:capitalize">none</item>
112     <item name="android:layout_margin">3dp</item>
113     <item name="android:textAllCaps">>false</item>
114 </style>
115
116 <style name="orange_button">
117     <item name="android:background">@drawable/button_orange_custom</item>
118     <item name="android:layout_height">35dp</item>
119     <item name="android:layout_width">wrap_content</item>
120     <item name="android:textColor">@color/white</item>
121     <item name="android:paddingLeft">6dp</item>
122     <item name="android:paddingRight">6dp</item>
123     <item name="android:paddingTop">1dp</item>
124     <item name="android:paddingBottom">1dp</item>
125     <item name="android:textSize">15dp</item>
126     <item name="android:capitalize">none</item>
127     <item name="android:layout_margin">3dp</item>
128     <item name="android:textAllCaps">>false</item>
129 </style>
130
131 <style name="input_text">
132     <item name="android:background">@drawable/campo_blanco_rounded</item>
133     <item name="android:layout_height">20dp</item>
134     <item name="android:layout_marginLeft">5dp</item>
135     <item name="android:padding">8dp</item>
```

```
136     <item name="android:textSize">15sp</item>
137     <item name="android:layout_marginBottom">4dp</item>
138 </style>
139
140 <style name="circle_points">
141     <item name="android:background">@drawable/circle_points</item>
142     <item name="android:textColor">@color/white</item>
143     <item name="android:textStyle">bold</item>
144     <item name="android:textSize">33sp</item>
145     <item name="android:layout_width">70dp</item>
146     <item name="android:layout_height">70dp</item>
147     <item name="android:gravity">center</item>
148     <item name="android:layout_marginTop">4dp</item>
149 </style>
150
151 <style name="AppTheme.NoActionBar">
152     <item name="windowActionBar">false</item>
153     <item name="windowNoTitle">>true</item>
154 </style>
155
156 <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar"
157 />
158 <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
159 <!-- Row Layout Child -->
160 <style name="row_child_linear_name">
161     <item name="android:layout_marginTop">10dp</item>
162     <item name="android:layout_marginLeft">50dp</item>
163 </style>
164 <style name="child_background">
165     <item name="android:background">@drawable/blue_title</item>
166 </style>
167 <style name="congrats_background">
168     <item name="android:background">@drawable/green_price_title</item>
169 </style>
170 <style name="row_child_name">
171     <item name="android:textStyle">bold</item>
172     <item name="android:textColor">@color/white</item>
173     <item name="android:textSize">14sp</item>
174 </style>
175 <style name="row_child_age">
176     <item name="android:textStyle">normal</item>
177     <item name="android:textColor">@color/white</item>
178     <item name="android:textSize">12sp</item>
179 </style>
180 <style name="row_prize_points">
181     <item name="android:textStyle">bold</item>
182     <item name="android:textColor">@color/yellow</item>
183     <item name="android:textSize">11sp</item>
184 </style>
185 <style name="row_child_image">
186     <item name="android:layout_width">85dp</item>
187     <item name="android:layout_height">85dp</item>
188     <item name="android:layout_margin">5dp</item>
189 </style>
190 <style name="open_child_name">
191     <item name="android:textStyle">bold</item>
192     <item name="android:textColor">@color/white</item>
193     <item name="android:textSize">16sp</item>
194 </style>
195 <style name="open_child_image">
196     <item name="android:layout_width">100dp</item>
197     <item name="android:layout_height">100dp</item>
198     <item name="android:layout_marginLeft">5dp</item>
199     <item name="android:layout_gravity">left</item>
200 </style>
201 <style name="circle_points_on_list">
202     <item name="android:background">@drawable/circle_points</item>
```

```
203     <item name="android:textColor">@color/white</item>
204     <item name="android:textStyle">bold</item>
205     <item name="android:textSize">20sp</item>
206     <item name="android:layout_width">50dp</item>
207     <item name="android:layout_height">50dp</item>
208     <item name="android:gravity">center</item>
209     <item name="android:layout_marginRight">5dp</item>
210 </style>
211 <style name="circle_points_to_give">
212     <item name="android:background">@drawable/circle_points</item>
213     <item name="android:textColor">@color/white</item>
214     <item name="android:textStyle">bold</item>
215     <item name="android:textSize">15sp</item>
216     <item name="android:layout_width">30dp</item>
217     <item name="android:layout_height">30dp</item>
218     <item name="android:gravity">center</item>
219     <item name="android:layout_margin">10dp</item>
220 </style>
221 <style name="row_child_points">
222     <item name="android:textColor">@color/white</item>
223     <item name="android:textStyle">normal</item>
224     <item name="android:textSize">11sp</item>
225     <item name="android:layout_width">50dp</item>
226     <item name="android:layout_height">wrap_content</item>
227     <item name="android:gravity">center</item>
228     <item name="android:layout_marginRight">5dp</item>
229 </style>
230 <style name="prizes_grid">
231     <item name="android:columnWidth">60dp</item>
232 </style>
233 <style name="prizes_given_grid">
234     <item name="android:columnWidth">50dp</item>
235 </style>
236 <style name="icon_prizes">
237     <item name="android:layout_width">50dp</item>
238     <item name="android:layout_height">50dp</item>
239     <item name="android:layout_margin">10dp</item>
240     <item name="android:layout_gravity">center_horizontal</item>
241 </style>
242 <style name="grid_icons">
243     <item name="android:layout_marginBottom">20dp</item>
244     <item name="android:columnWidth">60dp</item>
245 </style>
246 <style name="grid_prizes">
247     <item name="android:layout_width">50dp</item>
248     <item name="android:layout_height">50dp</item>
249     <item name="android:layout_marginBottom">2dp</item>
250     <item name="android:layout_gravity">center_horizontal</item>
251 </style>
252 <style name="grid_given_prizes">
253     <item name="android:layout_width">35dp</item>
254     <item name="android:layout_height">35dp</item>
255     <item name="android:layout_marginBottom">1dp</item>
256 </style>
257 <style name="expanded_image">
258     <item name="android:layout_width">240dp</item>
259     <item name="android:layout_height">240dp</item>
260 </style>
261 </resources>
262
```



```
1 <resources>
2 <string name="banner_ad_unit_id">ca-app-pub-2861342412299558/3271483429</string>
3 <string name="app_name">TinPoint</string>
4 <string name="TheChildren">Children</string>
5 <string name="male">male</string>
6 <string name="female">female</string>
7 <string name="age">age</string>
8 <string name="yearsold">years old</string>
9 <string name="Birthdate">Birthdate</string>
10 <string name="Gender">Gender</string>
11 <string name="setDate">Select date</string>
12 <string name="changeDate">Change date</string>
13 <string name="allFields">Enter all fields</string>
14 <string name="intNumberOnly">Points only integers</string>
15 <string name="pointsAndPrizes">Points and Prizes</string>
16 <string name="lifetimePoints">Accumulated points</string>
17 <string name="prizesAvailable">Prizes available</string>
18 <string name="scoreapoint">Score points</string>
19 <string name="unassignPrizes">No added prizes</string>
20 <string name="prizesawarded">Prizes awarded</string>
21 <string name="no">No</string>
22 <string name="yes">Yes</string>
23 <string name="Congratulations">Congratulations!</string>
24 <string name="GivenPrize">Given prize</string>
25 <string name="YouHaveGivenThisPrize">You have given this prize</string>
26 <string name="giveapize">Give a prize</string>
27 <string name="areYouSurePrize">Are you sure to submit this prize?</string>
28 <string name="arwYouSureDeletePrize">Are you sure to delete this prize?</string>
29 <string name="arwYouSureDeleteProfile">Are you sure to delete this profile?</string>
30 <string name="areYouSurePoint">Are you sure to score points?</string>
31 <string name="editProfile">Edit profile</string>
32 <string name="Enterallfields">Enter all fields</string>
33 <string name="Editfields">Edit fields</string>
34 <string name="assignPrizes">Add prize</string>
35 <string name="NewPrizeFor">New prize for </string>
36 <string name="TakePhoto">Take photo</string>
37 <string name="GalleryPicture">Photo gallery</string>
38 <string name="save">Save</string>
39 <string name="addChild">Add child</string>
40 <string name="points">Points</string>
41 <string name="name">Name</string>
42 <string name="remove">Remove</string>
43 <string name="edit">Edit</string>
44 <string name="EditPrizeFor">Edit prize for</string>
45 <string name="AssignPrizeForAllProfiles">Assign prize for all profiles</string>
46 <string name="search">Search</string>
47 <string name="icons">Icons</string>
48 <string name="howManyPoints">How many points?</string>
49 <string name="replaceWithPhoto">Replace with photo</string>
50 <string name="replaceWithGallery">Replace with gallery</string>
51 <string name="replaceWithIcon">Replace with icon</string>
52 <string name="ContinuesToAddPointsToRedeemThisPrize">Continues to add points to redeem
this prize</string>
53 <string name="permission">These permissions are required to use TinPoint</string>
54 <string name="cancel">Cancel</string>
55 <string name="accesto">You need to grant access to</string>
56
57 <string name="pin__description_create">Enter a %1$s digit code to lock %2$s</string>
58 <string name="pin__description_confirm">Re-enter your PIN to confirm</string>
59 <string name="pin__description_unlock">%1$s locked! \n Enter PIN to unlock</string>
60
61 <string name="pin__unlock_error_no_matching_pin_found">There is no PIN to unlock!</string>
62 <string name="pin__unlock_error_match_failed">The PIN code does not match!</string>
63 <string name="pin__unlock_error_insufficient_selection">You have not selected the required
number of digits! Please try again</string>
64 <string name="pin__unlock_error_retry_limit_exceeded">You have failed too many times. You can
try again in %1$s.</string>
65
```

```
66 <string name="pin_toast_lock_success">%1$s locked!</string>
67 <string name="pin_toast_unlock_success">%1$s unlocked correctly!</string>
68 <string name="pin_toast_unlock_required">You need to unlock to do that!</string>
69
70 <string-array name="icons_array">
71 <item>i1</item>
72 <item>i2</item>
73 <item>i3</item>
74 <item>i4</item>
75 <item>i5</item>
76 <item>i6</item>
77 <item>i7</item>
78 <item>i8</item>
79 <item>i9</item>
80 <item>i10</item>
81 <item>i11</item>
82 <item>i12</item>
83 <item>i13</item>
84 <item>i14</item>
85 <item>i15</item>
86 <item>i16</item>
87 <item>i17</item>
88 <item>i18</item>
89 <item>i19</item>
90 <item>i20</item>
91 </string-array>
92 <string-array name="points_array">
93 <item>1</item>
94 <item>2</item>
95 <item>3</item>
96 <item>4</item>
97 <item>5</item>
98 <item>6</item>
99 <item>7</item>
100 <item>8</item>
101 </string-array>
102 </resources>
103
```