UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

"APLICACIÓN PARA EL DESARROLLO MUSICAL INFANTIL EN DISPOSITIVOS ANDROID, USANDO METODOLOGÍA SUM PARA VIDEOJUEGOS"

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: JUAN CARLOS BOLAÑOS TARAPUES

DIRECTOR: ING. CARPIO PINEDA

IBARRA-ECUADOR

2016

DECLARACIÓN

Yo, Juan Carlos Bolaños Tarapues, estudiante de la Facultad de Ingeniería en Ciencias Aplicadas - Carrera de Ingeniería en Sistemas Computacionales, libre y voluntariamente declaro que el presente trabajo de investigación, es de mi autoría y no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo la propiedad intelectual correspondiente a este trabajo a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normativas vigente de la Universidad técnica del Norte.

Firma:

Nombre: Juan Carlos Bolaños Tarapues

C.I: 1002413753

Ibarra a los 14 días del mes de febrero del 2016

CERTIFICACIÓN

Certifico que el Sr. Bolaños Tarapues Juan Carlos; estudiante de la Facultad de Ingeniería en Ciencias Aplicadas-Carrera de Ingeniería en Sistemas Computacionales, ha desarrollado y terminado en su totalidad el trabajo de titulación, "APLICACIÓN PARA EL DESARROLLO MUSICAL INFANTIL EN DISPOSITIVOS ANDROID, USANDO METODOLOGÍA SUM PARA VIDEOJUEGOS", bajo mi supervisión por lo cual firmo su constancia.

ING. CARPIO PINEDA
DIRECTOR DE TESIS

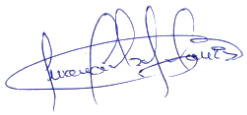
UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Bolaños Tarapues Juan Carlos portador de la cedula Nro. 100241375-3, manifiesto que es mi voluntad de ceder a la Universidad Técnica del Norte, los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador Art. 4, 5 y 6 en calidad de autor del Trabajo de Grado denominado: **"APLICACIÓN PARA EL DESARROLLO MUSICAL INFANTIL EN DISPOSITIVOS ANDROID, USANDO METODOLOGÍA SUM PARA VIDEOJUEGOS.",** que ha sido desarrollado para obtener el título de INGENIERO EN SISTEMAS COMPUTACIONALES en la Universidad Técnica del Norte, quedando facultada la Universidad para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada.

En concordancia se suscribe este documento en el momento en que se hace la entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.



(Firma):

Nombre: Juan Carlos Bolaños Tarapues

Cédula: 100241375-3

Ibarra a los 14 días del mes de febrero del 2016.

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo por sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente información.

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:		100241375-3	
APELLIDOS Y NOMBRES:		Bolaños Tarapues Juan Carlos	
DIRECCIÓN:		Juan José Paez 7-09 y Pablo Anibal Vela	
EMAIL:		jcbolitowow@gmail.com	
TELÉFONO FIJO:	062954-426	**TELÉFONO MÓVIL:**	0983301997

DATOS DE LA OBRA	
TÍTULO:	"APLICACIÓN PARA EL DESARROLLO MUSICAL INFANTIL EN DISPOSITIVOS ANDROID, USANDO METODOLOGÍA SUM PARA VIDEOJUEGOS."
AUTOR:	Bolaños Tarapues Juan Carlos
FECHA:	14 de febrero del 2016
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	■ PREGRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniería en Sistemas Computacionales
ASESOR/DIRECTOR:	Ing. Carpio Pineda

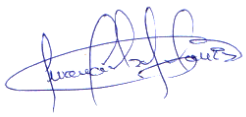
2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Juan Carlos Bolaños Tarapues portador de la cédula de ciudadanía Nro. 100241375-3, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago la entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar los derechos de autor de terceros, por lo tanto, la obra es original y que es titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá a defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra a los 14 días del mes de febrero del 2016

AUTOR: 

(Firma):

Nombre: Bolaños Tarapues Juan Carlos

Cédula: 100241375-3

AGRADECIMIENTOS

A mi madre por haberme brindado su amor apoyo y confianza durante estos años, ayudándome a terminar esta etapa de mi vida.

Agradezco también a María del Hierro y Patricio Aguinaga por quererme como a un hijo y haberme brindado su amor y paciencia.

A mi director de tesis Ing. Carpio Pineda por su paciencia, ayuda y motivación en estos meses, por su guía y enseñanzas en los años de estudio.

A todos mis amigos que de una u otra forma me ayudaron en mi desarrollo profesional, que me ayudaron con sus consejos y apoyo.

Juan Carlos Bolaños Tarapues

DEDICATORIA

Este proyecto de titulación está dedicado a mi madre María Lidia Tarapues, a María Del Hierro y Patricio Aguinaga por haberme criado, enseñarme a ser un hombre de bien, haberme educado y acompañado en todo momento de mi vida.

A todos mis amigos y familiares que me han dado su apoyo y han influido en mí de una manera positiva y me han llevado a este momento de mi vida.

Y a todos los profesores que con sus guías y enseñanzas me han convertido en un buen profesional.

Juan Carlos Bolaños Tarapues

ÍNDICE DE CONTENIDO

ÍNDICE DE GRÁFICOS

ÍNDICE DE TABLAS

RESUMEN

Este proyecto tiene como finalidad realizar y subir al mercado de Google Play una aplicación que permita incentivar el desarrollo de destrezas y habilidades musicales en niños menores de 6 años y brindar a los padres una nueva herramienta de fácil acceso para el desarrollo de la creatividad y la imaginación de sus hijos. Usando la metodología SUM para videojuegos y las mejores herramientas de software libre disponibles.

En el primer capítulo se redacta las razones que han llevado a la decisión de realizar este proyecto, y los objetivos que se pretende conseguir.

En el capítulo dos se realiza un estudio de la metodología de trabajo y las herramientas que se ha decidido usar para el desarrollo de esta aplicación.

En el capítulo tres se describe la planificación de la aplicación de acuerdo a las guías establecidas por la metodología SUM, terminando en el desarrollo de la lista de tareas que incluye todos los pasos para el desarrollo de la aplicación y el cronograma que seguirá dicha lista.

En el capítulo cuatro se detalla el desarrollo de la aplicación y las tareas que cada miembro del grupo de trabajo realizó.

En el capítulo cinco se encuentran las recomendaciones y conclusiones a las que se ha llegado después del desarrollo de este proyecto.

SUMMARY

.

The mean of this project is to develop an upload to Google Play Store an application that encourages the development of music skills and abilities on kids under 6 years old and offer parents an easy access tool to develop creativity and imagination on their child. Using SUM methodology for videogames and the best free software tools available.

In the first chapter are portrayed the reasons that have been followed to carry out this project, and the goals that are aimed to achieve.

In the second chapter a study about the work methodology and the tools that have been selected for the development of this application is realized.

In chapter three the planification of the application according to the SUM methodology guides its explained, ending in the development of the task list that includes all the steps for the development of the application and the timetable that the task list will follow.

In chapter four development of the application and the tasks that every member of the work group realized its detailed.

In chapter five the conclusions and recommendations reached after developing this project are found.

INTRODUCCIÓN

1.1 PROBLEMA

Actualmente la mayoría de padres en la sociedad desconocen las grandes ventajas de la educación musical en los niños, tampoco existen programas de educación musical en centros de desarrollo infantil, casi todas las actividades que desarrollan los padres o los centros de desarrollo infantil con los niños son actividades de desarrollo motriz o desarrollo de lenguaje, más ninguna está enfocada en el desarrollo de habilidades y destrezas musicales, todo esto a pesar de que la música promueve:

- Aumento en la capacidad de memoria, atención y concentración de los niños
- Mayor habilidad para resolver problemas matemáticos y de razonamiento complejos
- Estimula la creatividad y la imaginación infantil
- Ayuda a la integración social
- Mejora la memoria

Por otro lado, la mayoría de hogares en el sector urbano con hijos menores de 6 años cuentan con uno de los mejores dispositivos de enseñanza multimedia como son los teléfonos inteligentes o las *tablets*, pero actualmente en su gran mayoría son usadas casi exclusivamente para entretenimiento, de igual manera en la tienda Google Play casi no existen aplicaciones que promuevan el desarrollo musical de los niños.

1.2 OBJETIVOS

Objetivo general

Realizar y subir al mercado de Google Play una aplicación que permita incentivar el desarrollo de destrezas y habilidades musicales en niños menores de 6 años,

brindar a los padres una nueva herramienta de fácil acceso para el desarrollo de la creatividad y la imaginación de sus hijos.

Objetivos específicos

1. Investigar las mejores tecnologías de software libre aplicables para desarrollar la aplicación de Desarrollo Musical Infantil
2. Analizar la metodología SUM aplicada para la creación de juegos educativos
3. Investigar las mejores estrategias para implementar una aplicación educativa usando Android SDK.
4. Aplicar la metodología SUM para el desarrollo del proyecto "APLICACIÓN PARA EL DESARROLLO MUSICAL INFANTIL"

1.3 ALCANCE

La aplicación pondrá a disposición de los padres una nueva herramienta para el desarrollo intelectual de sus hijos, mediante juegos de memoria y coordinación musical que irán incrementando en dificultad de acuerdo a la respuesta de los niños, esta aplicación estará disponible para descarga desde la tienda Google Play para tablets y teléfonos inteligentes que funcionen con esta tecnología. Todo esto en base a la siguiente descripción:

- Inicio e interfaz sencillas, con grandes botones para fácil acceso
- Sonido de alta calidad
- Menú que permita empezar desde el principio o continuar desde el último punto usado.
- Evitar grandes contrastes de color en animaciones o cualquier parte de la aplicación
- Realizar seguimiento y estadísticas a los logros de los usuarios
- Permitir la incorporación de nuevo contenido por medio de actualizaciones

Para desarrollar estas especificaciones se utilizará el siguiente software libre:

- Android Studio
- Facebook SDK
- GIMP (manejo de imágenes)
- Audacity (editor de audio)
- Open Office (edición de texto y varios)

La metodología a utilizar será SUM, diseñada para videojuegos, la cual brindará una buena base para desarrollar una aplicación educativa, pero con alto contenido multimedia y que trate de ser lo más entretenida posible.

La aplicación estará dividida en 2 grandes partes: la primera, donde los niños aprenderán a llevar el ritmo y aprenderán los compases musicales usando una batería, para esto se tendrá en la pantalla las partes de la batería que se vaya a usar y un contador con los aciertos seguidos que haga el usuario y otro que marque el porcentaje de aciertos, comenzando en el nivel más fácil usando solo un tambor y compases intuitivos como ¾ hasta llegar a compases más difíciles y usar varias partes de la batería, conseguirán avanzar de nivel si pueden mantener el ritmo un mínimo número de aciertos mientras tocan algunas canciones infantiles conocidas, esta parte constará de 10 niveles; la segunda parte, enseñar a los niños a identificar las notas musicales y darles las primeras nociones para leer las notas en un pentagrama, se usará un piano en esta parte, la pantalla estará dividida en 2 partes la primera tendrá un pentagrama y la segunda tendrá 8 teclas de un piano, las notas saldrán en el nivel más fácil en el pentagrama con su respectivo nombre y en el teclado las teclas tendrán el nombre de la nota que representan, conforme subamos de dificultad se irán quitando las ayudas y las canciones irán subiendo de dificultad; de igual manera un mínimo número de aciertos es necesario para pasar de nivel, está diseñada para enseñar las notas y además leer un pentagrama musical, contará con 24 niveles.

Para incentivar la práctica se dará recompensas si logran acabar un nivel con un número muy alto de aciertos, se podrá visitar niveles ya completados y habrá un lugar donde podamos visualizar los niveles completados y los puntajes más altos, así también como logros, y compartir esto en Facebook.

1.3.1 Limitaciones

La aplicación está limitada a dispositivos que puedan acceder a la tienda Google Play, y cuenten mínimo con el sistema operativo Froyo (v2.2), de igual manera solo contará con actividades recomendables para niños menores de 6 años.

1.4 Justificación

Actualmente casi todas las actividades que realiza el ser humano pueden complementarse o mejorarse mediante el uso de la tecnología, en este caso la educación musical que se encuentra abandonada en nuestro medio puede favorecerse también con el desarrollo de aplicaciones que ayuden a esparcir su desarrollo o inculcar su práctica como en este caso.

Aprovechando una de las mejores tiendas y con más rápido crecimiento como es Google Play se tratará de compartir con el mayor número de personas una aplicación que pueda ayudar al desarrollo intelectual de los niños.

Además, para el desarrollo de aplicaciones para Android se provee de software libre de alta calidad como lo es Android SDK, se complementará esto con software para manejo de imágenes y audio libre como lo son GIMP y Audacity, esto ayudará a reducir costos ya que las otras opciones cuestan varios cientos de dólares.

Para el desarrollo de esta aplicación se usará SUM, una metodología diseñada para el desarrollo de videojuegos, que ha sido modificada de la popular SCRUM, lo que brindará un poco más de agilidad, ya que está diseñada para pequeños grupos de trabajo, proyectos cortos y brinda la posibilidad de modificar el producto, aunque ya esté terminado.

1.4.1 Impactos

Este proyecto ayudará a cambiar la percepción en los padres acerca de los dispositivos móviles al momento de pensar en métodos para la educación y desarrollo intelectual de sus hijos, de igual manera busca incentivar más el desarrollo de juegos educativos sobre todo en temas poco tratados como lo ha sido el musical.

MARCO TEÓRICO

2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE SUM

2.1.1 Objetivo

Acerenza, Coppes, Mesa y Viera (2009) afirman:

> La metodología SUM para videojuegos tiene como objetivos desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar la eficacia y eficiencia de esta. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. (p. 38)

Esto se logra ya que SUM permite unir fácilmente equipos interdisciplinarios, en la elaboración de videojuegos se tendrá diferentes aspectos como lo son audio, video, historia y trama del juego, no solo el trabajo se reparte de una mejor manera, sino que cada uno estará trabajando en su área de experiencia.

2.1.2 Roles

La metodología define cuatro roles: Equipo de Desarrollo, Productor Interno, Cliente y Verificador Beta.

- Equipo de desarrollo. El equipo de desarrollo definirá sub roles que se adapten a las necesidades del juego que se desarrollará, siendo los más comunes: Programador, Artista Gráfico, Artista Sonoro y Diseñador de Juego.
- Productor interno. Liderará el equipo y será encargado de mediar entre todas las partes y asegurarse que todos los miembros del equipo sigan todas las reglas y principios establecidos.
- Cliente. Es el representante de las partes interesadas en el desarrollo del proyecto, aparte de definir objetivos y metas para el proyecto él deberá colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.

- Verificador Beta. Aparte de descubrir errores en la aplicación, brindará otro punto de vista que permita ir corrigiendo el proyecto, ya que el videojuego que se desarrolle deberá ser entretenido para el público base.

2.1.3 Proceso de entrega

De acuerdo a: (Acerenza et al., 2009)

El proceso de entrega se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos que se realiza durante todo el proyecto. Las cinco fases secuenciales son: Concepto, Planificación, Elaboración, Beta y Cierre.

- **Concepto**: Para el desarrollo del concepto se deberá realizar 3 tareas, definir aspectos de negocios, técnicos y de juego; en esta instancia deberán participar todos los roles y la mejor estrategia será desarrollar las 3 tareas en paralelo ya que las 3 influyen entre sí.
- **Planificación:** Dos tareas componen esta fase; en la planificación administrativa se definirá el resto de fases del proyecto, y en la especificación del videojuego quedará detallado todas las características del proyecto. Es mejor desarrollar estas 2 tareas en paralelo, ya que la duración del proyecto depende de la complejidad y extensión de los requerimientos del sistema, es decir la una depende de la otra. Además, es recomendable no dedicar mucho tiempo a esta fase, ya que en la fase de desarrollo con cada iteración los requerimientos irán cambiando; es decir la planificación administrativa debe ser flexible.
- **Elaboración:** Se desarrolla de una forma iterativa e incremental, al final de cada iteración debe haber una versión ejecutable del videojuego, el objetivo es ajustar la funcionalidad e ir añadiendo cada vez más prestaciones.

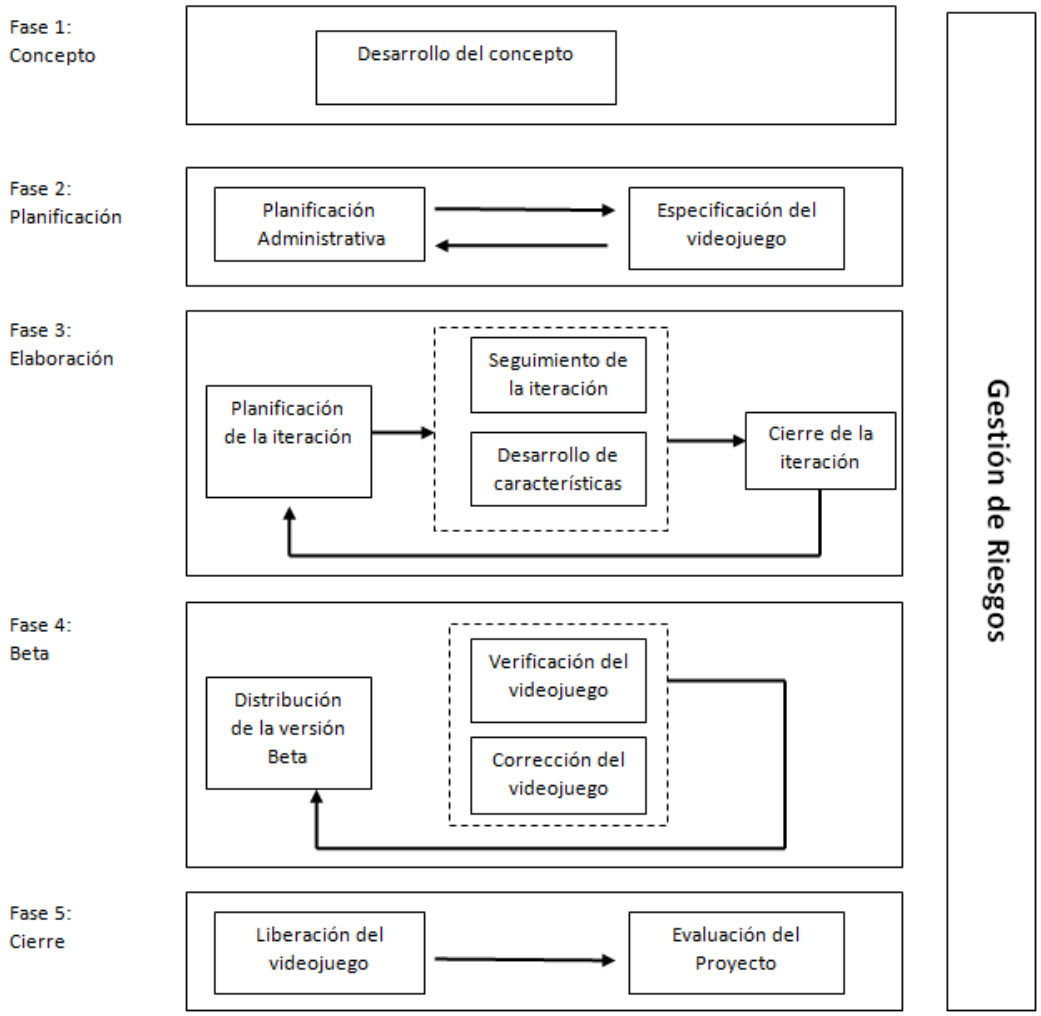


Figura 1: Fases del Proceso

- **Beta:** El propósito de esta fase es evaluar y ajustar distintos aspectos del videojuego, no solo se busca corregir errores que puedan existir en el juego, sino también evaluar aspectos como curva de aprendizaje, dificultad y diversión.
- **Cierre**: Se entregará al cliente no solo el videojuego terminado, sino también cualquier otra cosa pactada como manuales de usuario, documentación del diseño del juego, etc.

  Además, se realizará una evaluación del proyecto, anotando posibles mejoras al proceso, errores cometidos y lecciones aprendidas.

- **Gestión de riesgos:** Se tratará de identificar y monitorear posibles riesgos como atrasos, falta de recursos, abandono de un miembro del equipo, este trabajo lo realizarán el equipo de desarrollo y el Productor Interno.

2.2 APLICACIONES ANDROID

2.2.1 Introducción

Todas las aplicaciones Android están escritas en lenguaje Java; las herramientas de Android SDK compilan el código, datos y recursos en un APK, el mismo que será usado para instalar la aplicación en dispositivos Android.
Características:

- El sistema operativo Android es un sistema Linux multiusuario donde cada aplicación es un diferente usuario.
- Por defecto el sistema asigna a cada aplicación un único ID de usuario Linux, este ID es usado solo por el sistema y es desconocido para el usuario, y los archivos de una aplicación solo podrán ser usados por este ID.
- Cada proceso tiene su propia Máquina Virtual, para que el código de cada aplicación corra en aislamiento.
- Android comienza un proceso cuando un componente de una aplicación debe ser ejecutado, el mismo será cerrado automáticamente cuando no sea necesario o el sistema necesite memoria para otras aplicaciones.

2.2.2. Componentes de una aplicación

Los componentes de una aplicación Android pueden observarse en la tabla 1.

Tabla 1: Componentes de una aplicación Android

Nombre Componente	Funcionalidad	Ejemplos
Activity	Interactuar con el usuario mediante una interfaz gráfica.	Juegos, enviar correos, editar documentos.

Service	Es un proceso que trabaja sin la intervención del usuario, y sin molestar al usuario.	Gestor de descargas, reproductor de música.
Content Provider	Guardar y recuperar datos.	Recuperar contactos del teléfono, guardar notas.
*Broadcast*Receiver	Recibir mensajes y dependiendo el caso iniciar una acción.	La pantalla está inactiva, la batería esta baja.

- **Actividades.** Una actividad generalmente está representada por una pantalla gráfica con una interfaz de usuario. Una aplicación generalmente está compuesta de varias actividades que permiten al usuario poder interactuar completamente con una aplicación, pero todas las actividades que la conforman son independientes unas de otras y en algunos casos podrán ser usadas por otras aplicaciones.
- **Servicios.** Un servicio es un componente que no tiene una interfaz de usuario y que sirve para realizar tareas fuera de la vista del usuario. Las actividades deben entrar en pausa cuando no están en primer plano, pero un servicio no, por eso es recomendable usar servicios para gestionar descargas, reproducir música o cualquier otro proceso que no necesite la intervención del usuario.
- **Proveedores de contenido.** Un proveedor de contenido maneja los datos de una aplicación como fotos, texto, sonido, su uso y su almacenamiento, dado el caso también podrá autorizar a otras aplicaciones a acceder a dichos datos. De la misma manera un proveedor de contenido puede ser útil para leer y guardar información que no se desee que nadie use, excepto la aplicación.
- **Receptores de mensajes.** Es un componente que no tiene interfaz de usuario y solo está esperando a una señal ya sea de otra aplicación o del sistema, una vez que sea momento de entrar en acción llamará a un servicio que se haga cargo o mostrará una alerta en la barra de notificaciones al usuario, como ocurre cuando la batería del teléfono esta baja.

2.2.3 Iniciando componentes

A excepción de los proveedores de contenido los demás tipos de componentes son activados usando un mensaje asíncrono llamado ***intent*.**
Un intent permite enlazar 2 componentes, permitiendo que el primero solicite una acción del segundo, también permite que compartan información y no es necesario que los 2 componentes pertenezcan a la misma aplicación.
Un intent es creado con un objeto Intent, el mismo que define si se activará un componente específico o un tipo específico de componente, un intent puede ser explícito o implícito.
Para actividades y servicios un intent define la acción a realizar y especifica el URI (Identificador de Recursos Uniforme) de los datos que el componente necesita para iniciarse. Para receptores de mensajes el intent simplemente lleva el mensaje que se necesita transmitir.

2.2.4 El archivo *AndroidManifest*

Antes que el sistema operativo pueda iniciar el componente de una aplicación, este debe saber que el componente existe, y lo realiza leyendo el archivo AndroidManifest.xml; las aplicaciones deberán declarar todos sus componentes en este archivo.
Aparte de la declaración de componentes se realizan las siguientes tareas en el archivo AndroidManifest.xml:

- Identificar todos los permisos que la aplicación necesite.
- Declarar el API minino en el que puede funcionar la aplicación.
- Declarar componentes de hardware o software específicos que pueda necesitar la aplicación.
- Declarar librerías API que la aplicación podría necesitar como por ejemplo la librería API de Google maps.

2.2.5 Declarando componentes

La tarea más importante del archivo AndroidManifest es informar al sistema acerca de los componentes de una aplicación. Por ejemplo, una forma de declarar una actividad podría ser la siguiente:

```
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="Music Game" >
        <activity
            android:name="com.musicgame.bateria.bateria.bateria"
            android:configChanges="keyboard|keyboardHidden|orientation"
            android:label="Music Game"
            android:screenOrientation="landscape" >
            <intent-filter...>
        </activity>
    </application>

</manifest>
```

Figura 2: Declaración de una actividad: AndroidManifest.xml

Si las actividades, servicios y proveedores de contenido no son declarados en el manifiesto, estarán ocultos para el sistema y no podrán ser ejecutados. El único componente que puede ser declarado dinámicamente en código o en el archivo AndroidManifest es el receptor de mensajes.

2.2.6 Declarando las capacidades de los componentes

La verdadera funcionalidad y poder de los intents radica en el concepto de *implicitintents*. Un implicitintent le indica al equipo que acciones y que datos puede manejar un componente, esto le permite al sistema encontrar componentes en el dispositivo que puedan realizar determinadas acciones e iniciarlos, en caso de que haya varios componentes que puedan realizar esa acción el usuario elegirá uno. El sistema identifica que componentes pueden responder a un intent buscando en la etiqueta <intent-filter> de cada archivo AndroidManifest uno por cada aplicación instalada en el dispositivo.

Por ejemplo, si se ha desarrollado una aplicación que maneje correo, se puede declarar un *intent-filter* que indique que la aplicación puede manejar a los intents que contengan "enviar correo" de la siguiente forma:

```
<manifest ... >
    ...
    <application ... >
        <activity android:name="com.example.project.ComposeEmailActivity">
            <intent-filter>
                <action android:name="android.intent.action.SEND" />
                <data android:type="*/*" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figura 3: Declaración de un intent-filter: AndroidManifest.xml

Si otra aplicación crea un intent con la acción "ACTION_SEND", el usuario podrá elegir la aplicación de entre los demás programas que puedan enviar correos. En la figura 2.4 se puede apreciar esto.

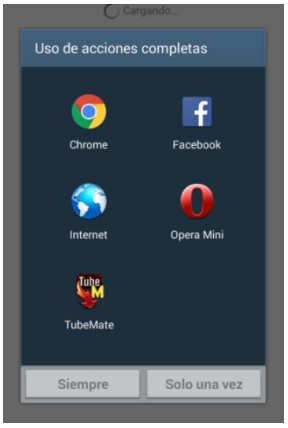


Figura 4: Función de intentfilter:Android OS

2.2.7 Declarando requerimientos mínimos para las aplicaciones

Hay una gran variedad de dispositivos android, y no todos ellos tienen las mismas características o capacidades. Para evitar que las aplicaciones sean instaladas en dispositivos que no las puedan usar, es importante indicar en el archivo AndroidManifest que hardware y software necesita cada aplicación para funcionar. Estas declaraciones servirán para que servicios externos como Google Play filtren las aplicaciones que cada dispositivo puede o no puede usar.

Por ejemplo, si la aplicación usa una cámara y tiene un API de nivel 7, se debe declarar esto en los requerimientos de la siguiente forma.

```
<manifest ... >
    <uses-feature android:name="android.hardware.camera.any"
                  android:required="true" />
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="19" />
    ...
</manifest>
```

Figura 5: Declaración de requerimientos: AndroidManifest.xml

Dispositivos que no posean una cámara y como mínimo una versión de Android 2.1 (api nivel 7) no podrán instalar esta aplicación, es más será invisible para ellos en la tienda de Google.

Si nuestra Aplicación usa cámara, pero no es algo esencial para el funcionamiento de la aplicación, podremos poner el atributo required a falso (false). Y dentro de la aplicación deshabilitar cualquier funcionalidad que use una cámara el momento que se reconozca que el dispositivo no tiene cámara.

2.2.8 Permisos del sistema

De la página AndroidDevelopers (s.f.):

> Android es un sistema operativo de privilegios separados, en el cual cada aplicación se ejecuta con un único sistema de identidad (ID de usuario Linux e ID de grupo). Partes del sistema también son separadas en entidades diferentes. Así Linux aísla a las aplicaciones unas de otras y del sistema.

A través de un sistema de permisos se provee de una seguridad más personalizable, exacta y precisa que provee restricciones para las operaciones que un proceso puede realizar y para la información que puede solicitar.

2.2.8.1 Arquitectura de la seguridad Android

Un punto central de la seguridad en Android es que ninguna aplicación por defecto tiene permisos para realizar operaciones que puedan impactar negativamente a otras aplicaciones, al usuario, al mismo dispositivo o al sistema operativo. Esto incluye no solo corromper y borrar información sino también compartir información privada del usuario o dañar el dispositivo obligándolo a trabajar de una manera anormal, etc.

Debido a que cada aplicación se ejecuta en un *processsandbox,* las aplicaciones deben compartir explícitamente recursos y datos. Esto lo hacen declarando los permisos que necesitan para poder usar recursos que no están disponibles en el *processsandbox* donde se ejecutan. Las aplicaciones declaran estáticamente los permisos que van a requerir y es el sistema operativo Android el que le pregunta al usuario si deben ser concedidos.

2.2.8.2 Id de usuarios y acceso a archivos

Al momento de la instalación de cada aplicación el sistema operativo de Android da una identificación única de Linux llamada Linux user ID. Esta identificación permanece constante mientras la aplicación permanezca en el dispositivo; debemos recordar que esta aplicación tendrá otra identificación de Linux en diferentes dispositivos.

Debido a que la aplicación de la seguridad ocurre a un nivel de procesos, normalmente el código de dos aplicaciones no puede trabajar en el mismo proceso, debido a que las aplicaciones deben trabajar como diferentes usuarios de Linux. Es posible usar el atributo sharedUserId en el archivo AndroidManifest, para que las dos aplicaciones compartan el mismo ID de usuario (asignando a ambas el mismo valor), al hacer esto las 2 aplicaciones serán tratadas como una sola teniendo los mismos permisos e ID de usuario de Linux.

Si se desea que la información de una aplicación sea usada por otras aplicaciones, ya sea en archivos de texto, fotos, sonidos o base de datos hay maneras de compartirlos, seguirán siendo datos de la aplicación, pero estarán accesibles para otras aplicaciones.

2.2.9 Ciclo de vida en las Aplicaciones Android

En el Sistema operativo las actividades se manejan como una pila, cuando una nueva actividad se inicia esta es colocada en el tope de la pila y se convierte en la actividad en ejecución, enviando a la anterior actividad en ejecución al tope de la pila -1, esta actividad no saldrá al primer plano hasta que la actividad que se está ejecutando se termine.

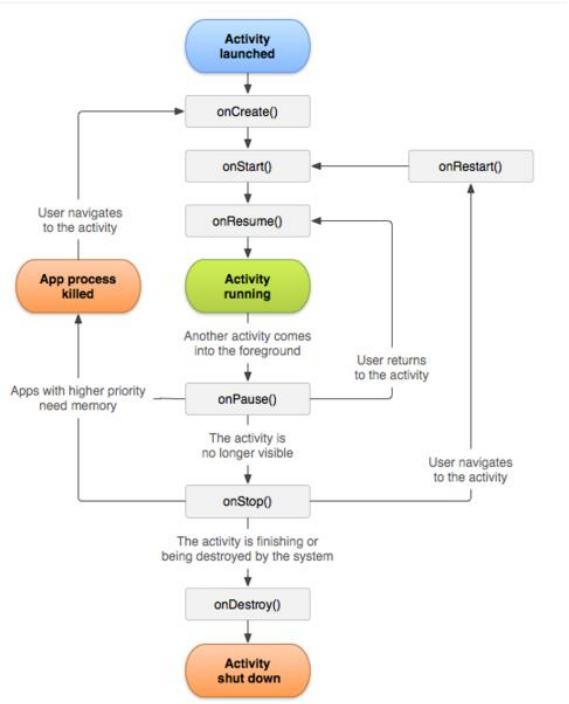


Figura 6: Ciclo de vida aplicaciones Android. Fuente:Android. (s.f.). Activities. Recuperado de: http://developer.android.com/intl/ptbr/reference/android/app/Activity.html

Una actividad puede tener 4 estados, pero solo uno a la vez:

- **Activa** (Running): La actividad está en el tope de la pila y se encuentra ejecutándose, es decir está en primer plano.
- **Pausada**(Paused): La actividad es visible pero otra actividad está en primer plano quitándole el foco, una actividad pausada mantiene todos sus datos e información, pero puede ser eliminada por el sistema en casos de memoria muy baja.
- **Detenida**(Stopped): Esto se da cuando una nueva actividad ocupa completamente la pantalla, es recomendable guardar la información que se encuentre en pantalla.
- **Destruida**(Destroyed): Cuando el sistema necesite recursos destruirá a las aplicaciones que estén pausadas o detenidas.

En la figura 7 se observa los cambios importantes que puede sufrir una aplicación durante su ciclo de vida; los rectángulos grises representan métodos callback, que

se podrá implementar cuando la actividad va pasando de un estado a otro durante su ciclo de vida.

El ciclo de vida de una actividad está dado por los siguientes métodos:

Tabla 2: Ciclo de vida de una actividad Android

Método	Descripción	Siguiente
onCreate()	Es llamado cuando la actividad es creada. Aquí se deberá crear las vistas e inicializar datos, en caso de que sea iniciado desde un proceso destruido, también contendrá un paquete donde se provean los datos previos a la pausa.	onStart()
onRestart()	Se llamara a este método después de que una actividad estuvo detenida, justo antes de onStart().	onStart()
onStart()	Llamada cuando la actividad es visible para el usuario. Seguida de onResume() si la actividad es pausada o por onStop() si la actividad es detenida.	onResume() oronStop()
onResume()	Se usará cuando la actividad vuelve a interactuar con el usuario. En este momento la actividad está en el inicio de la pila de actividades.	onPause()
onPause()	Llamada cuando el sistema está a punto de resumir otra actividad, es usada para guardar datos y detener recursos que puedan estar consumiendo memoria. Seguida por onResume() si la actividad regresa a primer plano o por onStop() si es detenida por otra actividad.	onResumeoronStop()
onStop()	Llamada cuando una actividad nueva está siendo iniciada, una existente ocupa la pantalla o la actividad es destruida.	onRestart() oronDestroy()
onDestroy()	La última llamada antes de que la actividad sea destruida, esto puede pasar debido a que el sistema necesita recursos o alguien utilizó el método finish() en la	

actividad. Con el método isfinishing() podremos saber cuál fue el caso.

2.3 ANDROID STUDIO

2.3.1 Introducción

AndroidSDK provee un Framework rico en aplicaciones que permite construir aplicaciones y juegos innovadores para dispositivos móviles en lenguaje Java.
Android Studio en un par de años desde su lanzamiento Beta ha pasado a ser la herramienta oficial y recomendada por Google para el desarrollo de aplicaciones para el Sistema Operativo Android, ya que las aplicaciones desarrolladas en Android Studio se adaptan a cualquier dispositivo que use Sistema Operativo Android.
Características

- Editor de código inteligente. En el núcleo de Android Studio encontramos un editor de código predictivo, no solo capaz de completar código sobre la marcha, sino de detectar errores, re factorizar y analizar código.
- Plantillas de código e integración con Github. La integración y la facilidad para buscar código a través de Github, realmente ayuda a cualquier programador a familiarizarse con Android Studio y todas las posibilidades que ofrece en el momento de crear aplicaciones.
- Desarrollo para diferentes pantallas. Android Studio por medio de su diseño a través de XML permite que las aplicaciones móviles funcionen en diferentes dispositivos tales como relojes, televisiones, celulares y tablets.
- Emulador elegante y eficiente que contiene casi todos los dispositivos android disponibles en el mercado.
- Compilación mejorada. Con Gradle se podrá crear múltiples instaladores que tengan diferentes características para nuestras aplicaciones usando el mismo proyecto.

2.3.2 Descripción General Android Studio

Vista de proyecto de Android Studio

Esta vista muestra la estructura de los proyectos en dos dimensiones, para brindar un rápido acceso a los archivos del proyecto y ayudar al usuario a trabajar con el sistema de compilado Graddle.

La vista de proyecto:

- Indica los directorios más importantes en modo jerárquico, donde se almacenan los archivos de código.
- Agrupa a todos los archivos necesarios para compilar el código en una sola carpeta.
- Agrupa todos los archivos AndroidManifest.xml de todos los módulos en una sola carpeta.
- Señala todos los archivos de recursos de los diferentes conjuntos de desarrollo Gradle.
- Todos los diseños para diferentes tipos de dispositivos agrupados en una sola carpeta.

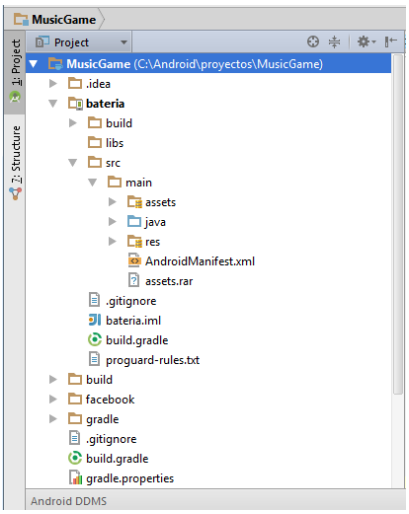


Figura 7: Vista de proyecto: Android Studio

Aunque la vista de proyecto es muy completa se podrá personalizar las vistas para centrarse en aspectos específicos del proyecto como lo son:

- Paquetes
- Archivos de proyecto

- Errores de código
- Producción
- Pruebas

2.3.3 Sistema para Desarrollo de Android

Es el conjunto de herramientas que permiten crear, probar, ejecutar y empaquetar las aplicaciones; aunque está integrado con Android Studio también es flexible y podremos ejecutarlo desde la línea de comandos. Algunas de sus características más importantes son:

- Permite personalizar, configurar, y extender el proceso de compilado.
- Android Build System permite crear múltiples APK con variadas características para las aplicaciones usando el mismo proyecto y módulos.
- Rehusar código y recursos.

2.3.4 Android Virtual Device (AVD) Manager

AVD Manager es un emulador que permite al usuario ejecutar las aplicaciones en dispositivos virtuales, permitiendo realizar pruebas y prototipado rápido; también brinda una amplia gama de dispositivos a emular con las configuraciones más populares en tipos de pantalla y resoluciones.
Todo esto respaldado con el Supervisor de Ejecución Acelerada por Hardware de Intel (HAXM por sus siglas en ingles), que es instalado por Android Studio.

2.3.5 Monitor de Memoria y CPU

Ya que dentro del sistema operativo Android la memoria RAM y el procesador siempre son compartidos por varias aplicaciones al mismo tiempo, es muy importante que cada aplicación use lo estrictamente necesario. Android Studio cuenta con un monitor que ayuda a verificar el rendimiento de cada aplicación, El monitor de memoria y CPU permite encontrar objetos desasignados, fugas de memoria y visualizar gráficamente el uso de memoria y CPU en tiempo real. Como se muestra en la figura 9.

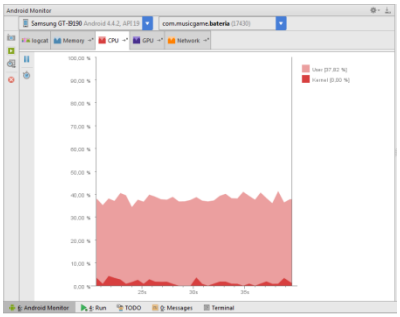


Figura 8: Monitor de Memoria y CPU: Android Studio

2.3.5.1 HeapDump

Es una función incorporada en el monitor de Memoria y CPU de Android Studio, que permite conocer en un determinado momento que clases se encuentran activas, instancias de cada clase y un árbol de referencia que ayuda a identificar el uso de memoria y encontrar fugas de memoria.

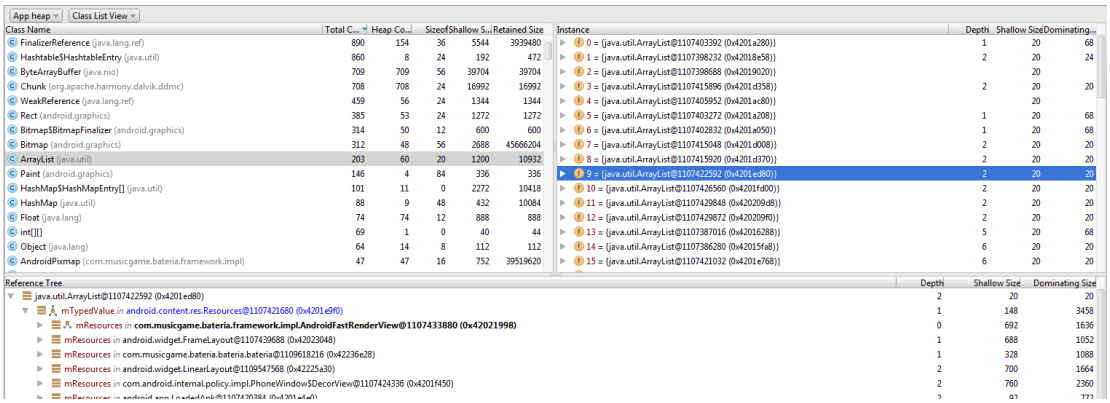


Figura 9: Heap Dump: Android Studio

2.3.5.2 Allocation Tracker

Android Studio permite saber cómo y dónde se está asignando la memoria mientras monitorea su uso; esto permite al usuario saber dónde van ciertos objetos cuando se realizan ciertas acciones; el saber esto permitirá manejar más eficientemente las llamadas a esos métodos sobretodo en la creación y destrucción.

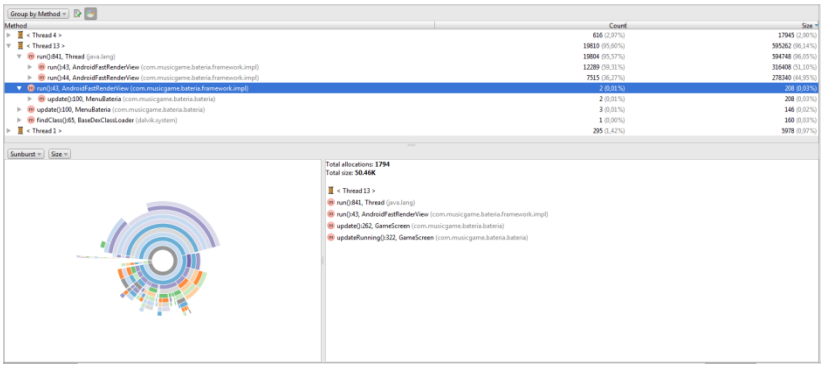


Figura 10: AllocationTracker: Android Studio

2.3.6 Inspecciones de código

En la página web de AndroidDevelopers (s.f.) se manifiesta que:

> Lint es una herramienta de análisis de código estático que verifica los archivos de código de los proyectos buscando posibles errores, optimizando la seguridad, desempeño, usabilidad, accesibilidad, exactitud e internacionalización.

Una de las ventajas de Lint es que es muy personalizable, podemos crear perfiles y habilitar o deshabilitar inspecciones de acuerdo al tipo de proyecto.

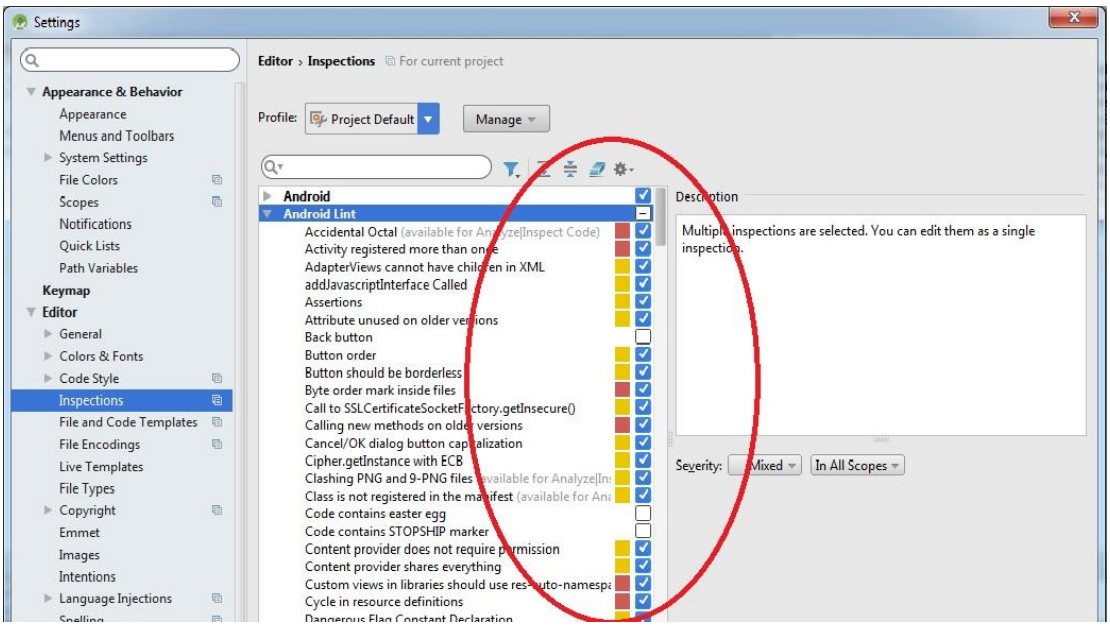


Figura 11: Configurar Lint: Android Studio

2.4 FACEBOOK SDK

2.4.1 Introducción

Facebook SDK es un conjunto de herramientas y librerías, que permite a aplicaciones de diferentes plataformas acceder a las ventajas que brinda Facebook, además de brindar una manera rápida de crear cuentas.

Al ser un producto de una de las más grandes empresas del planeta Facebook SDK continuamente está evolucionando, haciéndose más rápido y confiable, brindando continuamente una mejor experiencia para usuarios y para desarrolladores.

2.4.2 Razones para usar Facebook SDK

2.4.2.1 Creación de cuentas

Las aplicaciones que incorporen Facebook Login tendrán una gran ventaja sobre otras aplicaciones, ya que los usuarios que posean Facebook podrán crear cuentas con solo aplastar un botón, esto no solo se traduce en un ahorro de tiempo y molestias para el usuario, sino en una herramienta para desarrolladores, ya que les brinda una conexión directa con sus usuarios en la plataforma social más grande del planeta.

2.4.2.2 Personalización

Facebook puede brindar datos que serían muy difíciles de recopilar de otra manera, como cosas que le gustan al usuario, lugar de nacimiento, fecha de cumpleaños entre otras. Herramientas que permiten a un desarrollador brindar una experiencia diferente para cada usuario.

2.4.2.3 Desarrollo social

La mayoría de las aplicaciones que un usuario juega durante un largo periodo de tiempo tienen algo en común, el aspecto social, la posibilidad de jugar o interactuar con amigos. En el 2014 de los 10 juegos para Facebook más jugados escogidos por la cadena de noticias Time en su portal, todos tenían un aspecto social que les permitía interactuar con otros amigos que tuvieran ese juego.

2.4.3 LoginReview

Una de las preocupaciones de Facebook es que todos sus usuarios tengan una experiencia segura confiable y consistente, es por eso que todas las aplicaciones que usan datos del usuario pasen por una rigurosa revisión. Para asegurar la mejor experiencia para los usuarios y al mismo tiempo que proteja su privacidad.

Las aplicaciones solo pueden obtener 3 datos del usuario sin ser objeto de una revisión:

- Perfil publico
- Correo electrónico
- Amigos en común que usen la aplicación

2.4.4 Ingresos a través de Facebook

Los desarrolladores podrán tener ingresos económicos a través de sus juegos en Facebook de dos formas:

- Propaganda. Los desarrolladores podrán introducir propaganda en sus aplicaciones, y en caso de cumplir las metas propuestas por los patrocinadores los desarrolladores recibirán una compensación monetaria.
- Productos virtuales. Los desarrolladores podrán ofrecer productos virtuales, por ejemplo: Niveles extras, Objetos exclusivos o moneda virtual. Estos productos los podrán adquirir a cambio de un pago realizado con una tarjeta de crédito o algún otro método soportado por Facebook que se incluyen en Facebook SDK, como se indica en la figura 12.

Figura 12: Facebook métodos de pago.

2.4.5 Facebook Analitics

Esta herramienta provee a desarrolladores varios tipos de estadísticas acerca de sus aplicaciones, la gente que las usa, los lugares donde las descargan, el número de veces que las usan. Todo orientado para brindar una mejor aplicación a los usuarios y que el desarrollador pueda llegar al mayor número posible de gente con su aplicación. En la figura 14 se puede apreciar algunas de las estadísticas disponibles en Facebook analitics.

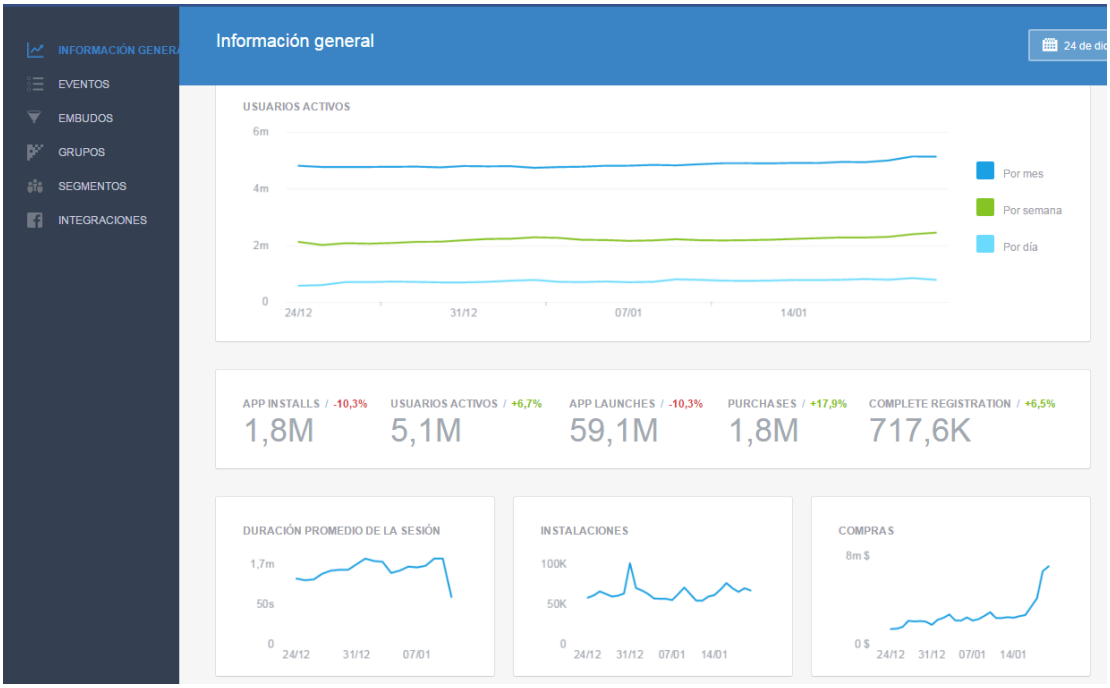


Figura 13: Facebook Analitics: Ejemplo

DESCRIPCIÓN Y FUNCIONAMIENTO DEL SISTEMA

Antes de comenzar se definirá los roles del equipo ya que es importante que todos aporten ideas y participen desde el desarrollo del concepto.

Tabla 3: *Equipo de trabajo*

Nro.	Rol	Nombre
1	Productor Interno	Juan Carlos Bolaños
2	Cliente	Lcda. Amparo Pozo
3	Equipo de desarrollo	Lic. Jorge Luis Untuña
4	Equipo de desarrollo	Tecnólogo Andrés Espinoza
5	Equipo de desarrollo	Juan Carlos Bolaños
6	Verificador Beta	Juan Carlos Bolaños

3.1 DESARROLLO DEL CONCEPTO

3.1.1 Visión

Es un juego educativo pensado para niños, el cual enseñará acerca del ritmo y de las notas musicales por medio de juegos musicales donde los niños puedan aprender mientras se divierten. El juego está pensado para dispositivos móviles, ya que será mucho más fácil para los niños interactuar con el juego en una pantalla táctil.

3.1.2 Género

Es un juego educativo, musical e infantil.

3.1.3 Mecánica del juego

En la pantalla el jugador contará con controles que simulen un bombo y platillos, mientras suena una canción el usuario deberá tratar de llevar el ritmo tocando estos instrumentos de percusión, en la parte central de la pantalla podrá ver qué instrumentos deberá tocar a continuación y a qué tiempo. En la segunda parte del juego el usuario podrá tocar un piano para aprender a reconocer las siete notas musicales, practicará con sencillas canciones y también podrá ver en la pantalla las notas que vienen a continuación.

3.1.4 Público Objetivo

Este juego está destinado para niños menores de 6 años y padres que deseen una herramienta para el desarrollo intelectual de sus hijos.

3.1.5 Plataforma Objetivo

Este juego está diseñado para dispositivos móviles que tengan sistema operativo Android.

3.1.6 Tecnología y herramientas

El lenguaje de programación será Java, y el juego se desarrollará en Android Studio, la parte gráfica y el sonido serán desarrollados con herramientas de software libre como lo son GIMP y Audacity.

3.1.7 Bocetos

Los primeros bocetos desarrollados por el equipo de desarrollo, basados en los aportes de todo el grupo de trabajo se plasman a continuación.

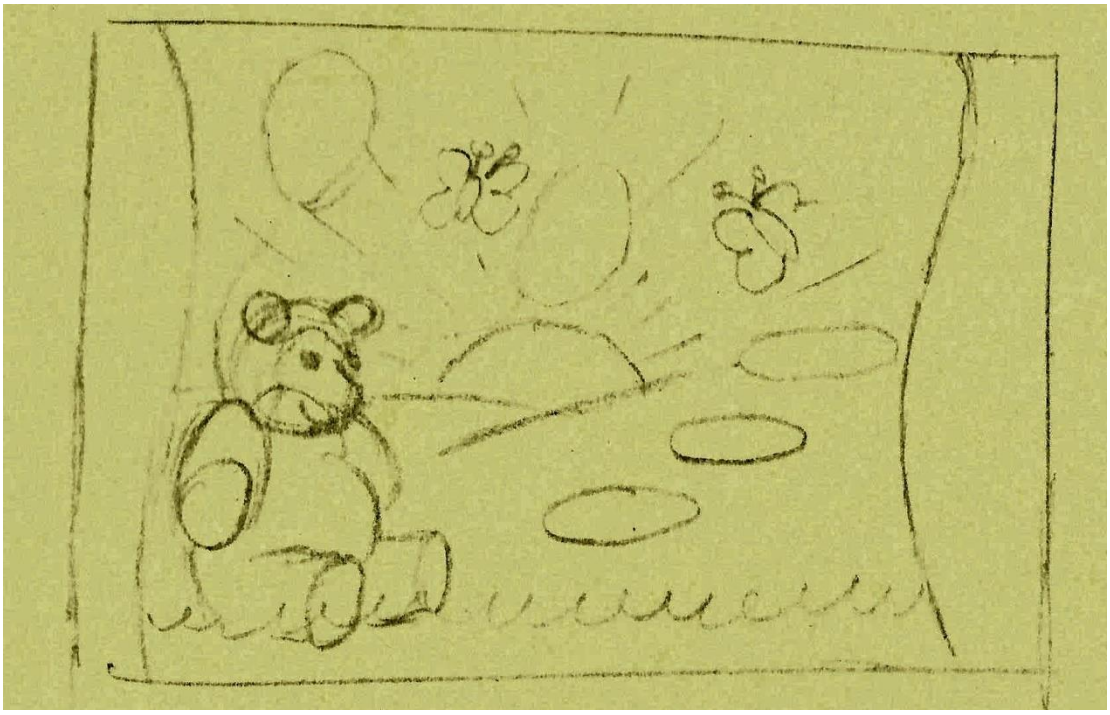


Figura 14: Boceto #1 Pantalla Principal

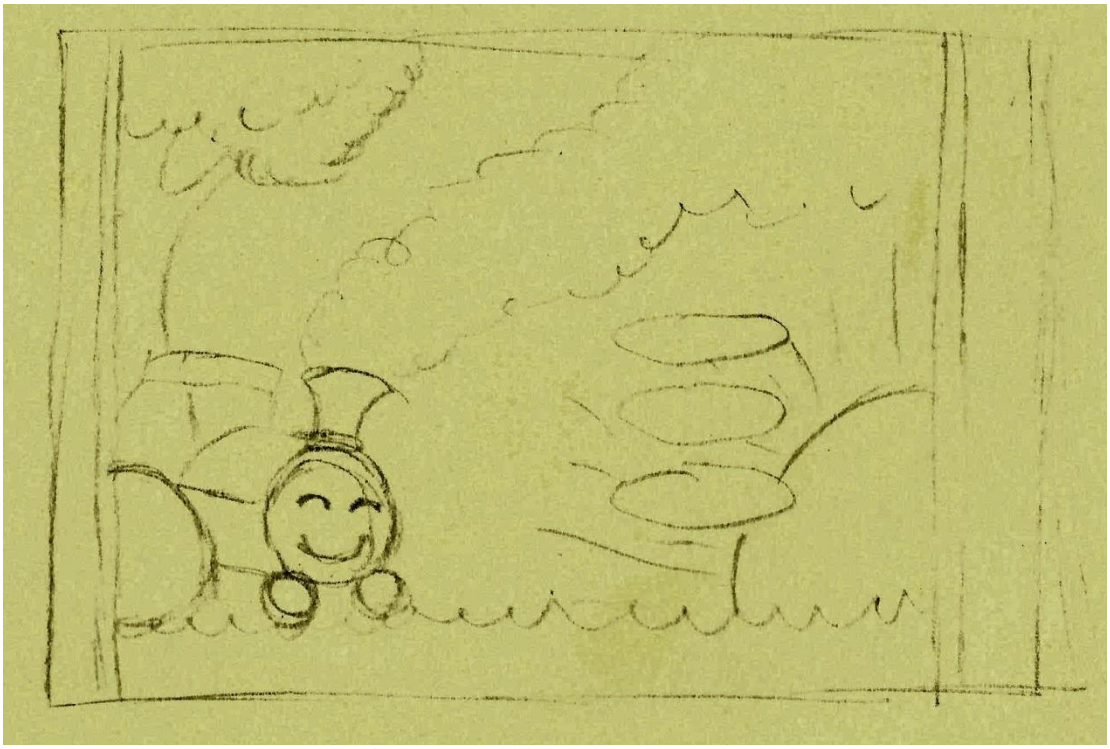


Figura 15: Boceto #2 Pantalla Principal

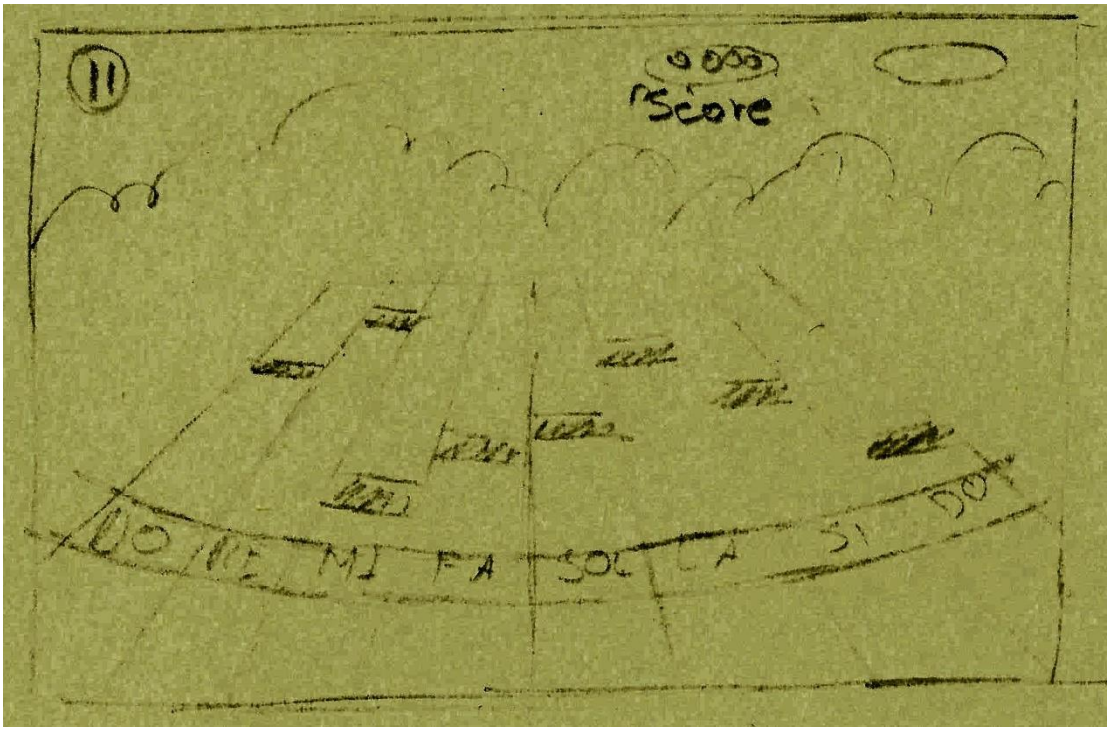


Figura 16: Boceto #1 Juego Piano

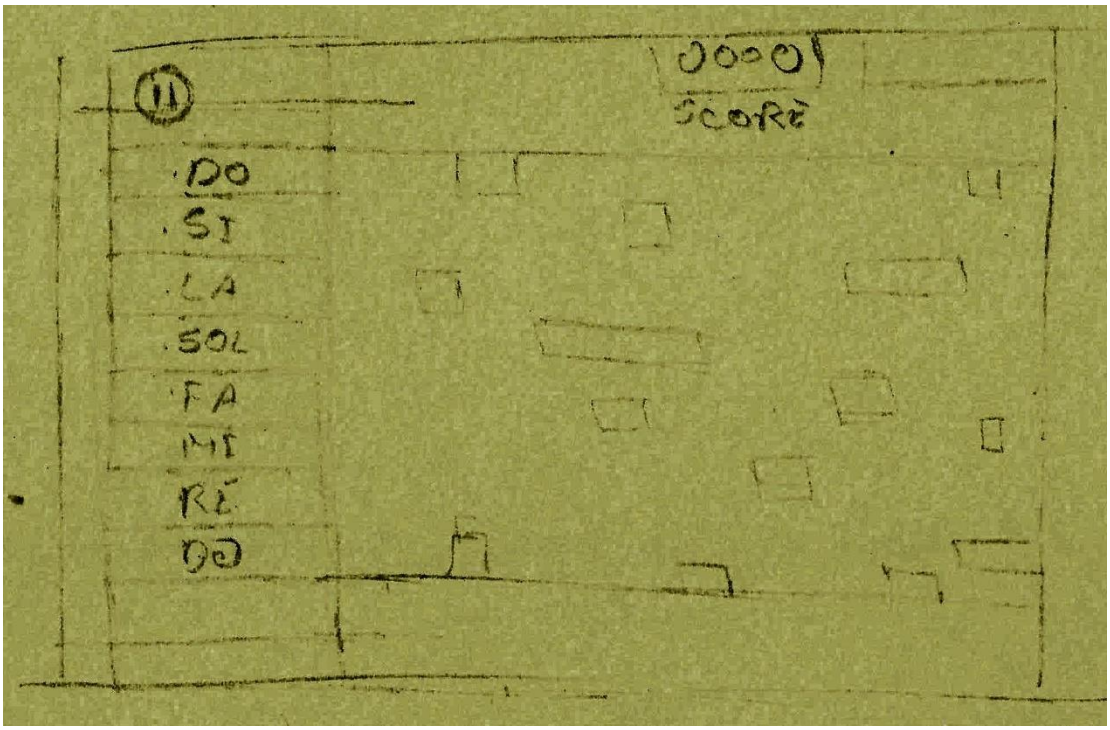


Figura 17: Boceto #2 Juego Piano

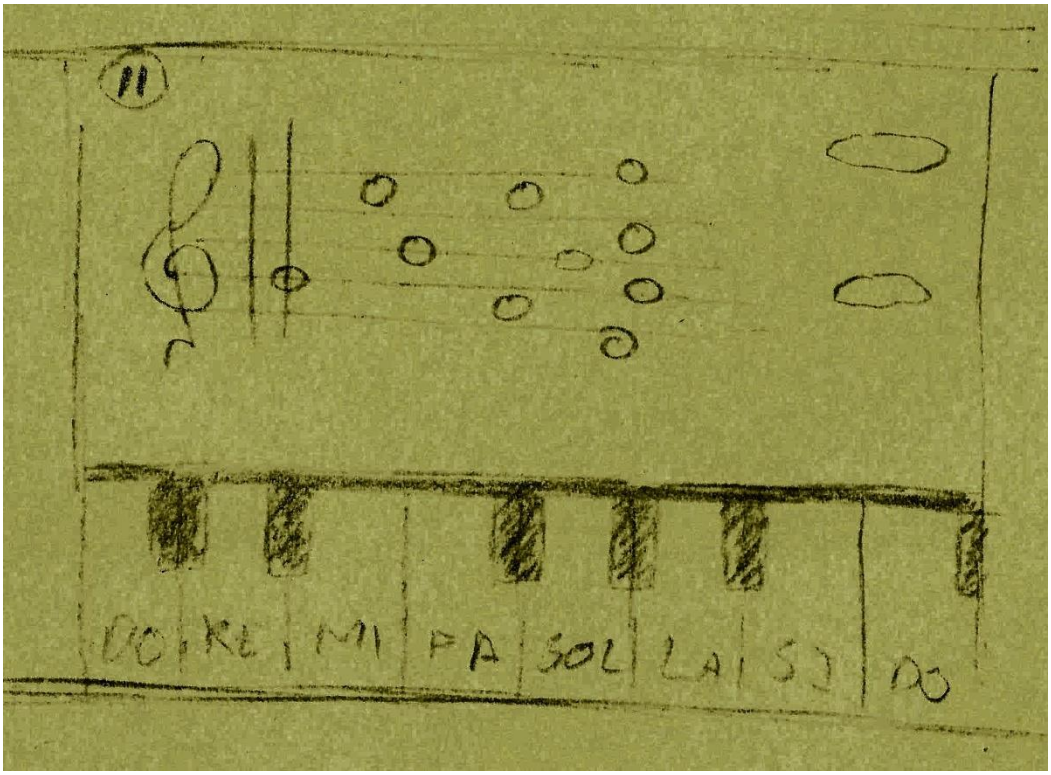


Figura 18: Boceto #3 Juego Piano

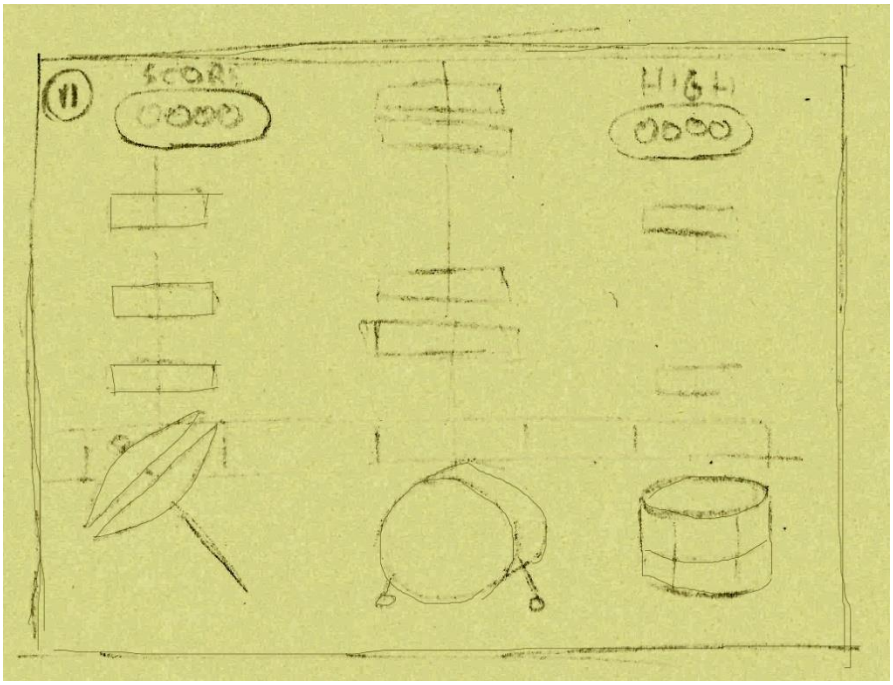


Figura 19: Boceto #1 Juego Batería

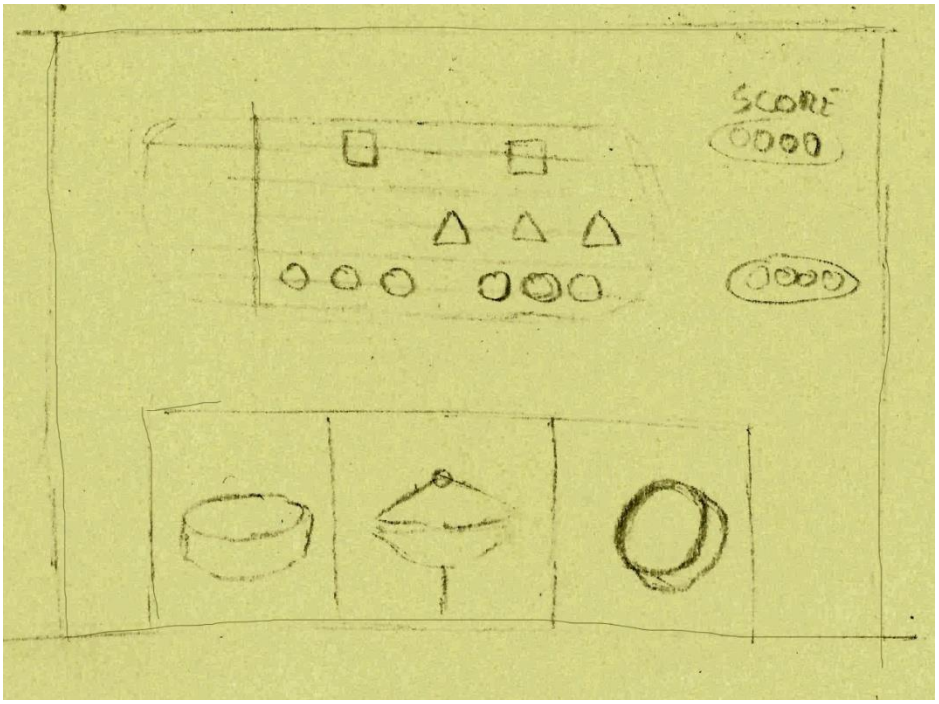


Figura 20: Boceto #2 Juego Batería

3.2 PLANIFICACIÓN DE LA APLICACIÓN

En conjunto con el Cliente se definieron los objetivos que deberá cumplir el proyecto, las características del videojuego y el cronograma detallado.

3.2.1 Objetivos del proyecto

3.2.1.1 Objetivo general

Desarrollar un juego que permita incentivar el desarrollo de habilidades y destrezas musicales en niños menores de 6 años.

3.2.1.2 Objetivos específicos

- Enseñar a los niños a llevar el ritmo
- Enseñar a los niños las notas musicales
- El juego debe ser fácil de aprender
- La interfaz del juego debe ser atractiva y con un tema infantil

- Permitir que los usuarios se conecten con Facebook para beneficios adicionales

3.2.2 Características del videojuego

Todas las características que el equipo de desarrollo deberá tomar en cuenta en el videojuego se detallan a continuación.

3.2.2.1 Modos de juego

- **Percusión/Ritmo**
  En este modo de juego el usuario contará con 2 o 3 instrumentos de percusión (tambores, bombo o platillos), estos estarán ubicados en la parte baja de la pantalla, y como el juego es para dispositivos móviles con pantallas táctiles serán controlados simplemente con los dedos, el usuario oirá una canción infantil, y en una barra que se va actualizando constantemente podrá ver que instrumento deberá tocar y que instrumentos siguen a continuación, también contendrá diferentes elementos que aumenten la inmersión del jugador como animaciones especiales cuando este ganando.

- **Piano/notas**
  Este modo de juego enseñara al usuario las diferentes notas musicales y como están situadas en un piano, el usuario interactuara por medio de un pequeño teclado compuesto de 8 teclas que representaran una octava y una tecla extra, en un pentagrama el usuario podrá apreciar las teclas que deberá tocar para armar la canción, así como el momento en el cual deberá tocarlas. Dependiendo de sus aciertos el jugador podrá avanzar a más niveles.

3.2.2.2 Interfaz

La interfaz se realizará con motivos para niños, en colores pasteles y sin mucho contraste para no molestar sus ojos, además para acceder a las diferentes partes

del juego contarán con botones grandes que faciliten la navegación. En la siguiente figura se muestra las pantallas que tendrá el juego.

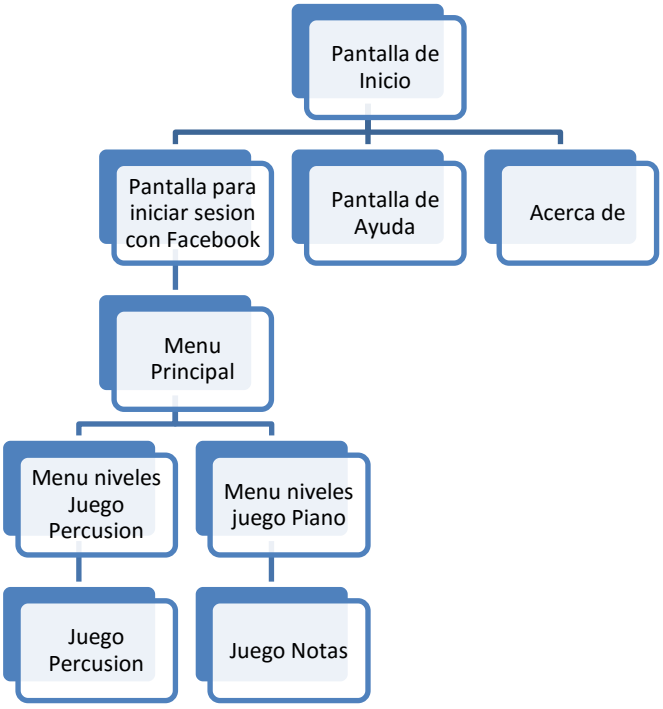


Figura 21: Pantallas de Juego: MusicGame

3.2.2.3 Controles

Los controles son muy intuitivos ya que son táctiles, estos ocupan gran parte de la pantalla para que sea fácil accionarlos para niños pequeños. Como se muestra en la figura 22.

Figura 22: Controles: MusicGame

3.2.2.4 Conectividad con Facebook

Facebook no solo permite brindar beneficios a los jugadores como comparar puntajes con sus otros amigos, compartir records, también permitirá jugar con diferentes cuentas en el mismo dispositivo.

3.2.3 Definir equipo de desarrollo

El equipo de desarrollo estará compuesto por un diseñador gráfico, un programador y un técnico en sonido.

Tabla 4: ***Equipo de desarrollo***

Nro.	Rol	Nombre	Alias
1	Programador	Juan Carlos Bolaños	JB
2	Diseñador gráfico	Tecnólogo Andrés Espinoza	EX
3	Artista sonoro	Lic. Jorge Luis Untuña	JL

3.2.4 Lista de Tareas Final

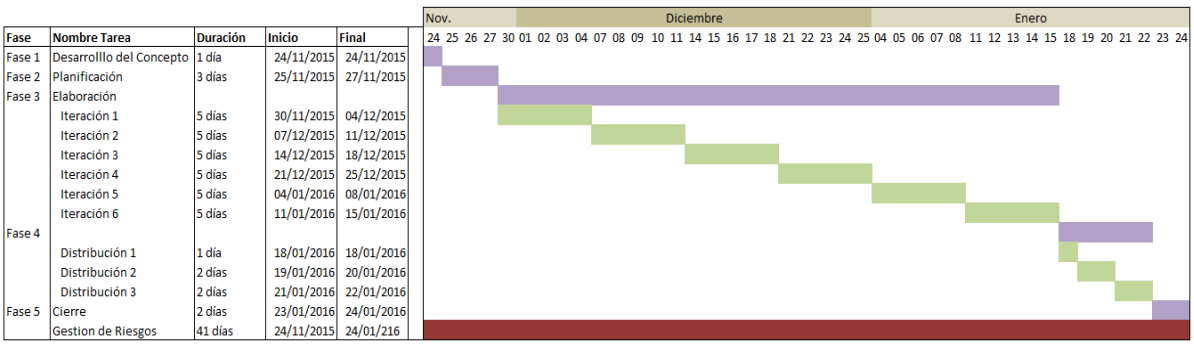
Tabla 5: *Lista de Tareas Final*

Ítem	Descripción	Horas
	Iteración 1	
1	Realizar el borrador interfaz de la pantalla "Juego percusión" (Gráficos)	12
2	Elegir canción y sonidos necesarios para la primera prueba, así como tiempos en donde se debería tocar cada instrumento. (Sonido)	4
3	Implementar clases y métodos que permitan manejar sonido y gráficos. (Programación)	6

4	Formar el primer borrador del juego con los recursos proporcionados en las 3 tareas anteriores	20
	Iteración 2	
5	Realizar diseño de la interfaz gráfica para "Menú principal", "Menú niveles juego percusión", "Pantalla record", "Pantalla de inicio".	8
6	Definir 9 canciones más para usar en el juego.	8
7	Implementar la navegación en las nuevas pantallas diseñadas.	2
8	Implementar métodos para controlar los diferentes estados que puede tener una aplicación Android.	8
9	Grabar físicamente progresos del jugador (niveles y records).	8
	Iteración 3	
10	Realizar diseño de la interfaz gráfica para "Pantalla de juego", "Pantalla de opciones", "Pantalla Acerca de", "Pantalla para iniciar sesión con Facebook" y "Menú niveles juego piano".	12
11	Terminar de implementar la navegación a través de todas las pantallas del videojuego	2
12	Implementar Facebook para el videojuego	16
	Iteración 4	
14	Diseñar la interfaz gráfica para juego notas.	6
15	Revisar colores y temas a través de todas las pantallas gráficas en el videojuego.	4
16	Entregar 9 canciones restantes con los tiempos donde debe tocarse cada instrumento.	16

17	Revisar y optimizar código	4
	Iteración 5	
18	Definir 10 canciones en piano para "Juego Notas"	8
19	Realizar las clases y métodos necesarios para incorporar "Juego Notas".	12
	Iteración 6	
20	Revisar todo el contenido gráfico del juego y realizar las ultimas correcciones	2
21	Entregar 10 canciones de piano con sus respectivas notas para "Juego Notas".	16
22	Implementar las últimas canciones en "Juego Notas"	2
23	Crear recursos para idioma inglés	4

3.2.5 Cronograma



Fase	Nombre Tarea	Duración	Inicio	Final
Fase 1	Desarrolllo del Concepto	1 día	24/11/2015	24/11/2015
Fase 2	Planificación	3 días	25/11/2015	27/11/2015
Fase 3	Elaboración			
	Iteración 1	5 días	30/11/2015	04/12/2015
	Iteración 2	5 días	07/12/2015	11/12/2015
	Iteración 3	5 días	14/12/2015	18/12/2015
	Iteración 4	5 días	21/12/2015	25/12/2015
	Iteración 5	5 días	04/01/2016	08/01/2016
	Iteración 6	5 días	11/01/2016	15/01/2016
Fase 4				
	Distribución 1	1 día	18/01/2016	18/01/2016
	Distribución 2	2 días	19/01/2016	20/01/2016
	Distribución 3	2 días	21/01/2016	22/01/2016
Fase 5	Cierre	2 días	23/01/2016	24/01/2016
	Gestion de Riesgos	41 días	24/11/2015	24/01/216

Figura 23: Cronograma: MusicGame

DESARROLLO DE LA APLICACIÓN

4.1 ELABORACIÓN

El desarrollo del proyecto se completó en 6 semanas, a continuación, se detalla el trabajo realizado cada semana en periodos llamados Iteraciones.

4.1.1 Elaboración Iteración 1

El objetivo de esta iteración es obtener una versión básica pero funcional de la primera parte del proyecto "Juego Percusión".

4.1.1.1 Planificación de la iteración

Dividir las tareas de la Lista de Tareas en sub tareas, y asignar a cada miembro del equipo.

Tabla 6: *Lista de tareas iteración 1*

Ítem	Descripción	Horas	Encargado
	Primera iteración		
1	Realizar el borrador interfaz de la pantalla "Juego percusión" (Gráficos)	12	EX
1.1	Diseñar borrador en papel de la interfaz grafica	3	EX
1.2	Digitalizar imágenes y guardarlas en formato PNG	9	EX
2	Elegir canción y sonidos necesarios para la primera prueba, así como tiempos en donde se debería tocar cada instrumento. (Sonido)	4	JL
2.1	Elegir canción apropiada que cumpla los requisitos previstos.	1	JL

2.2	Grabar canción en formato OGG, al igual que los sonidos de bombo, tambor y platillo.	3	JL
3	Implementar clases y métodos que permitan manejar sonido y gráficos. (Programación)		JB
3.1	Desarrollar clase LoadingScreen, para cargar gráficos y sonidos que se usarán en el juego.	1	JB
3.2	Desarrollar clases NotaCanción y Barra que permita seguir el comportamiento de las notas en pantalla, seguir aciertos y errores del usuario cuando toque los instrumentos	19	JB
3.3	Desarrollar clase GameScreen, la cual presentará la interfaz gráfica para que el usuario juegue, animaciones, sonidos y la lógica del juego.		JB
4	Formar el primer borrador del juego con los recursos proporcionados en las 3 tareas anteriores	6	JB

4.1.1.2 Desarrollo de características

A continuación, se detalla el trabajo realizado por el equipo de desarrollo.

- **Diseñador gráfico**

Los pasos que sigue el diseñador gráfico para desarrollar las pantallas y elementos que se van a usar se detallan en la siguiente tabla:

Tabla 7: Procedimiento para la realización de la pantalla del Juego Batería

Paso	Procedimiento
1	Basado en los bocetos aprobados en el desarrollo del concepto realizar bosquejos del fondo y de cada figura por separado.
2	Simplificar cada ilustración

3	Escanear las ilustraciones
4	Dimensionar las imágenes de acuerdo a los requerimientos proporcionados
5	Editar las ilustraciones en GIMP
6	Depurar gráficos y colores para cada gráfico
7	Convertir imágenes a ficheros PNG

En la figura 24 podremos apreciar la primera pantalla del juego.

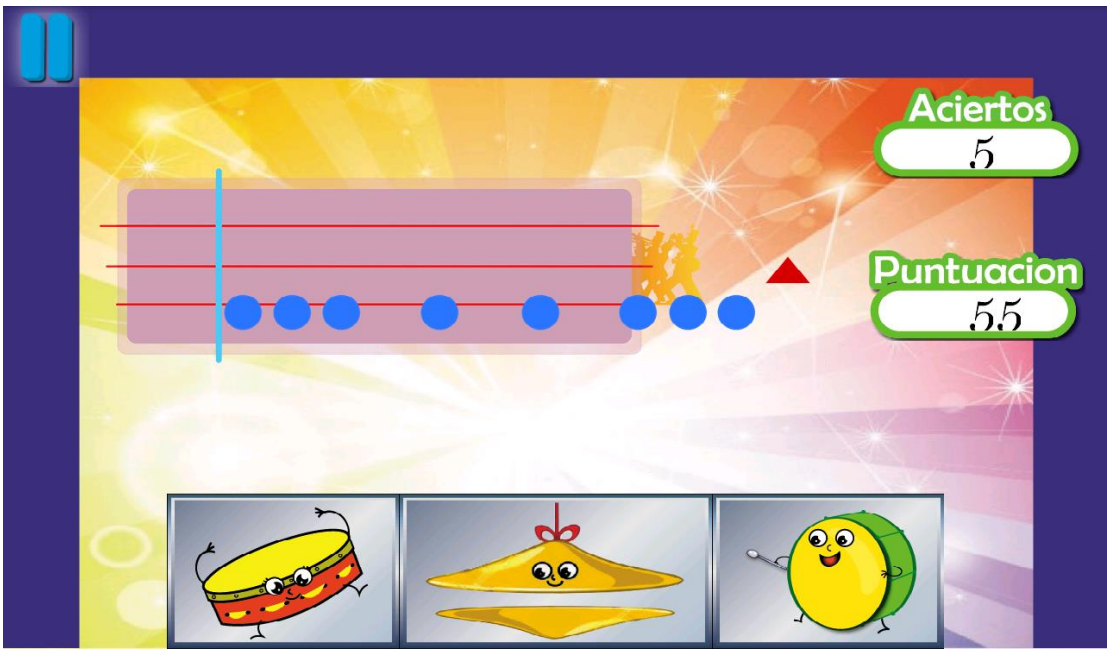


Figura 24: Imagen digitalizada de: Juego Percusión

- **Técnico sonido**

El procedimiento realizado por el técnico de sonido en la creación de la melodía fue el siguiente:

Tabla 8: *Pasos para el diseño de melodías*

Pasos	Procedimiento
1	Buscar una canción infantil que sea sencilla y tenga instrumentos de percusión o se los pueda añadir sencillamente.

2	Elegir una canción y buscar un archivo midi de la misma.
3	Abrir el archivo midi en audacity
4	Organizar los canales de instrumentos
5	Cambiar los sonidos de cada canal para mejorar su sonido
6	Contar las notas
7	Ajustar el ritmo y el tiempo
8	Anotar en que tiempo debe tocarse cada instrumento (en centésimas de segundo)
9	Guardar la melodía en formato OGG

El archivo de audio se puede encontrar en los anexos del CD.

- **Programador**

Los pasos que sigue el programador se detallan en la tabla 9.

Tabla 9: *Pasos para la creación de las clases en la 1ra Iteración*

	Procedimiento
1	Definir las mecánicas del juego
2	Definir los actores que participan en el juego
3	Crear clases que puedan imitar el comportamiento de los diferentes actores
4	Definir el mundo donde se desarrolla el juego
5	Crear una clase que pueda imitar el comportamiento del mundo, es decir donde actuarán los actores.
6	Crear métodos y clases necesarios para manejar sonidos e imágenes.

7	Crear una clase que permita al usuario visualizar e interactuar con las clases que representan al mundo y a los actores en el juego, incorporar los recursos gráficos y de sonido proporcionados.

En la figura 26 se puede ver las clases creadas para la Primera iteración

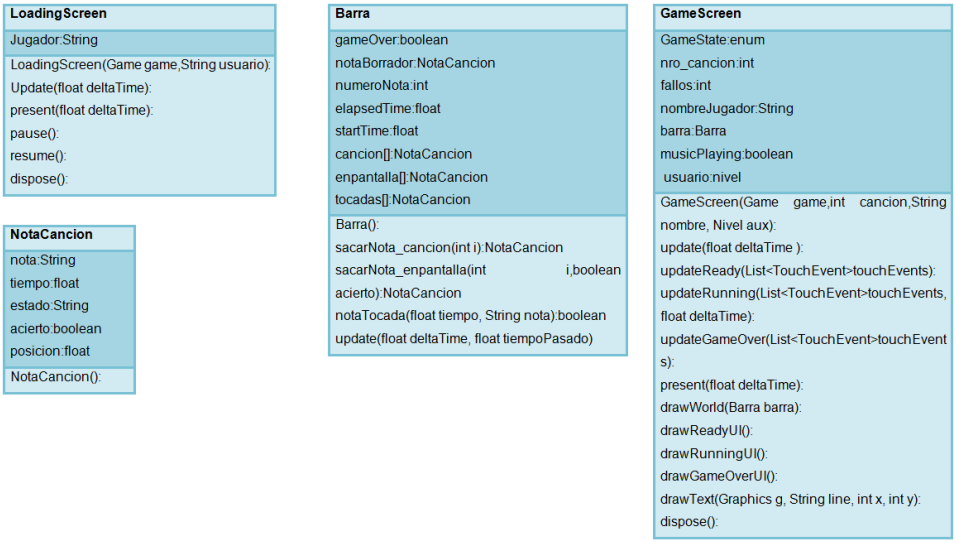


Figura 25: Clases y métodos: Primera iteración

4.1.1.3 Seguimiento de la iteración

En la gráfica 26 podemos ver el progreso real y el planificado para la primera iteración.

Burndown chart

Iteración	Día	Progreso Planeado	Actual	Balance Planeado	Actual	Porcentaje del proyecto
0	0			192	192	0
1	1	8	4	184	188	4,17
1	2	8	8	176	180	8,33
1	3	9	7	167	173	13,02
1	4	8	8	159	165	17,19
1	5	9	9	150	156	21,88

Figura 26: Burn Down Chart: Primera Iteración

4.1.2 Elaboración Iteración 2

El objetivo en esta iteración es pulir el trabajo realizado en la iteración 1, hasta obtener un producto con el que se pueda jugar, obtener y guardar puntajes, y que pueda manejar los diferentes estados que puede enfrentar una aplicación Android.

4.1.2.1 Planificación de la iteración

Tabla 10*: Lista de Tareas Iteración 2*

Ítem	Descripción	Horas	Encargado
Segunda iteración			
5	Realizar diseño de la interfaz gráfica para "Menú principal", "Menú niveles juego percusión", "Pantalla record", "Pantalla de inicio".	8	EX
5.1	Diseñar interfaz "Menú principal"	2	EX
5.2	Diseñar interfaz "Menú niveles juego percusión"	2	EX
5.3	Diseñar interfaz "Pantalla record"	2	EX
5.4	Diseñar interfaz "Pantalla de inicio"	2	EX
6	Definir 9 canciones más para usar en el juego.	8	JL
7	Implementar la navegación en las nuevas pantallas diseñadas.	2	JB
8	Implementar métodos para controlar los diferentes estados que puede tener una aplicación Android.	8	JB
8.1	Crear métodoupdatePause() en la clase GameScreen	4	JB
8.2	Crear métodoonResume() en la clase GameScreen	3	JB
8.3	Crear método onPause() en la clase GameScreen	1	JB
9	Grabar físicamente progresos del jugador (niveles y records).	8	JB
9.1	Crear métodograbar_puntajes() en la clase GameScreen	6	JB

9.2	Crear método cargar en la clase GameScreen	2	JB

4.1.2.2 Desarrollo de características

- **Diseñador gráfico**

Se siguió el mismo procedimiento descrito en la tabla 7.

Figura 27: Pantalla inicio

Las demás pantallas creadas en esta iteración están disponibles en el CD de anexos.

- **Programador**

Se implementaron los siguientes métodos en la clase GameScreen para permitir cálculo de puntaje y porcentajes de aciertos, además permitir grabar el progreso y permitir manejar pausas tanto del usuario como del sistema.

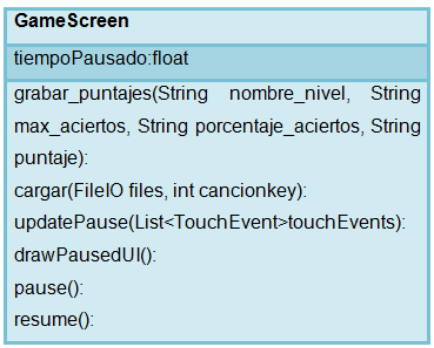


Figura 28: Métodos creados: Segunda iteración

4.1.2.3 Seguimiento de la iteración

El progreso realizado en este ciclo se puede apreciar en la siguiente figura.

Burndown chart						
Iteración	Día	Progreso Planeado	Actual	Balance Planeado	Actual	Porcentaje del proyecto
2	6	6	4	144	152	25,00
2	7	6	4	138	148	28,13
2	8	6	7	132	141	31,25
2	9	6	8	126	133	34,38
2	10	8	8	118	125	38,54

Figura 29: Burn Down Chart: Segunda Iteración

4.1.3 Elaboración Iteración 3

El objetivo principal en esta iteración es permitir al usuario conectarse con Facebook, y terminar el desarrollo de la interfaz gráfica.

4.1.3.1 Planificación de la iteración

Tabla 11: *Lista de Tareas Iteración 3*

Ítem	Descripción	Horas	Encargado
Tercera iteración			
10	Realizar diseño de la interfaz gráfica para "Pantalla de juego", "Pantalla de opciones", "Pantalla Acerca de", "Pantalla para iniciar sesión con Facebook" y "Menú niveles juego piano".	12	EX
10.1	Diseñar interfaz "Pantalla de juego"	2	EX

10.2	Diseñar interfaz "Pantalla de opciones"	2	EX
10.3	Diseñar interfaz "Pantalla Acerca de"	2	EX
10.4	Diseñar interfaz "Pantalla para iniciar sesión con Facebook"	2	EX
10.5	Diseñar interfaz "Menú niveles juego piano"	4	EX
11	Terminar de implementar la navegación a través de todas las pantallas del videojuego	2	JB
12	Implementar Facebook para el videojuego	16	JB
12.1	Implementar los métodos y clases necesarios para permitir el inicio de sesión con Facebook	14	JB
12.2	Crear métodos y clases que permitan validar la sesión en Facebook a través de todo el juego.	2	JB

4.1.3.2 Desarrollo de características

- **Diseñador gráfico**

Se siguió el mismo procedimiento descrito en la tabla 7, los diseños de las pantallas se pueden apreciar en los anexos.

- **Programador**

Se implementa la siguiente clase, con el propósito de permitir que los usuarios puedan conectarse al juego por medio de Facebook.

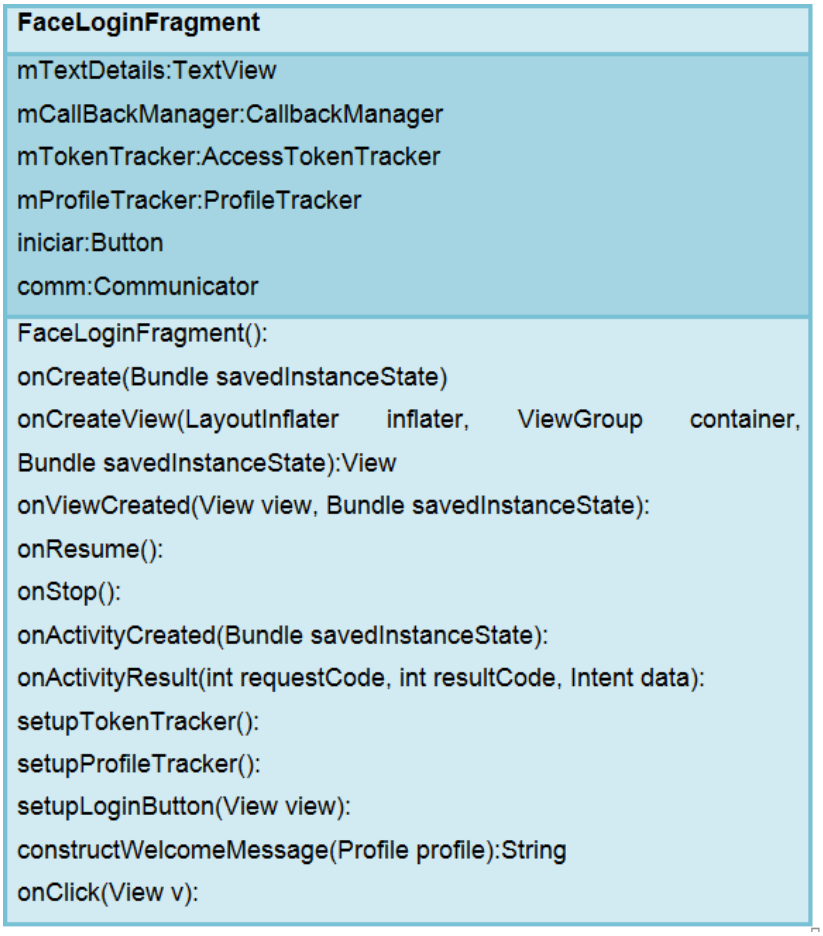


Figura 30: FragmentFacebook: Tercera Iteración

4.1.3.3 Seguimiento de la iteración

El progreso realizado en este ciclo se puede apreciar en la figura 31.

Burndown chart

Iteración	Día	Progreso Planeado	Progreso Actual	Balance Planeado	Balance Actual	Porcentaje del proyecto
3	11	8	6	110	119	42,71
3	12	8	8	102	111	46,88
3	13	8	7	94	104	51,04
3	14	8	7	86	97	55,21
3	15	6	6	80	91	58,33

Figura 31: Burn Down Chart: Iteración 3

4.1.4 Elaboración Iteración 4

En esta iteración se prepara el camino para Juego notas, que es la segunda parte del videojuego, todo el trabajo se detalla a continuación.

4.1.4.1 Planificación de la iteración

En la siguiente tabla se puede ver como se ha repartido el trabajo para esta iteración.

Tabla 12: *Lista de Tareas Iteración 4*

Ítem	Descripción	Horas	Encargado
Cuarta iteración			
14	Diseñar la interfaz gráfica para juego notas.	6	EX
15	Revisar colores y temas a través de todas las pantallas gráficas en el videojuego.	4	EX
16	Entregar 9 canciones restantes con los tiempos donde debe tocarse cada instrumento.	16	JL
17	Revisar y optimizar código	4	JB

4.1.4.2 Desarrollo de características

- **Diseñador gráfico**

Se siguió el mismo procedimiento descrito en la tabla 7, a continuación, se puede ver el diseño final para la segunda parte del Juego.

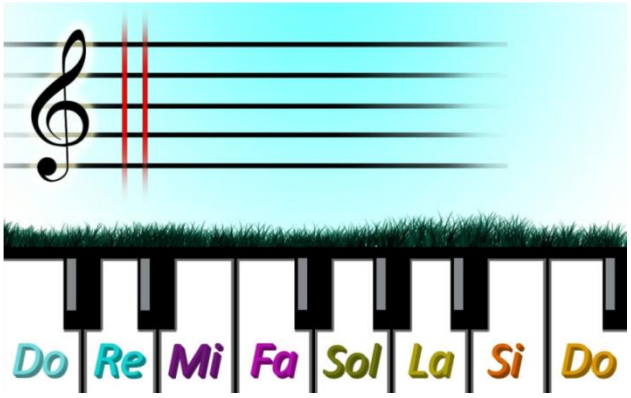


Figura 32: Pantalla principal: Juego Notas

- **Técnico sonido**

Se siguió el mismo procedimiento descrito en la tabla 8, las canciones pueden oírse en los anexos presentados en CD.

- **Programador**

Con ayuda del inspector de código de Android Studio se elimina variables y asignaciones no utilizadas, redundancia y cualquier otra cosa que pueda conducir a un mal uso de la memoria RAM por parte de la aplicación.

4.1.4.3 Seguimiento de la iteración

El seguimiento a la cuarta iteración se puede apreciar en la siguiente figura.

Burndown chart

Iteración	Día	Progreso		Balance		Porcentaje del proyecto
		Planeado	Actual	Planeado	Actual	
4	16	8	10	72	81	62,50
4	17	8	9	64	72	66,67
4	18	8	14	56	58	70,83
4	19	6	6	50	52	73,96
4	20	6	8	44	44	77,08

Figura 33: Burn Down Chart: Iteración 4

4.1.5 Elaboración Iteración 6

El objetivo principal en esta iteración es implementar la segunda parte del Juego, las tareas a realizar se pueden ver a continuación.

4.1.5.1 Planificación de la iteración

En la siguiente tabla se puede ver como se ha repartido el trabajo para esta iteración.

Tabla 13: *Lista de Tareas Iteración 5*

Ítem	Descripción	Horas	Encargado
Quinta iteración			
18	Definir 10 canciones en piano para "Juego Notas"	8	JL
19	Realizar las clases y métodos necesarios para incorporar "Juego Notas".	8	JB
19.1	Implementar clases y métodos que permitan al usuario interactuar con"Juego Notas".	4	JB
19.2	En la pantalla de ayuda implementar varias pantallas para que el usuario pueda aprender cómo jugar.	4	JB

4.1.5.2 Desarrollo de características

- **Programador**

  Se crea la siguiente clase para permitir al usuario interactuar con Juego Notas.

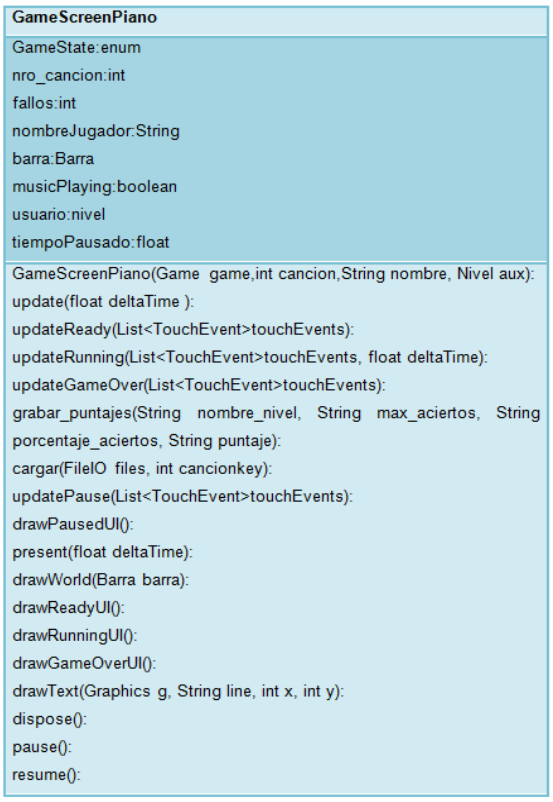


Figura 34: FragmentFacebook: Quinta Iteración

- **Seguimiento de la iteración**

  El avance de la iteración 5 se puede ver en la siguiente figura.

Burndown chart						
		Progreso		Balance		Porcentaje
Iteración	Día	Planeado	Actual	Planeado	Actual	del proyecto
5	21	4	4	40	40	79,17
5	22	4	0	36	40	81,25
5	23	4	0	32	40	83,33
5	24	4	8	28	32	85,42
5	25	4	8	24	24	87,50

Figura 35: Burn Down Chart: Iteración 5

4.1.6 Elaboración Iteración 6

En esta iteración se dan los últimos pasos para terminar el juego, el desarrollo de la misma se puede ver a continuación.

4.1.6.1 Planificación de la iteración

Tabla 14: *Lista de Tareas Iteración 6*

Ítem	Descripción	Horas	Encargado
Sexta iteración			
20	Revisar todo el contenido grafico del juego y realizar las ultimas correcciones	2	EX
21	Entregar 10 canciones de piano con sus respectivas notas para "Juego Notas".	16	JL
22	Implementar las últimas canciones en "Juego Notas"	2	JB
23	Crear recursos para idioma inglés	4	JB

4.1.6.2 Desarrollo de características

- **Técnico sonido**

  En la siguiente tabla podemos apreciar los pasos que sigue el técnico de sonido para obtener los archivos necesarios para la segunda parte del juego.

Tabla 15: *Pasos para el diseño de melodías piano*

Pasos	Procedimiento
1	Buscar una canción infantil que sea sencilla y pueda ser tocada con una mano y en una octava del piano.
2	Elegir una canción y buscar un archivo midi de la misma.
3	Abrir el archivo midi en audacity

4	Transportar la melodía en caso de ser necesario
5	Cambiar los sonidos de cada canal para mejorar su sonido
6	Contar las notas
7	Ajustar el ritmo y el tiempo
8	Guardar en un archivo de texto en que tiempo debe tocarse cada nota (en centésimas de segundo)
9	Grabar en ficheros OGG los sonidos de cada nota necesaria para tocar las melodías seleccionadas usando un piano

---

4.1.6.3 Seguimiento de la iteración

En la figura 36 se puede ver el trabajo realizado en la última iteración, y en la figura 37 se aprecia el trabajo pendiente a través de las seis iteraciones.

Burn down chart

Iteración	Día	Progreso		Balance		Porcentaje del proyecto
		Planeado	Actual	Planeado	Actual	
6	26	6	8	18	16	90,63
6	27	4	8	14	8	92,71
6	28	4	8	10	0	94,79
6	29	6	0	4	0	97,92
6	30	4	0	0	0	100,00

Figura 36: Burn Down Chart: Iteración 6

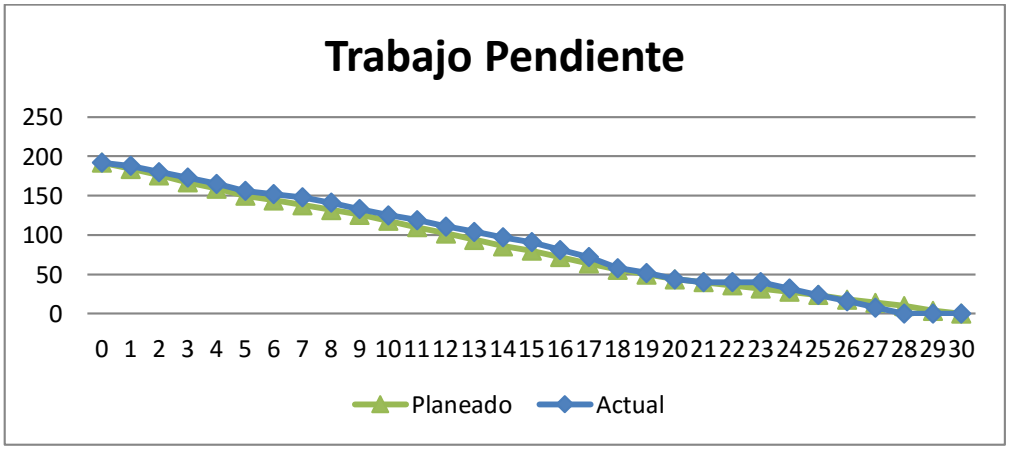


Figura 37: Trabajo Pendiente: Sexta Iteración

4.2 FASE BETA

4.2.1 Verificador Beta

Antes de que el juego salga al mercado se debe probar el juego y encontrar la mayor cantidad de errores que el resto del equipo haya pasado por alto, debido a que el juego está dedicado para niños, se debe mostrar el juego a niños y reportar sus reacciones y comentarios.

4.2.2 Primera Iteración

4.2.2.1 Aspectos a verificar

- Errores
- Gameplay
- Entretenimiento
- Dificultad
- Curva de aprendizaje

4.2.2.2 Evaluación y errores encontrados

En la siguiente tabla se detallan los errores y recomendaciones sugeridas por el verificador.

Tabla 16: Evaluación del verificador Beta

Errores	
Tipo	Descripción
Sonido	Cuarto nivel juego batería la canción no se detiene después de terminado el nivel.
Programación	Juego percusión el botón de reiniciar no funciona, reinicia en otros niveles.

Programación	Juego notas, el contador de aciertos no trabaja adecuadamente cuando se toca 2 notas a la vez.
Gráficos	Juego notas y juego batería, debe decir porcentaje y no puntuación.
Gráficos	Foto del usuario no escala correctamente en dispositivos pequeños.
Programación	Spiner idioma, pantalla Opciones no guarda la opción seleccionada
Programación	Juego notas el quinto nivel se muestra como bloqueado en algunas ocasiones aun cuando se ha pasado exitosamente el cuarto nivel.
Programación	Juego notas y juego batería se congelan pequeños momentos.

Gameplay

Tipo	Descripción
Ninguna	No hubo sugerencias para cambiar el modo de juego.

Entretenimiento

Tipo	Descripción
Diseño	Juego batería, se sugiere eliminar los 2 primeros niveles las canciones son muy repetitivas.
Gráficos	Las pantallas de inicio, usuario y opciones son muy similares.

Dificultad

Tipo	Descripción
Sonido	Juego notas, canción "London Bridge" muy difícil.
Sonido	Juego batería, juego notas ningún instrumento o nota debe repetirse 2 veces o más en un periodo de tiempo menor a 20 milisegundos.

Curva de aprendizaje

Tipo	Descripción
Sonido	Juego batería y notas. Se debe reorganizar los niveles hay canciones difíciles al principio y fáciles al final.

Diseño	Juego batería, los niños tienen dificultad en tocar 2 instrumentos al mismo tiempo se recomienda poner canciones con estas características en niveles finales.
Diseño	Juego notas, se recomienda tomar en cuenta que los niños tienen dedos pequeños, la mayor dificultad para ellos es tocar notas muy separadas en poco tiempo, se recomienda poner esas canciones en últimos niveles.

4.2.1.3 Lista de cambios priorizados

Lo más importante que se pudo apreciar en la primera iteración fue que el juego consume mucha memoria RAM y en dispositivos antiguos esto afecta mucho el modo de juego.

La curva de aprendizaje debe revisarse detenidamente, es importante que los niños puedan avanzar desde el primer al último nivel del juego en un solo camino sin baches ni retrocesos.

Tabla 17: Lista de Tareas Iteración 1 Fase Beta

Ítem	Descripción	Horas	Encargado
Quinta iteración			
1	Reducir el consumo de memoria RAM de la aplicación.	8	JB
2	Arreglar errores y bugs reportados por verificador.	2	JB
3	Desarrollar otro fondo de pantalla	1	EX
4	Reorganizar los niveles para que se adecuen a las sugerencias presentadas por el verificador	6	JL

4.2.3 Segunda Iteración

4.2.3.1 Aspectos a verificar

- Errores
- Entretenimiento
- Dificultad
- Curva de aprendizaje

4.2.3.2 Evaluación y errores encontrados

En la tabla 18 se detallan los errores y recomendaciones sugeridas por el verificador.

Tabla 18: *Evaluación del verificador Beta*

Errores	
Tipo	Descripción
Programación	Juego batería, puntaje decimo nivel se graba en el lugar del noveno.
Programación	Pantalla de inicio y menús, cuando el juego se va a segundo plano varias veces al regresar se mezclan los gráficos de la pantalla anterior con la nueva.
Programación	Juego notas, bajo ciertas circunstancias el octavo nivel no se inicializa, en su lugar el juego carga el séptimo nivel.
Entretenimiento	
Tipo	Descripción
Ninguna	No hay sugerencias
Dificultad	
Sonido	Canción "London Bridge" continúa siendo muy difícil.
Curva de aprendizaje	

Diseño	Juego notas, faltan canciones de nivel intermedio

4.2.3.3 Lista de cambios priorizados

Se debe prestar total atención a la curva de aprendizaje, ya que es uno de los aspectos más importantes del juego permitir a los niños aprender de una manera fácil y divertida.

Tabla 19: *Lista de Tareas Iteración 2 Fase Beta*

Ítem	Descripción	Horas	Encargado
Quinta iteración			
1	Arreglar errores y bugs reportados por verificador.	2	JB
2	Incorporar 3 nuevas canciones con un nivel de dificultad medio	10	JL
3	Reducir la dificultad canción "London Bridge"	1	JL

4.2.4 Tercera Iteración

4.2.4.1 Aspectos a verificar

- Errores
- Dificultad
- Curva de aprendizaje

4.2.4.2 Evaluación y errores encontrados

En la siguiente tabla se detallan los errores y recomendaciones sugeridas por el verificador.

Tabla 20: *Evaluación del verificador Beta*

Errores	
Tipo	Descripción
Programación	Problemas para grabar puntuaciones cuando se ha iniciado sesión en Facebook y después se ha cortado el internet.
Dificultad	
Ninguna	No se reporta ninguna sugerencia
Curva de aprendizaje	
Diseño	No se reporta ninguna sugerencia

4.2.4.3 Lista de cambios priorizados

Tabla 21: *Lista de Tareas Iteración 3 Fase Beta*

Ítem	Descripción	Horas	Encargado
Quinta iteración			
1	Corregir errores de conexión con Facebook	4	JB

4.3 Fase Cierre y distribución

El juego esta subido en Google Store, para que pueda ser descargado de forma gratuita, en la siguiente tabla se presenta los pasos que se siguieron.

Tabla 22: *Pasos para subir una aplicación en Google Store*

Paso	Descripción
1	Crear una cuenta de desarrollador en: https://play.google.com/apps/publish/
2	Ir a la consola de desarrollador de Google Play

3	Seleccionar Tus Aplicaciones -> Añadir nueva aplicación
4	Escoger lenguaje para la aplicación, e ingresar un nombre para la misma.
5	Seleccionar subir APK, se escogerá el APK de la aplicación que se desea subir.
6	Seleccionar Prepara ficha de Play Store
7	Llenar información acerca de la aplicación, descripción breve y descripción completa.
8	Subir capturas de pantalla para 4 pulgadas,7 pulgadas y dispositivos de 10 pulgadas, así como un icono en alta resolución, imagen destacada y una imagen promocional, esto servirá para que los usuarios visualicen la aplicación en la tienda de Google.
9	Contestar preguntas y cuestionarios para asignarle una categoría y una clasificación adecuada al juego.
10	Ingresar datos de contacto
11	Publicar aplicación

En la figura 38 se presenta la aplicación en el mercado de Google Store,

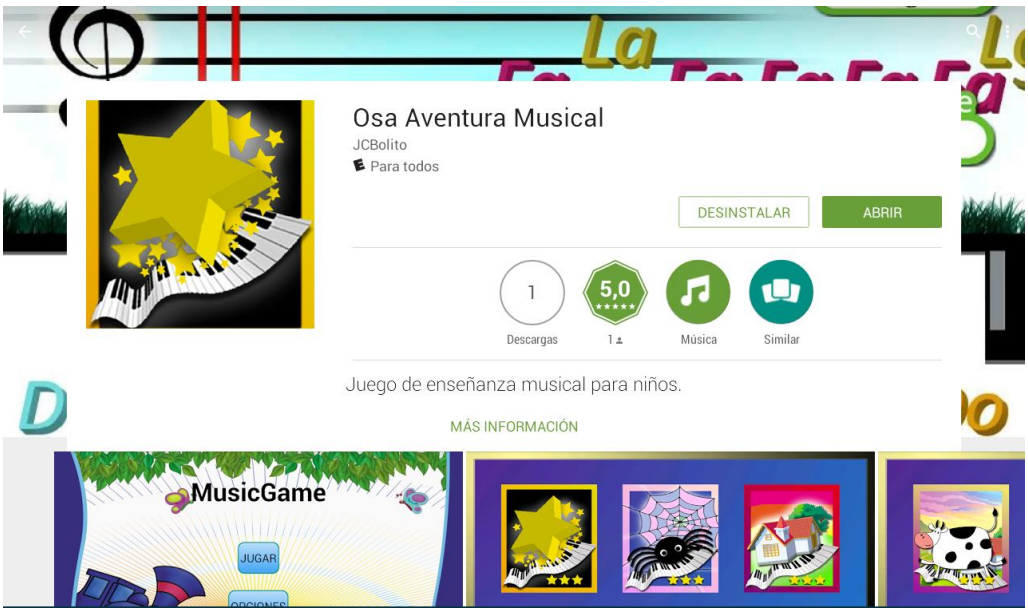


Figura 38: Google Store.

IMPACTO, CONCLUSIONES Y RECOMENDACIONES

5.1 IMPACTO EDUCATIVO

Para medir el impacto que ha tenido la aplicación en los niños, se ha realizado una evaluación de 6 preguntas que será realizada por un grupo de 25 estudiantes de la Unidad Educativa la Victoria de edades comprendidas entre 5 y 6 años. Los alumnos han usado la aplicación por un periodo de 8 días por un periodo mínimo de 10 minutos diarios. La prueba ha sido diseñada por el Lic. Jorge Luis Untuña docente de música en la Unidad Educativa la Victoria, de acuerdo a los objetivos planteados para la aplicación.

La evaluación consta de las siguientes preguntas:

1.-Cual es un pentagrama musical

a.-

b.-

c.-

2.- El orden las notas musicales es:

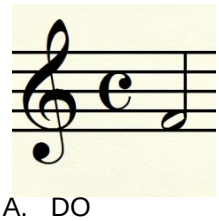
a) DO RE MI FA SOL LA SI DO
b) RE MI SOL FA LA SI DO
c) SOL MI LA FA SI DO RE
d) LA SI DO DO RE MI FA

3.- Según la ubicación en el pentagrama de la blanca que nota corresponde?

A. DO
B. RE
C. MI

D. FA

4.- Según la ubicación en el pentagrama de la blanca que nota corresponde?



A. DO
B. RE
C. MI
D. FA

5.- Según la ubicación en el pentagrama de la blanca que nota corresponde?

A. SOL
B. RE
C. DO
D. FA

6.- Según la ubicación en el pentagrama de la blanca que nota corresponde?

B. LA
C. SI
D. FA

5.1.1 RESULTADOS

La evaluación se calificó sobre 6 puntos, el resultado medio del grupo fue una calificación de 4.9 sobre 6, equivalente a 8.1 sobre 10.

En la siguiente tabla se puede observar cómo se sitúa esto de acuerdo al sistema de calificación del Ecuador.

Tabla 23: Sistema de calificación Ecuador

Calificación	Notas
10.00 - 9.00	Domina los aprendizajes requeridos
8.99 - 7.00	Alcanza los aprendizajes requeridos
6.99 - 5.00	Está próximo a alcanzar los aprendizajes requeridos
4.99 - 1.00	No alcanza los aprendizajes requeridos

En la siguiente figura se puede observar como los alumnos evaluados se sitúan de acuerdo al sistema de calificación de Ecuador.

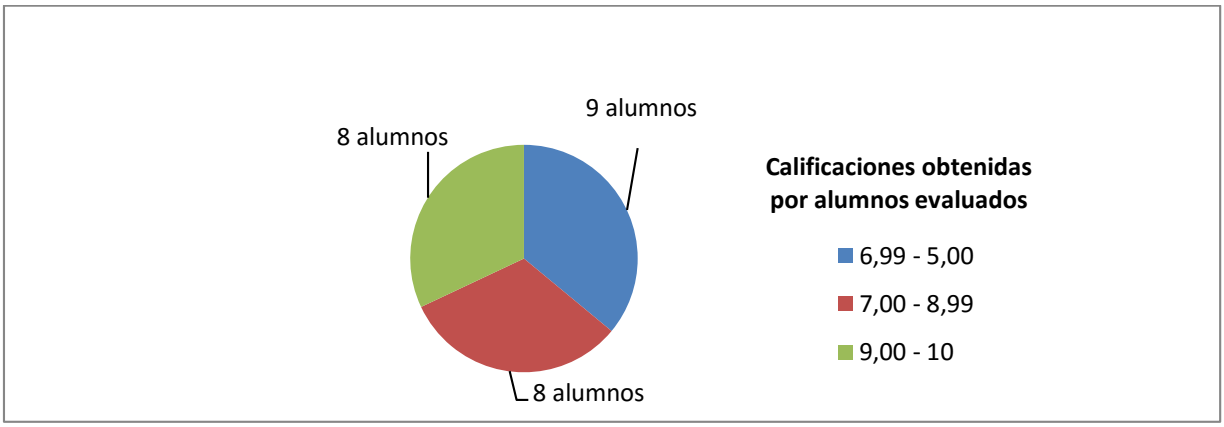


Figura 39: Desempeño alumnos evaluados.

En la figura 39 podemos apreciar que 16 alumnos correspondiente al 64% de los alumnos evaluados han conseguido adquirir todos los conocimientos propuestos con la implementación de la aplicación y 9 alumnos correspondiente al 36% restante está en camino a conseguirlos.

La aplicación ha tenido éxito brindando los conocimientos básicos que se ha propuesto a los niños evaluados, así como dando a conocer varias canciones infantiles, demostrando ser una buena herramienta para ayudar a introducir a los niños al mundo de la música. A continuación, se analizará en que preguntas tuvieron más dificultad los alumnos evaluados.

Tabla 24: Desempeño por pregunta

Evaluado	Primera Pregunta	Segunda Pregunta	Tercera Pregunta	Cuarta Pregunta	Quinta Pregunta	Sexta Pregunta	Puntaje	Puntaje Sobre 10
1	1	1	0	0	1	1	4	6,7
2	1	1	1	0	1	1	5	8,3
3	1	1	1	1	1	1	6	10,0

4	1	1	1	0	0	1	4	6,7
5	1	1	1	1	1	1	6	10,0
6	1	1	1	1	0	1	5	8,3
7	1	1	1	1	1	1	6	10,0
8	1	1	0	1	0	1	4	6,7
9	1	1	1	1	1	0	5	8,3
10	1	1	1	1	0	1	5	8,3
11	1	1	1	0	1	0	4	6,7
12	1	1	1	1	1	1	6	10,0
13	1	1	1	0	0	0	3	5,0
14	1	1	1	0	1	1	5	8,3
15	1	1	0	1	1	0	4	6,7
16	1	1	1	1	1	1	6	10,0
17	1	1	1	1	0	1	5	8,3
18	1	1	1	1	1	1	6	10,0
19	1	1	1	1	0	1	5	8,3
20	1	1	0	1	1	0	4	6,7
21	1	1	1	1	1	1	6	10,0
22	1	1	1	1	1	1	6	10,0
23	1	0	1	0	1	0	3	5,0
24	1	1	0	1	1	1	5	8,3
25	1	1	1	0	1	0	4	6,7
Total	25	24	20	17	18	18		
Puntaje sobre 10	10	9.6	8.0	6.8	7.2	7.2		

En las preguntas 1 y 2 referentes al pentagrama y los nombres de las notas casi la totalidad de los alumnos a respondido bien, con una puntuación cercana a 10; en las preguntas 3, 4, 5 y 6 que consisten en identificar los nombres de las notas en el pentagrama se puede observar que los alumnos han tenido una mayor dificultad con puntuaciones cercanas a 7, aunque esta puntuación es satisfactoria e indica que los niños evaluados si han aprendido lo requerido, identificar donde tienen problemas los niños servirá para implementar nuevas estrategias en futuras entregas de la aplicación para mejorar el aprendizaje.

5.2 CONCLUSIONES

- La metodología SUM para videojuegos se adapta fácilmente a equipos nuevos, provee mucha flexibilidad y anima a los equipos de trabajo que la usan a aprender de sus errores y adaptar la metodología a su forma de trabajo.
- La conformación de equipos interdisciplinarios recomendados en la metodología SUM permite delegar tareas de acuerdo a la capacidad y conocimientos, permitiendo desarrollar mejores aplicaciones.
- Android Studio, herramienta utilizada en el desarrollo de la aplicación de este proyecto y respaldada por Google, se ha consolidado en solo 2 años como una excelente herramienta de desarrollo de aplicaciones para dispositivos móviles con sistema operativo Android por ser gratuita, disponer de una amplia documentación y varios foros dedicados para ayudar a nuevos desarrolladores.
- Una alternativa eficiente para dar a conocer nuevos juegos es Facebook, una de las plataformas sociales más importantes del mundo, ofrece un conjunto de herramientas para que las aplicaciones puedan incluir un aspecto social, así como un sistema para crear cuentas a través de varias plataformas.
- Cuando los gustos del público objetivo son difíciles de cuantificar como en el caso de los niños, la fase de Verificación (BETA) es de gran importancia; durante el presente proyecto permitió realizar muchos cambios que solo fueron evidentes cuando niños usaron el juego.
- Los niños evaluados han podido exitosamente reconocer un pentagrama y aprender los nombres de las notas musicales, pero es necesario investigar más estrategias para poderles ayudar a reconocer el lugar de cada nota en el pentagrama.

5.3 RECOMENDACIONES

- Al ser el consumo de la memoria RAM una de las principales preocupaciones que tienen los desarrolladores de aplicaciones móviles, se recomienda un análisis minucioso de los recursos a incorporar, ya que una aplicación que use muchos recursos no podrá ser instalada en dispositivos antiguos, lo que limitará el número de usuarios que pueda tener.
- Debido a la preocupación de los usuarios de Facebook por compartir su información, se sugiere pedir solamente aquella indispensable para el mejor funcionamiento de la aplicación, ya que mientras más requisitos se soliciten, también serán más rigurosos los controles que Facebook pondrá en la aplicación antes de permitir que ésta pueda ser publicada.
- Considerando la relevancia que tiene la metodología y el equipo de trabajo en el diseño de un videojuego, se sugiere realizar un estudio cuidadoso que permita que la selección satisfaga los requerimientos del proyecto; la mejor estrategia es tener un equipo interdisciplinario, especialistas en sonido, video, programación y narración en caso de que el juego lo requiera.
- Para nuevos desarrolladores una buena estrategia es desarrollar juegos gratuitos, la venta de las aplicaciones no es la única manera de conseguir una remuneración, también se puede recibir ganancias incluyendo publicidad en las aplicaciones, o por medio de la venta de nuevos niveles para el juego, la venta de objetos únicos en el juego o cualquier otra estrategia que le dé al usuario una experiencia única a cambio de dinero.
- En el caso de juegos se puede ajustar ciertos factores en la etapa de pruebas, pero en el caso de juegos educativos no se puede ajustar la metodología por la cual se va a enseñar; se recomienda investigar y dedicar el tiempo necesario para encontrar la metodología adecuada para la enseñanza antes de realizar la lista de tareas final.

BIBLIOGRAFÍA

Mario Zechner, Robert Green. (2012). Beginning Android Games.New York: Apress.

Acerenza, Coppes, Mesa, Viera. (2009). Una Metodología Ágil para Desarrollo de Videojuegos. (Tesis de Ingeniería en Sistemas). Universidad de la República. Uruguay.

SOFTENG. (s.f.). Proceso y Roles de Scrum. Recuperado de: https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html.

ANDROID (s.f.). Introduction to Android. Recuperado de: http://developer.android.com/intl/es/guide/index.html

FACEBOOK (s.f.) Facebook SDK for Android. Recuperado de: https://developers.facebook.com/docs/android?locale=es_LA

GOOGLE (s.f.) Subir una Aplicación. Recuperado de: https://support.google.com/googleplay/android-developer/answer/113469?hl=es

Kent Beck and Cynthia Andres (2004). Extreme Programming Explained: Embrace Change (2nd Edition) EstadosUnidos: Addison-Wesley Professional.

HenrikKniberg (2007). Scrum y XP desde las trincheras. Como hacemos Scrum. EstadosUnidos: InfoQ Enterprise Software DevelopmentCommunity.

Frank Ableson, Charlie Collis and RobiSen (2010). Android: Guia para desarrolladores. México: Anaya Multimedia.

Derek James, (2012). Android Game Programming for dummies. Estados Unidos:Wiley&Sons.

James S. Cho, (2014). The Beginner's Guide to Android Game Development.EstadosUnidos: Glasnevin

John Horton, (2015). Android Game Programming by Example.Estados Unidos: Packt.

John Horton, (2015). Learning Java by Building Android Games - Explore Java Trough Mobile Game Development. Estados Unidos: Packt.

Miguel A. Posso, (2011). Proyectos, Tesis y Marco Lógico. Planes e Informes de Investigación. Ecuador: Noción.

ANEXOS

ANEXO A. MANUAL DE USUARIO

A.1 INDICACIONES GENERALES

Es un juego educativo pensado para niños, el cual trata de enseñar acerca del ritmo y de las notas musicales por medio de juegos musicales donde los niños puedan aprender mientras se divierten.

La aplicación es gratuita y se la puede descargar de Google Store buscando "Osa Aventura Musical".

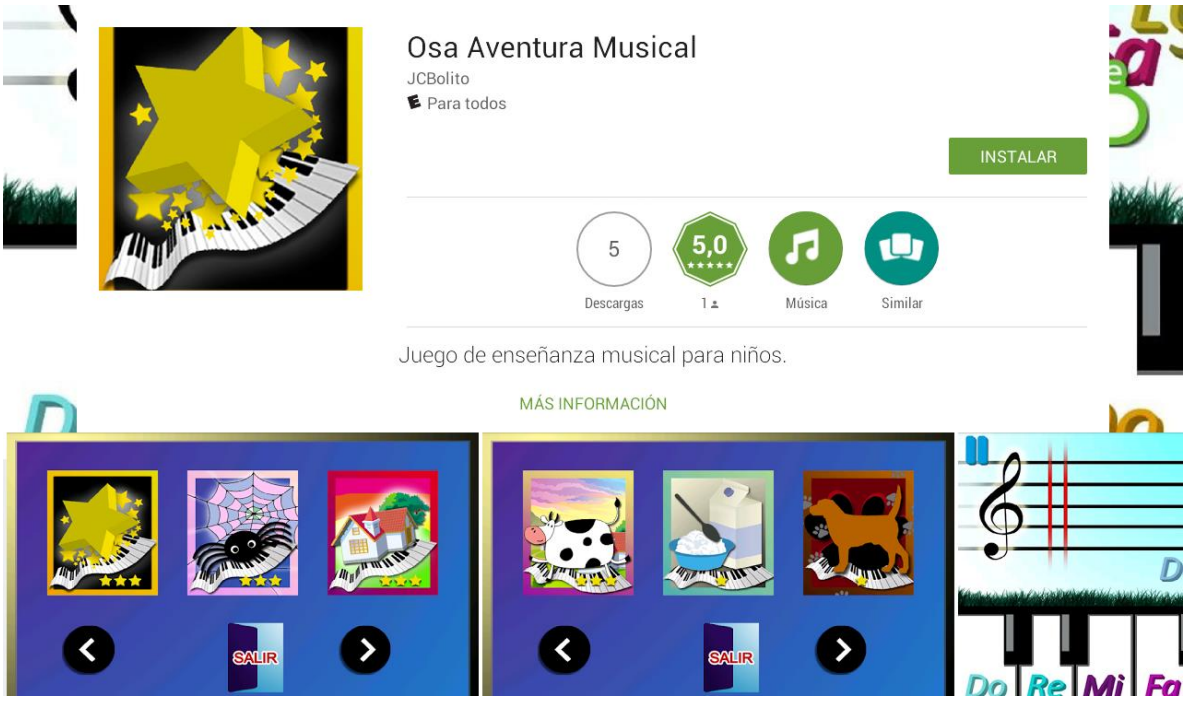


Figura 40: Aplicación disponible en Google Play: Osa Aventura Musical

Una vez se instale, se añadirá el siguiente icono, por medio del cual se podrá ingresar a la aplicación.

Figura 41: Icono aplicación: Osa Aventura Musical

A.2 DESCRIPCIÓN DE LA APLICACIÓN

La aplicación comenzará en la pantalla de inicio y a partir de allí el usuario podrá acceder y navegar por las siguientes pantallas.

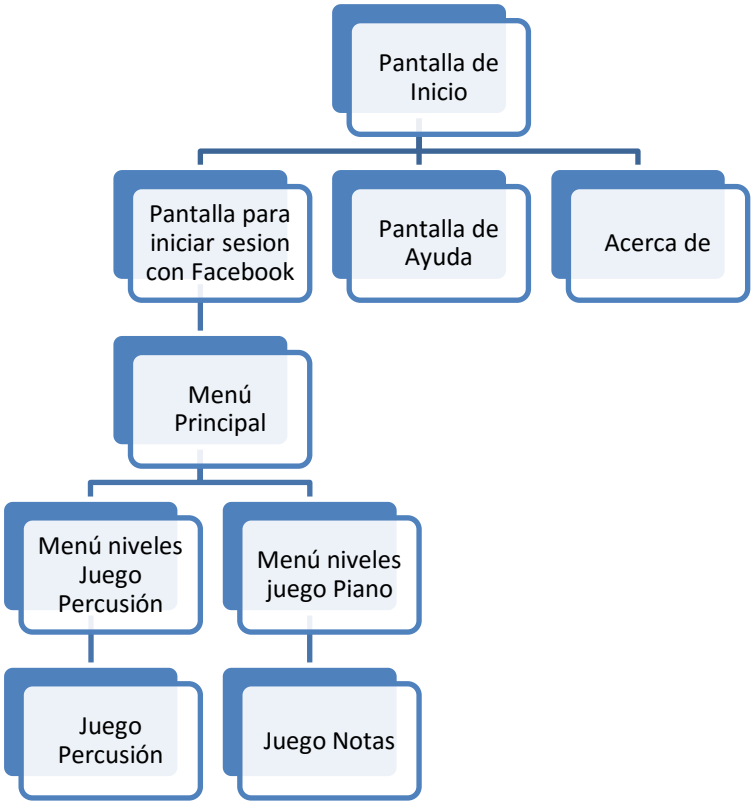


Figura 42: Pantallas de Juego: Osa Aventura Musical

A.2.1 Pantalla Inicio

En la pantalla de inicio hay 3 botones:

- **JUGAR:** Le permite al usuario conectarse con su cuenta de Facebook, en caso de ya estar conectado le permite al usuario acceder a Juego Percusión o a Juego Notas.
- **AYUDA:** El usuario podrá ver una pequeña ayuda para saber cómo jugar los diferentes niveles del juego.
- **ACERCA DE:** Se puede conocer acerca de las personas que colaboraron en la realización de esta aplicación.

Figura 43: Osa Aventura Musical: Pantalla Inicio

A.2.2 Pantalla Iniciar sesión con Facebook

Se presenta la opción de conectarse con Facebook o de jugar con una cuenta predeterminada.

Figura 44: Osa Aventura Musical: Pantalla Iniciar Sesión con Facebook

A.2.3 Pantalla Ayuda:

Da indicaciones para que el usuario conozca la aplicación.

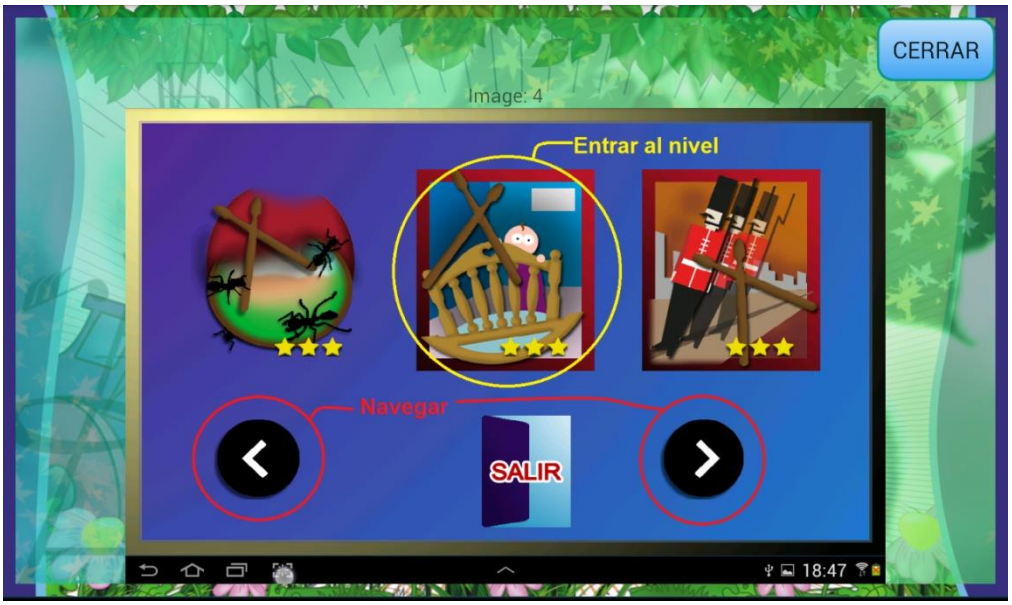


Figura 45: Osa Aventura Musical: Pantalla Ayuda

A.2.4 Pantalla Acerca De:

Permite conocer al usuario las personas que desarrollaron el juego.

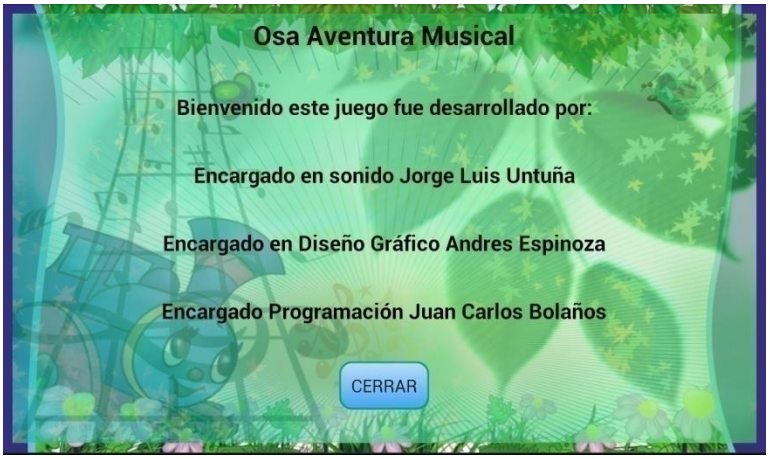


Figura 46: Osa Aventura Musical: Pantalla Acerca De

A.2.5 Pantalla Menú Principal

En este menú el usuario tiene 4 opciones:

- **CERRAR SESIÓN:** Permite al usuario desconectarse de su sesión de Facebook.

- **PERCUSIÓN:** Permite al usuario ingresar al menú de niveles del Juego Percusión.
- **NOTAS:** El usuario ingresará al menú de niveles del Juego Notas.
- **INICIO:** Simplemente permite al usuario regresar al inicio.

Figura 47: Osa Aventura Musical: Pantalla Menú Principal

A.2.6 Pantalla Menú Niveles Juego Percusión

El usuario podrá escoger que nivel jugar de los diferentes que tenga disponibles, comenzará solo con un nivel disponible, para conseguir nuevos niveles se debe acabar el nivel anterior con un porcentaje de aciertos mayor al 70%.

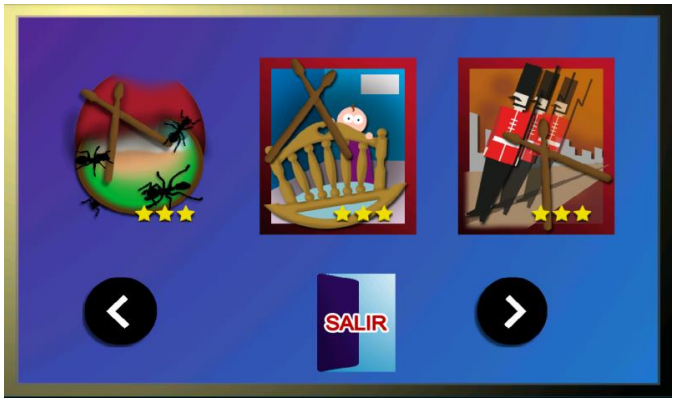


Figura 48: Osa Aventura Musical: Pantalla Menú Niveles Juego Percusión

A.2.7 Pantalla Menú Niveles Juego Notas

El usuario podrá escoger que nivel jugar de los diferentes que tenga disponibles, se comienza solo con un nivel disponible, para conseguir nuevos niveles se debe acabar el nivel anterior con un porcentaje de aciertos mayor al 70%.

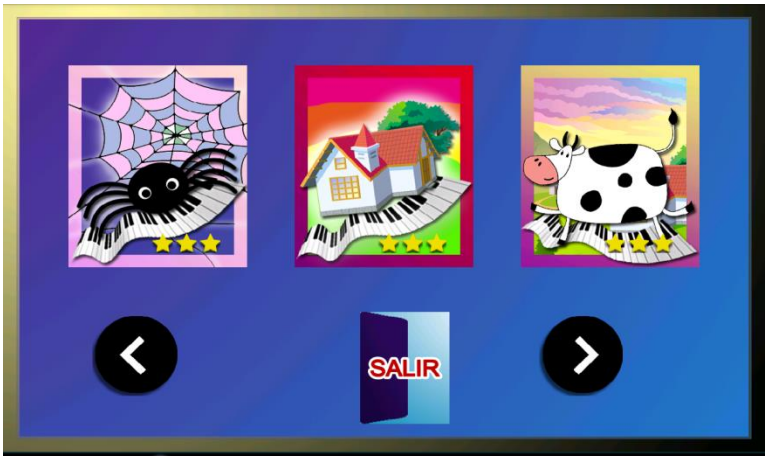


Figura 49: Osa Aventura Musical: Pantalla Menú Niveles Juego Notas

A.2.8 Pantalla Juego Percusión

El usuario dispondrá de 4 botones:

- **PAUSA:** En la esquina superior derecha se encuentra el botón de pausa, que permite pausar el juego en caso de que el usuario así lo decida.
- **TAMBOR:** Esquina inferior izquierda, se deberá tocar cuando su respectiva nota llegue a la línea azul.
- **PLATILLO:** Esquina inferior centro, reproduce el sonido de un platillo deberá ser tocado cuando su respectiva nota llegue a la línea azul.
- **BOMBO:** Esquina inferior derecha, deberá ser usado cuando su respectiva nota llegue a la línea azul.

Un porcentaje mayor a 70% indicará que se ha completado el nivel, el porcentaje se calcula en base a aciertos y errores.

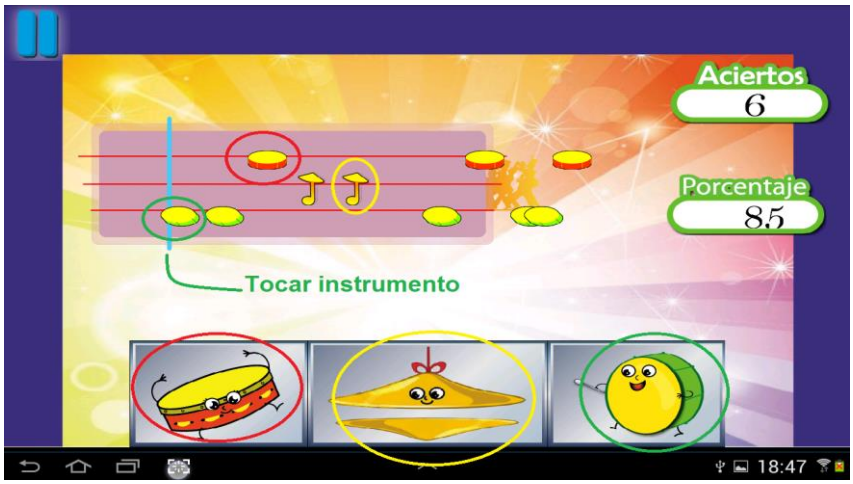


Figura 50: Osa Aventura Musical: Pantalla Juego Percusión

ANEXO B. PASOS PARA SUBIR UNA APLICACIÓN EN GOOGLE PLAY.

B.1 CREAR CUENTA GOOGLE PLAY DEVELOPERS

- Para comenzar se necesita una cuenta de correo de Google, si no se dispone de una se podrá crear una en la siguiente dirección **https://accounts.google.com/SignUp**

Figura 51: GMAIL: Seleccionar cuenta de correo

- El siguiente paso será entrar en la siguiente dirección **https://play.google.com/apps/publish/signup/**,seprocede a iniciar sesión con la cuenta de Google que se desee usar para subir aplicaciones, a continuación, se aceptará el Acuerdo para desarrolladores y se hará click en el botón "Continuar para completar el pago".

Figura 52: Google Play DeveloperConsole: Activar cuenta como desarrollador

- En el siguiente paso se debe realizar un pago de 25$, el pago debe realizarse con una tarjeta de crédito.

- En el paso final se deberá llenar información personal y de contacto, es muy importante llenar la información de una forma honesta, debido a que en caso de obtener dinero por la venta de aplicaciones, Google usará esta información para realizar los pagos.

B.2 AÑADIR UN ARCHIVO APK

- Una vez hecho el pago se ingresará a la consola, para subir la aplicación a la tienda se hará click en el botón "Añadir nueva aplicación".

Figura 53: Google Play DeveloperConsole: Añadir nueva Aplicación

- Se escoge el idioma, se da un nombre a la aplicación con el cual será visualizada en Google Store y se procede a hacer click en el botón "Subir APK".

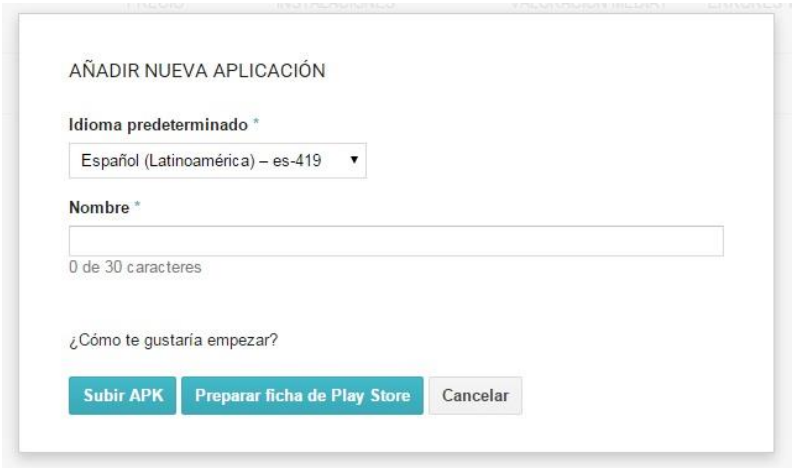


Figura 54: Google Play DeveloperConsole: Añadir nueva Aplicación

- En el caso de que la aplicación este en fase de prueba se puede elegir entre AlphaTesting o Beta Testing, si la aplicación esta lista para ingresar a distribución se deberá elegir Producción, se hará click en la pestaña correspondiente y después click en el botón "Subir APK".

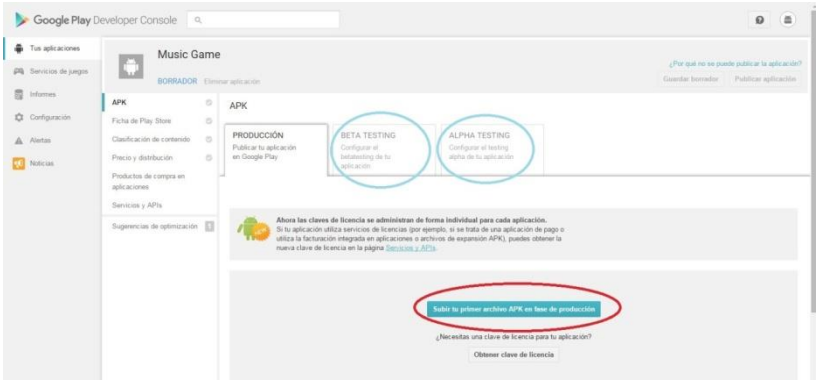


Figura 55: Google Play DeveloperConsole: Subir primer archivo APK.

- Si todavía no se ha generado el archivo APK, se procederá a generar uno en Android Studio, la opción se encuentra en Build ->GenerateSignedAPK, si es la primera vez se solicitará información del desarrollador para crear una llave que usará en la creación de sus APK's; el archivo APK se generará en la carpeta del proyecto dentro de la subcarpeta build.

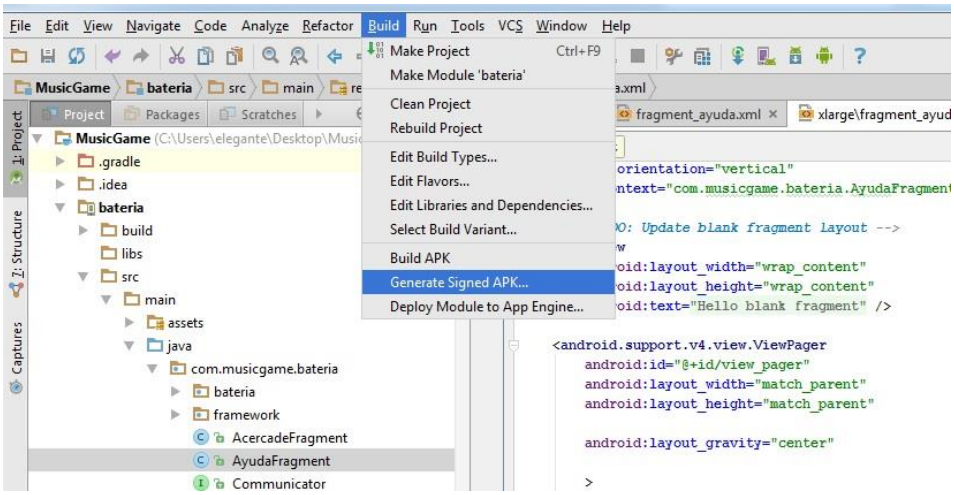


Figura 56: Generate Signed APK.

B.3 LLENAR FICHA PLAYSTORE

- El siguiente paso es llenar la ficha de Play Store, los campos que tengan una estrella azul en su parte superior son obligatorios, pero lo más recomendable es completar la ficha esto mejorará la presentación de la aplicación en la tienda de Google y permitirá que más gente la vea y pueda descargarla. Se debe asegurar de llenar el cuestionario de Contenido, en caso de no hacerlo se puede llenarlo después, está bajo la ficha de Play Store.

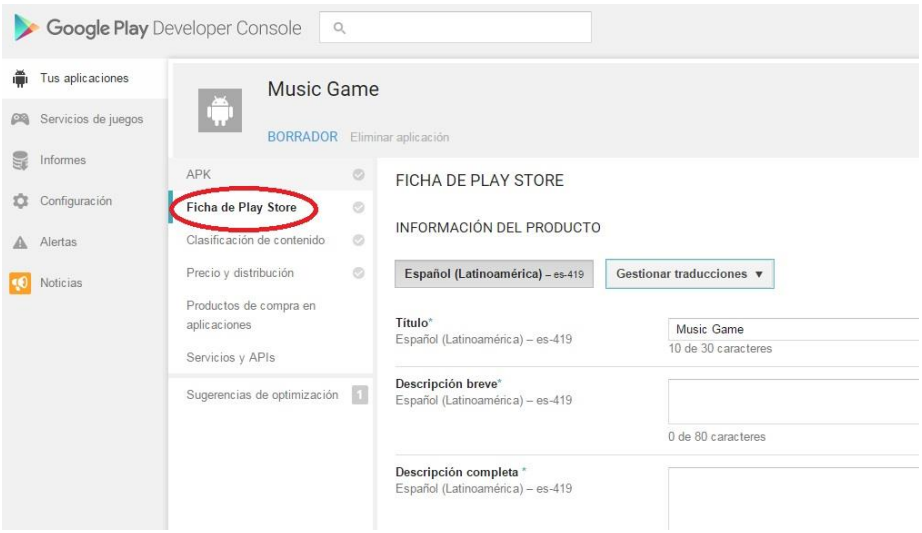


Figura 57: Google Play Developer Console: Ficha de Play Store

- Se deberá esperar un par de horas para que el cuestionario de Clasificación de contenido sea aprobado, después de eso se podrá publicar la aplicación desde la pestaña APK.

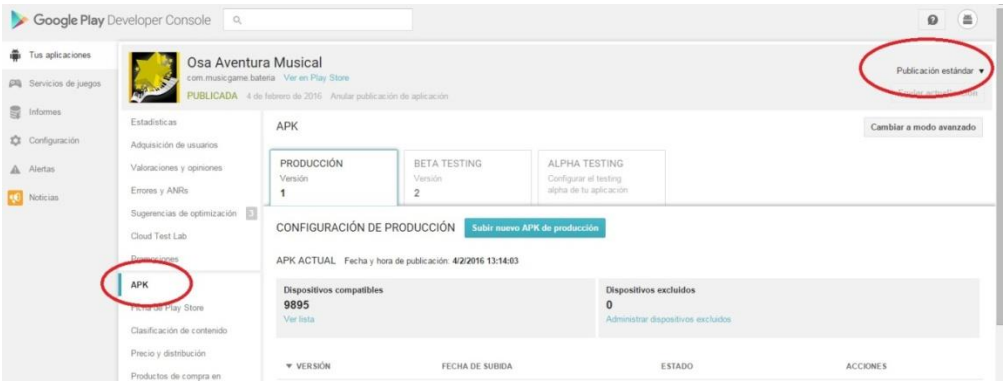


Figura 58: Google Play Developer Console: PublicarAplicación

- Si se escogió Beta Testing o Alfa Testing, es momento de añadir verificadores que puedan acceder a la aplicación. La aplicación estará en Google Play pero solo será visible para gente que sea aprobada por el desarrollador, para hacer esto se hará click en "CreateList", y se añadirá a los verificadores con nombre y correo, el correo debe estar vinculado a una cuenta en Google Store.

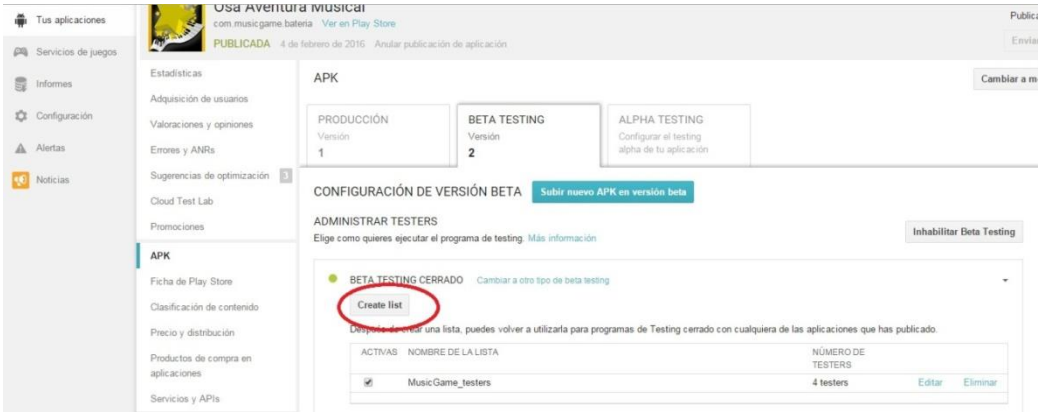


Figura 59: Google Play DeveloperConsole: Añadir Beta Testers