

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

# METROLLÍN

**Una aplicación Interactiva de entretenimiento enfocado a pasajeros en un sistema de tránsito rápido para Android en 2D, elaborada con software libre**

Juan Fernando Cañola García

Edwin Daniel Álvarez Vásquez

Ingeniería de Sistemas

Asesorado por:

Delio Augusto Aristizabal Martínez

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**Febrero 13 de 2017**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RESUMEN

---

El presente documento describe el proceso, recursos, diseño y software utilizados para la creación de un sistema de entretenimiento para dispositivo móvil llamado Metrollín.

Metrollín es un sistema de información de entretenimiento gráfico en 2D, desarrollado e integrado por herramientas de software libre, para teléfonos inteligentes y tabletas con sistema operativo Android 4.0 (*Ice Cream Sandwich*) o superior. El objetivo de la aplicación de entretenimiento es que los usuarios vayan ingresando y sacando pasajeros de los vagones, en cada una de las 19 plataformas de las estaciones del metro por medio de la pantalla táctil. Cada pasajero tiene sus características propias, entre ellas, color, velocidad y movimiento. Para superar las estaciones, el usuario debe colocar los pasajeros en la plataforma y el vagón correspondiente de acuerdo a su color. Cada estación tiene un límite de tiempo para el ingreso y la salida de los pasajeros. El nivel de dificultad va aumentando entre las diferentes estaciones, con diferentes retos en 7 niveles diferentes. De esta manera se sumará puntaje por cada una de los pasajeros ubicados en la plataforma y en los vagones correspondientes.

La primera parte del documento contiene el marco teórico donde se define que es Android, su historia, su arquitectura, sus librerías, sus versiones, seguridad y toda la información necesaria para el desarrollo de una aplicación móvil en esta plataforma.

La segunda parte contiene los diagramas, la arquitectura de la aplicación, el diseño de imágenes, personajes, sonido y todos los recursos utilizados para el funcionamiento de la aplicación.

La tercera parte muestra la aplicación final desarrollada alojada en la Play Store, donde puede ser descargada por cualquier persona con sistema operativo Android 4.0 (*Ice Cream Sandwich*) o superior.

Finalmente el documento muestra las recomendaciones para quienes deseen desarrollar una aplicación móvil con software libre, y el futuro trabajo para las actualizaciones del sistema de entretenimiento Metrollín.

**Palabras clave:** Video Juego, Android, Software Libre, Metro, Sistema Entretenimiento, 2D

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# RECONOCIMIENTOS

---

Y finalmente llegó el día. El día de presentar mi proyecto de grado de Ingeniería de Sistemas, y quiero agradecer a todas las personas que me han apoyado durante todo este proceso.

Primero empezaré por agradecer a mis docentes, quienes día a día trabajaban duramente para compartir su conocimiento conmigo y el resto de mis compañeros. Fue en el año 2010 cuando decidí estudiar la carrera de Sistemas en el ITM, y ese mismo año por recomendación de mi maravillosa madre, empecé a estudiar también un curso de Mantenimiento y Reparación de computadores en Comfama. Allí en una de tantas clases en la mañana, conocí a la profesora **Doreley Garcia Orrego**, quien más que una profesora parecía una amiga, porque recuerdo mucho que me motivaba a salir adelante en mi carrera de Sistemas, y me decía constantemente como marcaba la diferencia frente a mis compañeros por mi inteligencia, liderazgo, trabajo en equipo y realización de diferentes actividades. Hoy en día es mi amiga en facebook y siempre me sorprende verla sonriente en las fotos que sube de todos sus viajes.

Ese mismo año en el ITM en mi primer semestre de sistemas conocí a los primeros docentes quienes me darían las bases en mi carrera de Sistemas. Ellos son **Rafael Reyes Moreno, Diego de Jesús Tangarife Chalarca, Mauricio Alberto Correa Villa y Juan Camilo Giraldo Mejía**, a quienes recuerdo por qué les hacía muchas preguntas, ya que apenas estaba aprendiendo y entendiendo el mundo de los sistemas de información. Recuerdo mucho que al final de semestre hice mi primer video juego en java, que consistía en una lotería, vaya manera de empezar. En el segundo semestre de sistemas conocí a los docentes **Carlos Mario Soto Jaramillo** que en paz descansa, quien me enseñó la materia estructura de datos, y donde fui aprendiendo a conocer más sobre los lenguajes de programación orientado a objetos. Y una maravillosa profesora de Inglés **Fanny Andrea Gordillo Arboleda**, con quien me llevaba muy bien porque enseñaba inglés de una manera muy armoniosa, y de quien aprendí mucho vocabulario y tiempos en inglés.

En el año 2011 ya terminando mi curso en Comfama, realice mis prácticas en el centro Colombo Americano donde tres personas me tendieron sus manos. Ellos son **Alex, Oscar y Juan**, quienes fueron mis jefes para mi práctica de mantenimiento y reparación de computadores, en donde aprendí mucho del mundo empresarial de los sistemas. En mi cuarto semestre de sistemas conocí al docente **Delio Augusto Aristizabal Martínez**, quien fue mi primer docente en enseñarme sobre el desarrollo de software empresarial, al integrar y conectar un programa desarrollado en un lenguaje de programación de alto nivel con una base de datos, lo cual abrió mi mente porque me dio la idea de cómo se conectaban varias

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

tecnologías en un solo producto software. Ese mismo semestre conocí a **Carlos Arturo Castro Castro**, quien fue mi docente de programación para la web utilizando una tecnología de java llamada JSP, en donde complementó mi conocimiento dictado por el profesor Delio, ya que en una materia utilizaba tecnología propietaria de Microsoft: .net con lenguaje de programación C# y base de datos SQL Server, y en la otra materia utilizaba software libre: IDE netbeans, con lenguaje de programación java y base de datos mysql, y en donde además conocía más y más tecnologías para el desarrollo de software como HTML, CSS y XML.

En el 2012 y en mi quinto semestre conocí al docente **Diego Alejandro Guerrero Peña**, quien me enseñó la parte administrativa del mundo de los sistemas de información, lo cual no me llamó mucho la atención porque lo mío definitivamente es programar, pero a quien siempre recordaré por su frase “*el que se emociona pierde*”. También conocí a un profesor llamado **Jesús Albeiro Zuluaga Duque**, quien me enseñó un nuevo lenguaje de programación para la web llamado PHP, el cual le agradezco mucho porque me ha servido para ganar dinero extra como freelance desarrollando páginas web.

Ese mismo año tuve la maravillosa experiencia de hacer mis prácticas de tecnología de sistemas en una empresa llamada **Ingeneo S.A**, en donde trabaje por un año y aprendí sobre el desarrollo de software empresarial utilizando el lenguaje de programación java y utilizando tecnologías que no había visto en mi carrera. Por lo cual me siento muy orgulloso de haber participado y haber aportado al crecimiento de esta empresa. Finalizado mi periodo como analista desarrollador en Ingeneo S.A en el 2013, regrese al ITM para finalizar mi carrera de sistemas, ya como Ingeniero. Ese semestre conocí dos maravillosos docentes. Uno se llamaba **Alberto Mauricio Arias Correa**, mi docente de circuitos eléctricos y de quien me encanta su exigencia para hacer las cosas, porque aprendí demasiado sobre el lenguaje de bajo nivel de una máquina, procesadores, microcontroladores, y varios componentes de la electrónica. El otro docente es **Juan Carlos Zuluaga Correa**, quien me enseñó aún más sobre el desarrollo de software empresarial, y quien me dio las bases para el desarrollo de aplicaciones móviles, aunque en Windows Phone. En el año 2015, entre una de tantas materias no relacionadas con los sistemas de información conocí al docente **Carlos Orlando Zapata Garcia**, quien me dictó la materia llamada Introducción a la investigación, y de quien recuerdo mucho su educación, consejos y forma de comunicarse con las demás personas.

En el segundo semestre del año 2015 entre a hacer parte del programa jóvenes investigadores del ITM, gracias a la convocatoria realizada por **Jackeline Andrea Macias Urrego**, en donde me presente al grupo de investigación Innovación en Matemáticas y nuevas tecnologías para la educación GNOMON, donde conocí a mis tutores **Héctor Javier Herrera Mejía**, **Elkin Alberto Castrillón Jiménez** y **Carlos Mario Restrepo Restrepo**, con

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

quienes disfrute mucho y aprendí demasiado sobre el desarrollo de objetos virtuales de aprendizaje, y con quienes más que una relación tutor-joven investigador, me hicieron sentir como un docente de su grupo de investigación. Y finalmente un agradecimiento muy especial al ITM, quien además de abrirme las puertas del conocimiento, me permitió participar en diferentes convocatorias que me permitieron aprender y conocer a muchas más personas.

Ahora voy a agradecer a mis maravillosos amigos, las personas con quienes he compartido incalculables momentos en toda mi vida, y han sido uno de mis motores para salir adelante en mi vida personal y profesional. Primero mis amigos del colegio ***Felipe Restrepo Nohavá, Oscar David Martínez, y Eric Steven Parra***, con quienes aún disfruto conversando, jugando videojuegos y asistiendo a eventos familiares, y con quienes nunca he perdido contacto, aunque la vida nos separe temporalmente. Gracias a mis amigos del colegio, conocí otras maravillosas personas y todo gracias a un videojuego en línea llamado Ragnarok Online. En este juego pasábamos horas conociendo ciudades y calabozos, compitiendo con otros clanes, sumergidos día y noche en un juego virtual. Tal fue la experiencia de conocernos, que aún disfrutamos bebiendo y jugando, y apoyándonos para salir adelante. Ellos son ***Juan José Jiménez, Sebastián Córdoba, Johny Roldan, Jorge Pérez, David Palacio, Pablo Palacio y Jairo Saldarriaga***.

Una mención muy especial a mis dos queridas amigas del curso de Inglés de la universidad de Antioquia ***Elizabeth Andrea Tabares y María Isabel Sosa***, con quienes reí y disfrute mucho, conociendo diferentes lugares de Medellín y con quienes espero algún día reencontrarme para agradecerles por tan bellos momentos.

Ahora un agradecimiento muy especial a mis amigos de la universidad, con quienes he trasnochado aprendiendo, discutiendo y creando software que es lo que nos apasiona. Ellos son ***Mauricio López***, a quien conozco desde mi primer semestre de sistemas, ***Julián Vásquez, Edwin Daniel Álvarez, Mateo Guerra, Juan Castaño, Tatiana González, Ana María Ospina, Wilson Garcia, Verónica Flores, Marlon Tamayo, David Ortiz, Mile Orrego y Laura*** a quienes conocí en el transcurso de mi carrera de Sistemas, y quienes me enseñaron más que de Sistemas, porque gozaba escuchar las historias de su vida. Un agradecimiento a mis amigos y compañeros que no mencione aquí, porque aunque tengo pocos recuerdos, sé que estuvieron conmigo en diferentes momentos de mi vida.

Ahora quiero agradecer a mis amigos y compañeros de videojuego, ya que últimamente son con quienes interactuó más y quienes me alegran las mañanas con sus memes y comentarios graciosos. Ellos son los spamfriends ***Diego, Daniela, Alejandro, Elkin, Mauricio, Gian, Kchu, Michael, Leo, Bras, Pedro, Hugo, Sebastián, Juan David, David y Johanna Polli***, mis amigos de clan *New Kraken Team Colombia* ***Alexis Pacheco, Logan Quiroga, Camilo Ortegón y Brayan Alvarado***, de Perú ***David Fuentes***, de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Argentina **Christian Walsh**, y **Gonzalo Garcia**, y de Uruguay **César Corrales** y **Arturito Pérez**, los cuales hicieron parte de mi vida al conformar un clan para un videojuego llamado Splatoon, al enfrentar equipos de diferentes países y que hoy en día somos más que un clan, somos amigos.

Y ahora antes de mencionar a mis familiares, debo mencionar a la persona más importante en mi vida y quien me ha apoyado en las buenas y en las malas, quien me ha visto trasnochar estudiando y jugando, y con quien nos ayudamos mutuamente para cumplir nuestro sueño. Mi novia hermosa, bella, preciosa a quien le tengo más de 20 maravillosos adjetivos y cada día más **Joan Andrea Vanegas Alba** ♥. Ella es mi vida y mi mayor fortaleza para salir adelante y con quien deseo compartir el resto de mis días hasta la eternidad. Te debo esta vida y la otra, porque llegaste cuando menos lo esperaba y cuando más te necesitaba, y no tengo palabras para expresar cuanto me has cuidado, ayudado y depositado tu confianza en mí. Simplemente te amo y siempre lo haré.

Ahora quiero agradecer a mis primos que siempre me han hecho reír, y quienes me han cuidado desde que tengo memoria. A ellos les debo tanta felicidad **Julián, Mónica, Diego, Eliana, Mauricio, Liliana, Johanna, Marcela, Stephanie, Juan David** y **María Alejandra**. Son incontables las reuniones familiares que hemos tenido y que espero sigamos teniendo juntos.

Y por último a mi maravillosa familia. Mi hermano **Andrés Felipe Cañola Garcia**, con quien disfruto jugando y viendo partidos de fútbol y **María Isabel Cañola Garcia** quien me ayuda a cuidar de mi salud y yo le ayudo a cuidar la salud de su computador. Con ellos rio, comparto, discuto y nos ayudamos para salir adelante. A mi padre **Ramón Abad Cañola Argaez** quien siempre está pendiente de nosotros su familia, y a quien recuerdo por su espíritu de ayuda y siempre al servicio de los demás. Y por último, la persona más pero más importante en mi vida y la causa de que yo este con vida y con buena salud, mi madre hermosa **Sonia María Garcia Sánchez**, a quien le debo todo en la vida y que no tengo como pagarle el esfuerzo y sacrificio que ha hecho día a día. Siempre recordaré como madrugaba conmigo para atenderme, sin importar cuanto tiempo se hubiera trasnochado por su trabajo como enfermera.

A todos, desde el fondo de mi corazón, ¡gracias por todo!

*Juan Fernando Cañola Garcia*

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Es de gran felicidad y satisfacción hoy finalizar una etapa más de estudio, cumplir un sueño, ver el fruto de un gran esfuerzo, por ello quiero agradecer primordialmente a Dios por darnos la inteligencia, sabiduría, paciencia, entendimiento y la capacidad para ejercer este proyecto.

A mi madre Luz Vásquez por todo su apoyo, comprensión y confianza. Mi padre que sin importar su ausencia desde muy pequeño motivo siempre a estudiar, a progresar con el estudio, trabajo y esfuerzo. A nuestros hermanos Elenit Álvarez, Yamile Álvarez, Verónica Álvarez, Arley Álvarez, Dubier Álvarez y Ferney Álvarez que me ayudaron, motivado y orientado tantas veces.

Agradecer el acompañamiento del profesor Delio Aristizabal en nuestras asesorías, acompañamiento en el proceso, su conocimiento y su tiempo dedicado. A nuestros compañeros de clase Mauricio, Julián, con los siempre trabajamos y que hemos compartido conocimientos. Agradecer a Juan Cañola por su compromiso, acompañamiento en el proceso, motivación, conocimiento, esfuerzo, trabajo en equipo y paciencia en el trabajo realizado.

Familiares cercanos, personas que directa o indirectamente aportaron en la motivación y compartieron experiencias de aprendizaje para reflexionar y a aprender, especialmente: Elkin Montoya, Juan Carlos, Carlos Rodríguez, Sandra González, Duviel Díaz, Mauricio Vásquez, Ricardo Buitrago, Russ Fisher Ives, Keegan Fisher. A nuestros amigos y amigas por su apoyo, guianza, conocimientos, ánimos, Claudia Ceballos, Faicely Gutierrez, Natalia, Heidy, Alejandro Gómez, Yul Usuga, Carlos Gutierrez, Luis Vásquez, por estar motivando, siempre han sido un gran apoyo y un punto de referencia, por su paciencia, ayuda y motivación para terminar este proyecto.

Agradecer a la universidad por todos los espacios brindados, los procesos de aprendizaje realizados y los profesores que siempre estuvieron apoyando. Por las oportunidades entregadas en cuanto a todo el mundo del conocimiento.

*Edwin Daniel Álvarez Vásquez*

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# ACRÓNIMOS

---

**ART:** Android Run Time o Tiempo de ejecución Android

**API:** Application Programming Interface o Interfaz de programación de aplicaciones

**APK:** Android Application Package o Paquete de aplicación Android

**BSD:** Berkeley Software Distribution o Distribución de Software Berkeley

**DVM:** Dalvik Virtual Machine o Máquina Virtual Dalvik

**GPL:** General Public License o Licencia Pública General de GNU

**GPS:** Global Positioning System o Sistema de posicionamiento Global

**HTTP:** Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto

**JSON:** JavaScript Object Notation u Objeto de Notación JavaScript

**JIT:** Just-in-time o Justo a tiempo

**JRE:** Java Runtime Environment o ambiente de tiempo de ejecución Java

**MAC:** Media Access Control address o dirección de control de acceso al medio

**MTP:** Media Transfer Protocol o Protocolo de Transferencia de Medios

**NDK:** Android Native Development Kit o Equipo nativo de Android

**OHA:** Open Handset Alliance o Alianza de móvil abierta

**PTP:** Picture Transfer Protocol o protocolo de transferencia de imágenes.

**REST:** Representational State Transfer o Transferencia de estado representacional

**SOAP:** Simple Object Access Protocol o protocolo de acceso a objeto simple

**S.O:** Sistema Operativo

**SD:** Secure Digital o Tarjeta de memoria de seguro Digital

**SDK:** Software Development Kit o Equipo de desarrollo de software

**SQL:** Structured Query Language o Lenguaje de Consulta Estructurado

**USB:** Universal Serial Bus o Bus Universal en Serie

**XML:** Extensible Markup Language o Lenguaje de Marcado extensible

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# TABLA DE CONTENIDO

---

1. INTRODUCCIÓN .....	15
Objetivo General.....	15
Objetivos Específicos.....	15
2. MARCO TEÓRICO.....	17
2.1 Android.....	17
2.1.1 ¿Qué hace que Android sea especial?.....	17
2.1.2 Orígenes .....	18
2.1.3 Comparativa con otras plataformas .....	20
2.1.4 Arquitectura de Android .....	21
2.1.5 Las versiones de Android y niveles de API.....	24
2.1.6 Componentes de una aplicación.....	31
2.1.7 Ciclo de vida de una actividad .....	32
2.1.8 Base de datos en Android .....	35
2.1.9 Los tres pilares de la seguridad en Android.....	36
2.1.10 Usuario Linux y acceso a archivos.....	36
2.1.11 El esquema de permisos en Android .....	37
2.1.12 Licencias aplicaciones Android.....	37
3. METODOLOGÍA.....	38
3.1 Análisis de la aplicación.....	38
3.1.1 Especificaciones de la aplicación.....	38
3.1.1.1 Descripción del sistema de información .....	38
3.1.1.2 Características del sistema .....	38
3.1.2 Recursos utilizados para el desarrollo .....	40
3.1.2.1 Recurso Humano .....	40
3.1.2.2 Recursos de Hardware .....	40
3.1.2.3 Recursos de Software.....	40
3.1.3 Consultas.....	41
3.1.3.1 Herramientas de software libre utilizadas .....	41
3.1.3.2 Instalación de herramientas de software libre utilizadas .....	42
3.1.3.3 Subir aplicaciones a la Play Store.....	43
3.1.3.3.1 Obtener una cuenta de programador .....	43
3.1.3.3.2 Subir APK a la Play Store .....	43
3.1.3.3.3 Requerimientos para subir una aplicación a la Play Store .....	43

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.2 Diseño .....	45
3.2.1 Diagrama de contexto .....	45
3.2.2 Diagrama de Clases .....	46
3.2.3 Casos de Uso .....	48
3.2.3.1 Ingresar a la aplicación .....	48
3.2.3.2 Presionar Jugar .....	49
3.2.3.3 Seleccionar Estación .....	50
3.2.3.4 Interactuar con el juego .....	51
3.2.3.5 Presionar Instrucciones .....	53
3.2.3.6 Presionar Configuración.....	54
3.2.3.7 Presionar Puntuación .....	55
3.2.3.8 Presionar Acerca De .....	56
3.2.4 Diagrama de casos de uso .....	57
3.2.5 Arquitectura de la aplicación .....	58
3.2.5.1 Definición Método Fábrica .....	58
3.2.5.2 Método implementado en la aplicación .....	59
3.2.6 Diseño de Layouts .....	60
3.2.6.1 Pantalla principal .....	60
3.2.6.2 Pantalla Jugar.....	61
3.2.6.3 Pantalla Juego Estación.....	62
3.2.6.4 Pantalla Instrucciones .....	64
3.2.6.5 Pantalla configuración.....	65
3.2.6.6 Pantalla Puntuaciones .....	66
3.2.6.7 Pantalla Acerca de .....	67
3.2.6.8 Pantalla Transición .....	68
3.2.7 Diseño de imágenes .....	68
3.2.7.1 Icono .....	69
3.2.7.2 Botones.....	69
3.2.7.2.1 Botones del menú principal .....	69
3.2.7.2.2 Botón volver .....	70
3.2.7.2.3 Botón sonido .....	70
3.2.7.2.4 Botón estación.....	70
3.2.7.3 Personajes.....	71
3.2.7.3.1 Personaje Man.....	71
3.2.7.3.2 Personaje Woman .....	71
3.2.7.3.3 Personaje Child .....	72
3.2.7.4 Otros Objetos.....	72
3.2.7.4.1 Zonas .....	72
3.2.7.4.2 PlusOne.....	72

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.2.7.4.3 Error .....	73
3.2.7.4.4 Problema eléctrico.....	73
3.2.8 Efectos de sonido.....	73
3.2.8.1 Música .....	74
3.2.8.2 Efectos de sonido .....	74
3.2.9 Diseño de Bases de Datos .....	75
3.3 Desarrollo, Integración y Publicación .....	76
3.3.1 Estructura final del proyecto .....	76
3.3.2 Layouts.....	77
3.3.3 Imágenes sistema de entretenimiento finalizado.....	78
3.4 Servicios Web .....	83
3.4.1 Web Service ingresarUsuario.....	83
3.4.2 Web Service generarUsuarios .....	83
3.6 Aplicación en la Play Store.....	84
3.6.1 Subir aplicación a la tienda de Apps Play Store.....	84
3.6.2 Pruebas a la aplicación en la Play Store .....	86
3.6.3 Actualización y mantenimiento.....	86
3.6.4 Anuncio y divulgación de la aplicación.....	86
4. RESULTADOS Y DISCUSIÓN .....	87
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO .....	90
REFERENCIAS .....	93
APÉNDICE .....	94
Código del método fábrica aplicación Metrollín .....	94

 ITM Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# TABLA DE IMÁGENES

Figura 1. Arquitectura de Android.....	21
Figura 2. Ciclo de vida de una actividad Android.....	34
Figura 3. Ejemplo de permisos en el Android Manifest .....	37
Figura 4. Diagrama de Contexto.....	45
Figura 6. Diagrama de casos de uso.....	57
Figura 7. Diagrama UML del método Fábrica .....	58
Figura 8. Diagrama UML del método Fábrica implementado .....	59
Figura 9. Menú principal propuesto.....	60
Figura 10. Menú principal final .....	61
Figura 11. Selección de estaciones propuesto .....	61
Figura 12. Selección de estaciones final .....	62
Figura 13. Pantalla Estación Propuesto.....	62
Figura 14. Pantalla Estación Final .....	63
Figura 15. Imagen con las instrucciones .....	64
Figura 16. Pantalla Configuración final .....	65
Figura 17. Pantalla Puntuaciones final.....	66
Figura 18. Pantalla Acerca De propuesto.....	67
Figura 19. Pantalla Acerca De Final .....	67
Figura 20. Pantalla transición de vagones .....	68
Figura 21. Interfaz gráfica de GIMP.....	68
Figura 22. Icono de la aplicación propuesto.....	69
Figura 23. Icono de la aplicación final.....	69
Figura 24. Botones del menú principal .....	69
Figura 25. Imagen botón volver.....	70
Figura 26. Imagen Botón activar y desactivar sonido.....	70
Figura 27. Imagen botón animado selección de estación.....	70
Figura 28. Drawables Personaje Man .....	71
Figura 29. Drawables Personaje Woman .....	71
Figura 30. Drawables Personaje Child.....	72
Figura 31. Drawable Zona.....	72
Figura 32. Gif Animado Plus One .....	72
Figura 33. Gif Animado Error .....	73
Figura 34. Imagen Falla Eléctrica.....	73
Figura 35. Interfaz gráfica Linux Multimedia Studio .....	73
Figura 36. Imagen Archivos de música.....	74
Figura 37. Imagen archivos efectos de sonido .....	74

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 38. Entidad Puntaje .....	75
Figura 39. Diagrama de clases final.....	77
Figura 40. Lista de layouts final .....	77
Figura 41. Pantalla selección estación final .....	78
Figura 42. Pantalla estación final 1 .....	78
Figura 43. Pantalla estación final 2.....	79
Figura 44. Pantalla estación final 3.....	79
Figura 45. Pantalla estación final 4.....	80
Figura 46. Pantalla estación final 5.....	80
Figura 47. Pantalla estación final 6.....	81
Figura 48. Pantalla estación final 7.....	81
Figura 49. Pantalla estación final 8.....	82
Figura 50. Pantalla estación final 9.....	82
Figura 51. Opción para generar el APK.....	84
Figura 52. Asistente para la generación del APK.....	84
Figura 53. Apk generado en el explorador .....	85
Figura 54. Interfaz gráfica de la consola de desarrollador .....	85
Figura 55. App Metrollín Beta en la Play Store .....	86
Figura 56. Metrollín en la Play Store.....	87

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## TABLA DE TABLAS

---

Tabla 1. Comparación entre diferentes sistemas operativos móviles .....	20
Tabla 2. Lista de recurso humano .....	40
Tabla 3. Case de uso Ingresar a la aplicación.....	48
Tabla 4. Caso de uso Presionar Jugar .....	49
Tabla 5. Caso de uso Seleccionar Estación .....	50
Tabla 6. Caso de uso Interactuar con el juego .....	52
Tabla 7. Caso de uso Presionar Instrucciones .....	53
Tabla 8. Caso de uso Presionar Configuración.....	54
Tabla 9. Caso de uso Presionar Puntuación .....	55
Tabla 10. Caso de uso Presionar Acerca De .....	56

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# 1. INTRODUCCIÓN

---

Metrollín es un sistema de información de entretenimiento gráfico en 2D, desarrollado e integrado por herramientas de software libre, para teléfonos inteligentes y tabletas con sistema operativo Android 4.0 (*Ice Cream Sandwich*) o superior. El objetivo de la aplicación de entretenimiento es que los usuarios vayan ingresando y sacando pasajeros de los vagones, en cada una de las 19 plataformas de las estaciones del metro por medio de la pantalla táctil. Cada pasajero tiene sus características propias, entre ellas, color, velocidad y movimiento. Para superar las estaciones, el usuario debe colocar los pasajeros en la plataforma y el vagón correspondiente de acuerdo a su color. Cada estación tiene un límite de tiempo para el ingreso y la salida de los pasajeros. El nivel de dificultad va aumentando entre las diferentes estaciones, con diferentes retos en 7 niveles diferentes. De esta manera se suma puntaje por cada una de los pasajeros ubicados en la plataforma y en los vagones correspondientes.

Metrollín fue desarrollado única y exclusivamente con herramientas de software libre. Gimp es el software de edición de imagen utilizado para la creación de fondos, ambientes, personajes y botones de la aplicación. Linux multimedia Studio fue el software de creación de sonidos y junto con el software Audacity utilizado para la edición y creación de efectos. Arch Linux es el sistema operativo elegido para la instalación de otros software utilizados como Notepadqq, google Chrome, eclipse, entre otros. Y obviamente Android Studio, el IDE ideal para el desarrollo de aplicaciones de Android, que fue el integrador de todos los recursos para darle vida a la aplicación móvil.

## Objetivo General

- Implementar un sistema de información integrado por herramientas de software libre para la creación de un entorno gráfico 2D, en una aplicación interactiva de entretenimiento llamada Metrollín para teléfonos inteligentes y tabletas con sistema operativo Android.

## Objetivos Específicos

- Determinar las herramientas de software libre más adecuadas para el diseño de los recursos para la aplicación.
- Integrar los recursos para el funcionamiento de la aplicación, mediante el IDE Android Studio.
- Aplicar los conceptos de sistemas de información, para la creación de una aplicación interactiva de entretenimiento.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La **primera parte** del documento contiene el marco teórico donde se define que es Android, su historia, su arquitectura, sus librerías, sus versiones, seguridad, sus características y toda la información necesaria para el desarrollo de una aplicación móvil en esta plataforma

La **segunda parte** contiene la metodología, donde se describe el paso a paso sobre cómo se desarrolló la App Metrollín. En esta sección se describe los requerimientos, la consulta sobre las aplicaciones de software utilizados, el diseño de imágenes, iconos, pasajeros, fondos de pantallas, botones, animaciones, música y base de datos. La sección continúa con la programación e integración de los diferentes recursos creados para darle vida a la aplicación. Y finalmente el proceso para subir la aplicación a la *Play Store*, para que pueda ser descargada y utilizada por jugadores de todo el mundo.

La **tercera parte** muestra el resultado de la aplicación final alojada en la *Play Store*, donde se puede visualizar la cantidad de descargas y las opiniones de todos los jugadores quienes hayan descargado el juego.

La **cuarte parte** describe la experiencia de la creación del sistema y las recomendaciones para crear futuros sistemas de entretenimiento. Además menciona las mejoras que se le pueden hacer a la aplicación, para poder mejorarla y hacerla mucho más entretenida y posiblemente hacerla de pago, para comenzar a obtener ganancias.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2. MARCO TEÓRICO

---

### 2.1 Android

En la documentación del gran libro de Android (Girónes, 2012) , se determina que Android es un sistema operativo orientado a dispositivos móviles, basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., una pequeña empresa, que posteriormente fue comprada por Google; en la actualidad lo desarrollan los miembros de la *Open Handset Alliance* (liderada por Google).

Su presentación se realizó el 5 de noviembre de 2007 junto con la fundación *Open Handset Alliance*, en un consorcio de numerosas compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles.

Android consiste en un sistema abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones. Cualquier aplicación puede ser reemplazada libremente, además desarrolladas por terceros, a través de herramientas proporcionadas por Google, y mediante los lenguajes de programación Java y C.

El código fuente de Android está disponible bajo diversas licencias de software libre y código abierto, Google liberó la mayoría del código de Android bajo la licencia Apache. Todo esto permite que un desarrollador no solo pueda modificar su código sino también mejorarlo. A través de esas mejoras puede publicar el nuevo código y con él, ayudar a mejorar el sistema operativo para futuras versiones.

#### 2.1.1 ¿Qué hace que Android sea especial?

Existen muchas plataformas para móviles (*iPhone, Symbian, Windows Phone, BlackBerry, Palm, Java Mobile Edition, Linux Mobile (LiMo)*, etc.); sin embargo Android presenta una serie de características que lo hacen diferente. Es el primero que combina, en una misma solución, las siguientes cualidades:

- **Plataforma realmente abierta.** Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y “customizar” el sistema sin pagar royalties.
- **Portabilidad asegurada.** Las aplicaciones finales son desarrolladas en Java, lo que nos asegura que podrán ser ejecutadas en una gran variedad de dispositivos, tanto presentes como futuros. Esto se consigue gracias al concepto de máquina virtual.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- **Arquitectura basada en componentes inspirados en Internet.** Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una misma aplicación se ejecute en un móvil de pantalla reducida o en un notebook.
- **Filosofía de dispositivo siempre conectado a Internet.**
- **Gran cantidad de servicios incorporados.** Por ejemplo, localización basada tanto en GPS como en redes, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia, etc.
- **Aceptable nivel de seguridad.** Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que hereda de Linux. Además, cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.).
- **Optimizado para baja potencia y poca memoria.** Por ejemplo, Android utiliza la Máquina Virtual Dalvik. Se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.
- **Alta calidad de gráficos y sonido.** Gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en *OpenGL*. Incorpora los códecs estándar más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC, etc.

## 2.1.2 Orígenes

Google adquiere Android Inc. en el año 2005. Se trataba de una pequeña compañía que acababa de ser creada, orientada a la producción de aplicaciones para terminales móviles. Ese mismo año empiezan a trabajar en la creación de una máquina virtual Java optimizada para móviles (Dalvik VM).

En el año 2007 se crea el consorcio Handset Alliance con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros. Una pieza clave de los objetivos de esta alianza es promover el diseño y difusión de la plataforma Android. Sus miembros se han comprometido a publicar una parte importante de su propiedad intelectual como código abierto bajo la licencia Apache v2.0.

En noviembre del 2007 se lanza una primera versión del Android SDK. Al año siguiente aparece el primer móvil con Android (T-Mobile G1). En octubre, Google libera el código fuente de Android, principalmente bajo licencia de código abierto Apache (licencia GPL v2 para el núcleo). Ese mismo mes, se abre Android Market para la descarga de aplicaciones. En abril del 2009, Google lanza la versión 1.5 del SDK que incorpora nuevas características

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

como el teclado en pantalla. A finales del 2009 se lanza la versión 2.0 y durante el 2010 las versiones 2.1, 2.2 y 2.3.

Durante el año 2010, Android se consolida como uno de los sistemas operativos para móviles más utilizados, con resultados cercanos al iPhone, e incluso superando al sistema de Apple en EE.UU.

En el 2011 se lanzan la versión 3.x específica para tabletas y 4.x tanto para móviles como para tabletas. Durante este año, Android se consolida como la plataforma para móviles más importante, alcanzando una cuota de mercado superior al 50%. En 2012, Google cambia su estrategia en su tienda de descargas online, reemplazando Android Market por Google Play Store, donde en un solo portal unifica la descarga de aplicaciones como de contenidos.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 2.1.3 Comparativa con otras plataformas

	 Apple iOS 5.1	 Android 4.0	 Windows Phone 7	 BlackBerry OS 7	 Symbian 9.5
Compañía	Apple	Open Handset Alliance	Windows	RIM	Symbian Foundation
Núcleo del SO	Mac OS X	Linux	Windows CE	Mobile OS	Mobile OS
Familia CPU soportada	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM
Lenguaje de programación	Objective-C, C++	Java, C++	C#, muchos	Java	C++
Licencia de software	Propietaria	Software libre y abierto	Propietaria	Propietaria	Software libre
Año de lanzamiento	2007	2008	2010	2003	1997
Motor del navegador web	WebKit	WebKit	Pocket Internet Explorer	WebKit	WebKit
Soporte Flash	No	Si	No	Si	Si
HTML5	Si	Si	Parcial	Si	No
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry App World	Ovi Store
Número de aplicaciones	400.000	300.000	50.000	30.000	50.000
Coste publicar	\$99 al año	\$25 una vez	\$99 al año	Sin coste	\$1 una vez
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Interfaz personalizable	No	Si	Si	Si	Si
Actualizaciones automáticas del S.O.	Si	Depende del fabricante	Depende del fabricante	Si	Si
Soporte memoria externa	No	Si	No	Si	Si
Fabricante único	Si	No	No	Si	No
Variedad de dispositivos	Modelo único	Muy alta	Baja	Baja	Muy alta
Tipo de pantalla	Capacitativa	Capacitiva /resistiva	Capacitativa	Capacitativa /resistiva	Capacitiva /resistiva
Aplicaciones nativas	Si	Si	No	No	Si

Tabla 1. Comparación entre diferentes sistemas operativos móviles

Nota: Recuperado de *El gran Libro de Android*. (Girónes, 2012)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.1.4 Arquitectura de Android

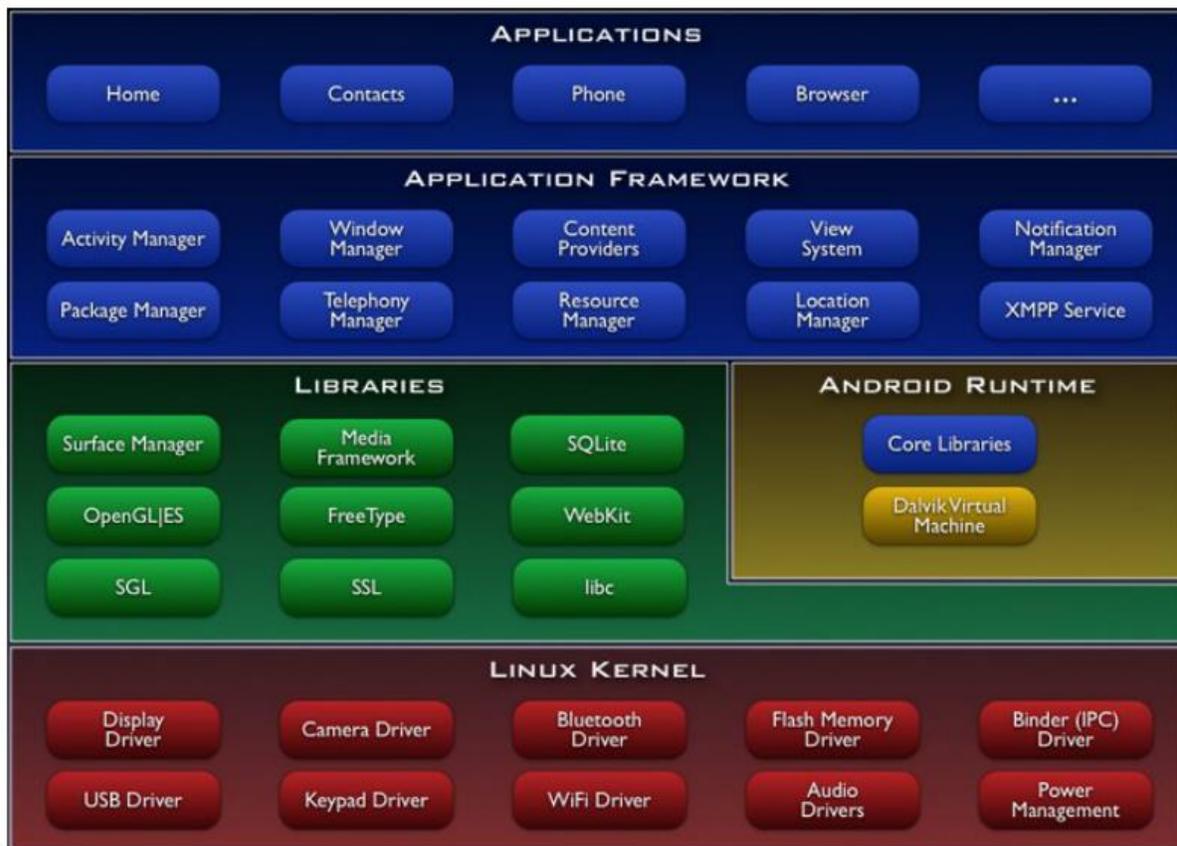


Figura 1. Arquitectura de Android

### El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux, versión 2.6. . Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de para dispositivos. Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

### Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de ejecutarse Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta archivos Dalvik ejecutables (.dex) (formato optimizado para

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ahorrar memoria). Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

También se incluye en el Runtime de Android el “core libraries” con la mayoría de las librerías disponibles en el lenguaje Java.

### **Librerías nativas**

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en el código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- **System C library:** una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework:** librería basada en PacketVideo's OpenCORE; soporta códecs de reproducción y grabación de multitud de formatos de audio, vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- **SGL:** motor de gráficos 2D.
- **Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** proporciona servicios de encriptación Secure Socket Layer.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes. Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.

Los servicios más importantes que incluye son:

**Views:** extenso conjunto de vistas (parte visual de los componentes).

**Resource Manager:** proporciona acceso a recursos que no son en el código.

**Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.

**Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

**Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

## Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de ser ejecutadas en la máquina virtual Dalvik para garantizar la seguridad del sistema. Normalmente, las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.1.5 Las versiones de Android y niveles de API

Cuando se lanza una nueva plataforma siempre ha sido compatible con las versiones anteriores. Es decir, solo se añaden nuevas funcionalidades y, en el caso de modificar alguna funcionalidad, no se elimina, se etiquetan como obsoletas pero se pueden continuar utilizando.

A continuación se describen las plataformas lanzadas hasta la fecha con una breve descripción de las novedades introducidas. Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre. El nivel de API corresponde a números enteros comenzando desde 1. Para los nombres se han elegido postres en orden alfabético Cupcake (v1.5), Donut (v1.6), Éclair (v2.1), Froyo (v2.2), Gingerbread (v2.3), Honeycomb (v3.0), Ice Cream (v2.4), etc.

### **Android 1.0 Nivel de API 1 (septiembre 2008)**

Primera versión de Android. Nunca se utilizó comercialmente, por lo que no tiene mucho sentido desarrollar para esta plataforma.

### **Android 1.1 Nivel de API 2 (febrero 2009)**

No se añadieron apenas funcionalidades, simplemente se fijaron algunos errores de la versión anterior. Es la opción a escoger si queremos desarrollar una aplicación compatible con todos los dispositivos Android. No obstante apenas existen usuarios con esta versión.

### **Android 1.5 Nivel de API 3 (abril 2009, Cupcake)**

Es la primera versión con algún usuario (un 1,3% en julio del 2011). Como novedades, incorpora la posibilidad de teclado en pantalla con predicción de texto, los terminales ya no tienen que tener un teclado físico, así como la capacidad de grabación avanzada de audio y vídeo. También aparecen los widgets de escritorio y live folders. Incorpora soporte para bluetooth estéreo, por lo que permite conectarse automáticamente a auriculares bluetooth. Las transiciones entre ventanas se realizan mediante animaciones.

### **Android 1.6 Nivel de API 4 (diciembre 2009, Donut)**

Permite capacidades de búsqueda avanzada en todo el dispositivo. También incorpora gestures y multi-touch. Permite la síntesis de texto a voz. También facilita que una aplicación pueda trabajar con diferentes densidades de pantalla. Soporte para resolución de pantallas WVGA. Aparece un nuevo atributo XML, onClick, que puede especificarse en una vista. Se mejora Play Store, antes Android Market, permitiendo una búsqueda más sencilla de aplicaciones. Soporte para CDMA/EVDO, 802. 1x y VPNs. Mejoras en la aplicación de la cámara.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### **Android 2.0 Nivel de API 5 (octubre 2009, Éclair)**

Esta versión de API apenas cuenta con usuarios, dado que la mayoría de fabricantes pasaron directamente de la versión 1.6 a la 2.1. Como novedades cabría destacar que incorpora un API para manejar bluetooth 2.1. Nueva funcionalidad que permite sincronizar adaptadores para conectarlo a cualquier dispositivo. Ofrece un servicio centralizado de manejo de cuentas. Mejora la gestión de contactos y ofrece más ajustes en la cámara. Se optimiza la velocidad de hardware. Se aumenta el número de tamaños de ventana y resoluciones soportadas. Nueva interfaz del navegador y soporte para HTML5. Mejoras en el calendario y soporte para Microsoft Exchange. La clase MotionEvent ahora soporta eventos en pantallas multitáctil.

### **Android 2.1 Nivel de API 7 (enero 2010, Éclair)**

Se considera una actualización menor, por lo que le siguieron llamando Éclair. Se destaca el reconocimiento de voz que permite introducir un campo de texto sin necesidad de utilizar el teclado. También permite desarrollar fondos de pantalla animados. Se puede obtener información sobre la señal de la red actual que posea el dispositivo. En el paquete WebKit se incluyen nuevos métodos para manipular bases de datos almacenadas en Web. También se permite obtener permisos de geolocalización y modificarlos en WebView. Incorpora mecanismos para administrar la configuración de la caché de aplicaciones, almacenamiento web y modificar la resolución de la pantalla. También se puede manejar vídeo, historial de navegación, vistas personalizadas, etc.

### **Android 2.2 Nivel de API 8 (mayo 2010, Froyo)**

Como característica más destacada se puede indicar la mejora de velocidad de ejecución de las aplicaciones (ejecución del código de la CPU de 2 a 5 veces más rápido que en la versión 2.1 de acuerdo a varios benchmarks). Esto se consigue con la introducción de un nuevo compilador JIT de la máquina Dalvik. Se añaden varias mejoras relacionadas con el navegador Web, como el soporte de Adobe Flash 10.1 y la incorporación del motor JavaScript V8 utilizado en Chrome, o la incorporación del campo de “subir archivo” en un formulario.

El desarrollo de aplicaciones permite las siguientes novedades: se puede preguntar al usuario si desea instalar una aplicación en un medio de almacenamiento externo (como una tarjeta SD), como alternativa a la instalación en la memoria interna del dispositivo. Las aplicaciones se actualizan de forma automática cuando aparece una nueva versión. Proporciona un servicio para la copia de seguridad de datos que se puede realizar desde la propia aplicación para garantizar al usuario el mantenimiento de sus datos. Por último, facilita que las aplicaciones interaccionen con el reconocimiento de voz y que terceras

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

partes proporcionen nuevos motores de reconocimiento. Se mejora la conectividad: ahora se puede utilizar el teléfono para dar acceso a Internet a otros dispositivos (tethering), tanto por USB como por Wi-Fi. También se añade el soporte a Wi-Fi IEEE 802.11n.

Se añaden varias mejoras en diferentes componentes: en el API gráfica OpenGL ES se pasa a soportar la versión 2.0. También puede realizar fotos o vídeos en cualquier orientación (incluso vertical) y configurar otros ajustes de la cámara. Para finalizar, permite definir modos de interfaz de usuario (“automóvil” y “noche”) para que las aplicaciones se configuren según el modo seleccionado por el usuario.

### **Android 2.3 Nivel de API 9 (diciembre 2010, Gingerbread)**

Debido al éxito de Android en las nuevas tabletas, ahora soporta mayores tamaños de pantalla y resoluciones (WXGA y superiores).

Incorpora un nuevo interfaz de usuario con un diseño actualizado. Dentro de las mejoras de la interfaz de usuario destaca la mejora de la funcionalidad de “cortar, copiar y pegar” y un teclado en pantalla con capacidad multitáctil. Incluye soporte nativo para varias cámaras, pensado en la segunda cámara usada en videoconferencia. La incorporación de esta segunda cámara ha propiciado la inclusión de reconocimiento facial para identificar al usuario del terminal.

La máquina virtual Dalvik para Android introduce un nuevo recolector de basura que minimiza las pausas de la aplicación, ayudando a garantizar una mejor animación y el aumento de la capacidad de respuesta en juegos y aplicaciones similares. Se trata de corregir así una de las lacras de este sistema operativo móvil, que en versiones previas no ha sido capaz de cerrar bien las aplicaciones en desuso. Se dispone de mayor apoyo para el desarrollo de código nativo (NDK). También se mejora la gestión de energía y control de aplicaciones. Y se cambia el sistema de archivos, que pasa de YAFFS a EXT4.

Entre otras novedades se destaca el soporte nativo para telefonía sobre Internet VoIP/SIP. El soporte para reproducción de vídeo WebMA/P8 y codificación de audio AAC. El soporte para la tecnología NFC. Las facilidades en el audio, gráficos y entradas para los desarrolladores de juegos. El soporte nativo para más sensores (como giroscopios y barómetros) y un gestor de descargas para las descargas largas.

### **Android 3.0 Nivel de API 11 (febrero 2011, Honeycomb)**

Para mejorar la experiencia de Android en las nuevas tabletas se lanza la versión 3.0 optimizada para dispositivos con pantallas grandes. La nueva interfaz de usuario ha sido completamente rediseñada con paradigmas nuevos para la interacción, navegación y

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

personalización. La nueva interfaz se pone a disposición de todas las aplicaciones, incluso las construidas para versiones anteriores de la plataforma.

Las principales novedades de este SDK son:

Con el objetivo de adaptar la interfaz de usuario a pantallas más grandes se incorporan las siguientes características: resolución por defecto WXGA (1280x800), escritorio 3D con widgets rediseñados, nuevos componentes y vistas, notificaciones mejoradas, arrastrar y soltar, nuevo cortar y pegar, barra de acciones para que las aplicaciones dispongan de un menú contextual siempre presente y otras características para aprovechar las pantallas más grandes. Se mejora la reproducción de animaciones 2D/3D gracias al renderizador OpenGL acelerado por hardware. El nuevo motor de gráficos Rederscript saca un gran rendimiento de los gráficos en Android e incorpora su propia API. Primera versión de la plataforma que soporta procesadores multinúcleo. La máquina virtual Dalvik ha sido optimizada para permitir multiprocesado, lo que permite una ejecución más rápida de las aplicaciones, incluso aquellas que son de hilo único.

Se incorporan varias mejoras multimedia, como listas de reproducción M3U a través de HTTP Live Streaming, soporte a la protección de derechos musicales (DRM) y soporte para la transferencia de archivos multimedia a través de USB con los protocolos MTP y PTP.

En esta versión se añaden nuevas alternativas de conectividad, como las nuevas APIS de Bluetooth A2DP y HSP con streaming de audio. También se permite conectar teclados completos por USB o Bluetooth.

El uso de los dispositivos en un entorno empresarial es mejorado. Entre las novedades introducidas destacamos las nuevas políticas administrativas con encriptación del almacenamiento, caducidad de contraseña y mejoras para administrar los dispositivos de empresa de forma eficaz.

A pesar de la nueva interfaz gráfica optimizada para tabletas, Android 3.0 es compatible con las aplicaciones creadas para versiones anteriores. La tecla de menú, inexistente en las nuevas tabletas, es reemplazada por un menú que aparece en la barra de acción.

### **Android 3.1 Nivel de API 12 (mayo 2011)**

Permite manejar dispositivos conectados por USB (tanto host como dispositivo). Protocolo de transferencia de fotos y vídeo (PTP/MTP) y de tiempo real (RTP).

### **Android 3.2 Nivel de API 13 (julio 2011)**

Optimizaciones para distintos tipos de tableta. Zoom compatible para aplicaciones de tamaño fijo. Sincronización multimedia desde SD.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### **Android 4.0 Nivel de API 14 (octubre 2011, Ice Cream Sandwich)**

La característica más importante es que se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo.

Entre las características más interesantes destacamos:

Se introduce un nuevo interfaz de usuario totalmente renovado. Por ejemplo, se reemplazan los botones físicos por botones en pantalla (como ocurría en las versiones 3.x).

Nuevo API de reconocedor facial que permite, entre otras muchas aplicaciones, desbloquear el teléfono. También se mejora en el reconocimiento de voz. Por ejemplo, se puede empezar a hablar en cuanto pulsamos el botón.

Aparece un nuevo gestor de tráfico de datos por Internet, donde podremos ver el consumo de forma gráfica y donde podemos definir los límites a ese consumo para evitar cargos inesperados con la operadora. Incorpora herramientas para la edición de imágenes en tiempo real, con herramientas para distorsionar, manipular e interactuar con la imagen al momento de ser capturada. Se mejora el API para comunicaciones por NFC y la integración con redes sociales.

En diciembre del 2011 aparece una actualización de mantenimiento (versión 4.0.2) que no aumenta el nivel de API.

### **Android 4.0.3 Nivel de API 15 (diciembre 2011)**

Se introducen ligeras mejoras en algunas APIs incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros. En marzo de 2012 aparece la actualización 4.0.4.

### **Android 4.1 Nivel de API 16 (julio 2012, Jelly Bean)**

En esta versión se hace hincapié en mejorar un punto débil de Android: la fluidez del interfaz de usuario. Con este propósito se incorporan varias técnicas, como: sincronismo vertical, triple búfer y aumentar la velocidad del procesador al tocar la pantalla.

Se mejoran las notificaciones con un sistema de información expandible personalizada. Los Widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio. El dictado por voz puede realizarse sin conexión a Internet (de momento en inglés).

Se introducen varias mejoras en Google Search. Se potencia la búsqueda por voz con resultados en forma de ficha. La función Google Now permite utilizar información de posición, agenda y hora en las búsquedas.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se incorporan nuevo soporte para usuarios internacionales: como texto bidireccional y teclados instalables. Para mejorar la seguridad las aplicaciones son cifradas. También se permite actualizaciones parciales de aplicaciones.

#### **Android 4.2 Nivel de API 17 (noviembre 2012)**

Una de las novedades más importantes es que se puede crear varias cuentas de usuario en el mismo dispositivo. Aunque, esta característica solo está disponible en tabletas. Cada cuenta tendrá sus propias aplicaciones y configuración.

Los Widgets de escritorio pueden aparecer en la pantalla de bloqueo. Se incorpora un nuevo teclado predictivo deslizante al estilo Swype. Posibilidad de conectar dispositivo y TVHD mediante wifi (Miracast). Mejoras menores en las notificaciones. Nueva aplicación de cámara que incorpora la funcionalidad Photo Sphere para hacer fotos panorámicas inmersivas (en 360°).

#### **Android 4.3 Nivel de API 18 (julio 2013)**

Esta versión introduce mejoras en múltiples áreas. Entre ellas los perfiles restringidos (disponible sólo en tabletas) que permiten controlar los derechos de los usuarios para ejecutar aplicaciones específicas y para tener acceso a datos específicos. Igualmente, los programadores pueden definir restricciones en las Apps, que los propietarios puedan activar si quieren. Se da soporte para Bluetooth Low Energy (BLE) que permite a los dispositivos Android comunicarse con los periféricos con bajo consumo de energía. Se agregan nuevas características para la codificación, transmisión y multiplexación de datos multimedia. Se da soporte para OpenGL ES 3.0. Se mejora la seguridad para gestionar y ocultar las claves privadas y credenciales.

#### **Android 5.0 Nivel de API 21 (noviembre 2014, Lollipop)**

La novedad más importante de Lollipop es la extensión de Android a nuevas plataformas, incluyendo Google Wear, Google TV y Google Card. Hay un cambio significativo en la arquitectura, al utilizar la máquina virtual ART en lugar de Dalvik. Esta novedad ya había sido incorporada en la versión anterior a modo de prueba. ART mejora de forma considerable el tiempo de ejecución del código escrito en Java. Además se soporta dispositivos de 64 bits en procesadores ARM, x86, y MIPS. Muchas aplicaciones del sistema (Chrome, Gmail,...) se han incorporado en código nativo para una ejecución más rápida.

Desde el punto de vista del consumo de batería, hay que resaltar que en Lollipop el modo de ahorro de batería se activa por defecto. Este modo desconecta algunos componentes en caso de que la batería esté baja. Se incorpora una nueva API

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

(android.App.job.JobScheduler) que nos permite que ciertos trabajos se realicen sólo cuando se cumplan determinadas condiciones (por ejemplo con el dispositivo cargando). También se incluyen completas estadísticas para analizar el consumo que nuestras aplicaciones hacen de la batería.

En el campo Gráfico Android Lollipop incorpora soporte nativo para OpenGL ES 3.1. Además esta versión permite añadir a nuestras aplicaciones un paquete de extensión con funcionalidades gráficas avanzadas (fragment shader, tessellation, geometry shaders, ASTC,...).

Otro aspecto innovador de la nueva versión lo encontramos en el diseño de la interfaz de usuario. Se han cambiado los iconos, incluyendo los de la parte inferior (Retroceder, Inicio y Aplicaciones), que ahora son un triángulo, un círculo y un cuadrado.

### **Android 6.0 (mayo 2015, Marshmallow)**

Introduce un modelo de permisos rediseñado: ahora hay únicamente ocho categorías de permisos, y las aplicaciones ya no conceden automáticamente todos sus permisos específicos en el momento de la instalación. Se utiliza ahora un sistema "opt-in", en el que los usuarios pueden conceder o denegar permisos individuales a una aplicación cuando lo requieran, tales como el acceso a la cámara o el micrófono. Las aplicaciones recuerdan las concesiones de permisos de cada usuario, y pueden ser ajustados en cualquier momento. Sólo las aplicaciones compiladas para Android "M" con su SDK utilizarán el nuevo sistema de permisos, mientras que todas las demás aplicaciones continuarán utilizando el modelo de permisos anterior.

Android "M" ofrece soporte nativo para el reconocimiento de huellas digitales, lo que permite el uso de las mismas para desbloquear los dispositivos, y para la autenticación de Play Store o Android Play; una API estándar también está disponible para la implementación de la autenticación basada en huellas digitales en otras aplicaciones. Un nuevo sistema de administración de energía conocida como "doze" reduce las actividades en fondo del dispositivo cuando el sistema operativo detecta que el dispositivo no está siendo tocado físicamente. Android "M" es compatible con USB Tipo-C, lo que implica la capacidad de carga a través de USB, hasta 5 veces más rápida, dependiendo del terminal. Android "M" también introduce "enlaces verificados" que se pueden configurar para abrir directamente una aplicación especificada, sin la necesidad de mostrar mensajes de usuario.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.1.6 Componentes de una aplicación

Existen una serie de elementos clave que resultan imprescindibles para desarrollar aplicaciones en Android.

**Vista (View):** Las vistas son los elementos que componen la interfaz de usuario de una aplicación. Son por ejemplo, un botón, una entrada de texto,... Todas las vistas van a ser objetos descendientes de la clase View, y por tanto, pueden ser definidos utilizando código Java. Sin embargo, lo habitual va a ser definir las vistas utilizando un archivo XML y dejar que el sistema cree los objetos por nosotros a partir de este archivo. Esta forma de trabajar es muy similar a la definición de una página web utilizando código HTML.

**Layout:** Un Layout es un conjunto de vistas agrupadas de una determinada forma. Vamos a disponer de diferentes tipos de Layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los Layouts también son objetos descendientes de la clase View. Igual que las vistas, los Layouts pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML.

**Actividad (Activity):** Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En Android cada uno de estos elementos, o pantallas, se conoce como actividad. Su función principal es la creación del interfaz de usuario. Una aplicación suele necesitar varias actividades para crear el interfaz de usuario. Las diferentes actividades creadas serán independientes entre sí, aunque todas trabajarán para un objetivo común. Toda actividad ha de pertenecer a una clase descendiente de Activity.

**Servicio (Service):** Un servicio es un proceso que se ejecuta “detrás”, sin la necesidad de una interacción con el usuario. Es algo parecido a un demonio en Unix o a un servicio en Windows. En Android disponemos de dos tipos de servicios: servicios locales, que pueden ser utilizados por aplicaciones del mismo terminal y servicios remotos, que pueden ser utilizados desde otros terminales.

**Intención (Intent):** Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos:

- Lanzar una actividad.
- Lanzar un servicio.
- Lanzar un anuncio de tipo broadcast.
- Comunicarnos con un servicio

Los componentes lanzados pueden ser internos o externos a nuestra aplicación. También se utilizan las intenciones para el intercambio de información entre estos componentes.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En muchas ocasiones una intención no será inicializada por la aplicación, si no por el sistema, por ejemplo, cuando se pide visualizar una página web. En otras ocasiones será necesario que la aplicación inicialice su propia intención. Para ello se creará un objeto de la clase Intent.

**Receptor de anuncios (Broadcast Receiver):** Un receptor de anuncios recibe y reacciona ante anuncios de tipo broadcast. Existen muchos originados por el sistema, como por ejemplo Batería baja, Llamada entrante,... Aunque, las aplicaciones también puede lanzar un anuncio broadcast. No tienen interfaz de usuario, aunque pueden iniciar una actividad para atender a un anuncio.

**Proveedores de contenido (Content Provider):** La compartición de información entre teléfonos móviles resulta un tema vital. Android define un mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de archivos. Con este mecanismo se puede acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones.

### 2.1.7 Ciclo de vida de una actividad

El ciclo de vida de una aplicación Android es bastante diferente del ciclo de vida de una aplicación en otros S.O., como Windows. La mayor diferencia es que, en Android el ciclo de vida es controlado principalmente por el sistema, en lugar de ser controlado directamente por el usuario. Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, conocidos como actividades. Además de varias actividades una aplicación también puede contener servicios. Son las actividades las que realmente controlan el ciclo de vida de las aplicaciones, dado que el usuario no cambia de aplicación, sino de actividad. El sistema va a mantener una pila con las actividades previamente visualizadas, de forma que el usuario va a poder regresar a la actividad anterior pulsando la tecla "atrás".

Una aplicación Android corre dentro de su propio proceso Linux. Este proceso es creado para la aplicación y continuará vivo hasta que ya no sea requerido y el sistema reclame su memoria para asignársela a otra aplicación.

Una característica importante, y poco usual, de Android es que la destrucción de un proceso no es controlado directamente por la aplicación. En lugar de esto, es el sistema quien determina cuando destruir el proceso, basándose en el conocimiento que tiene el sistema de las partes de la aplicación que están corriendo (actividades y servicios), acerca de qué tan importantes son para el usuario y cuánta memoria disponible hay en un determinado momento. Si tras eliminar el proceso de una aplicación, el usuario vuelve a ella, se creará de nuevo el proceso, pero se habrá perdido el estado que tenía esta aplicación. En estos casos, va a ser responsabilidad del programador almacenar el estado de las actividades, si

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

queremos que cuando sea reiniciada conserve su estado. Como vemos, Android es sensible al ciclo de vida de una actividad, por lo tanto necesitas comprender y manejar los eventos relacionados con el ciclo de vida si quieres crear aplicaciones estables.

Una actividad en Android puede estar en uno de estos cuatro estados:

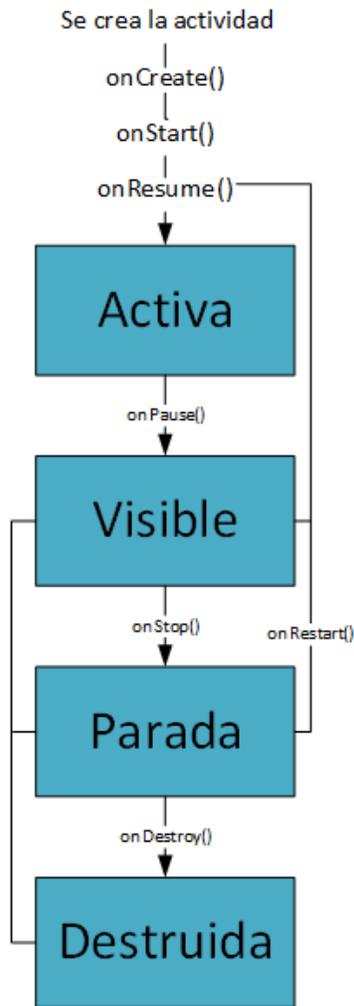
**Activa (Running).** La actividad está encima de la pila, lo que quiere decir que es visible y tiene el foco.

**Visible (Paused).** La actividad es visible pero no tiene el foco. Se alcanza este estado cuando pasa a activa otra actividad con alguna parte transparente o que no ocupa toda la pantalla. Cuando una actividad está tapada por completo, pasa a estar parada.

**Parada (Stopped).** Cuando la actividad no es visible, se recomienda guardar el estado de la interfaz de usuario, preferencias, etc.

**Destruída (Destroyed).** Cuando la actividad termina al invocarse el método finish (), o es matada por el sistema Android, sale de la pila de actividades.

Cada vez que una actividad cambia de estado se van a producir eventos que podrán ser capturados por ciertos métodos de la actividad. A continuación, se muestra un esquema que ilustra los métodos que capturan estos eventos.



**Figura 2. Ciclo de vida de una actividad Android**

**onCreate (Bundle):** Se llama en la creación de la actividad. Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Puede recibir información de estado de instancia (en una instancia de la clase Bundle), por si se reanuda desde una actividad que ha sido destruida y vuelta a crear.

**onStart():** Nos indica que la actividad está a punto de ser mostrada al usuario.

**onResume():** Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.

**onPause():** Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra aplicación es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**onStop():** La actividad ya no va a ser visible para el usuario. Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

**onRestart():** Indica que la actividad va a volver a ser representada después de haber pasado por onStop().

**onDestroy():** Se llama antes de que la actividad sea totalmente destruida.

Por ejemplo, cuando el usuario pulsa el botón <volver> o cuando se llama al método finish(). Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

## 2.1.8 Base de datos en Android

Las bases de datos son una herramienta de gran potencia en la creación de aplicaciones informáticas. Hasta hace muy poco resultaba muy costoso y complejo incorporar bases de datos a las aplicaciones. No obstante, Android incorpora la librería SQLite que permitirá utilizar bases de datos mediante el lenguaje SQL, de una forma sencilla y utilizando muy pocos recursos del sistema. Almacenar información en una base de datos no es mucho más complejo que almacenarlos en un archivo y, además, resulta mucho más potente.

SQL es el lenguaje de programación más utilizado para bases de datos.

Para manipular una base de datos en Android se utiliza la clase SQLiteOpenHelper que facilita tanto la creación de la base de datos, como el trabajar con futuras versiones de esta base de datos. Para crear un descendiente de esta clase hay que implementar los métodos *onCreate()*, y *onUpgrade()* y opcionalmente *onOpen()* . La gran ventaja de utilizar esta clase es que ella se preocupará de abrir la base de datos si existe o de crearla si no existe. Incluso de actualizar la versión si se decide crear una nueva estructura de la base de datos.

Además, esta clase tiene dos métodos *getReadableDatabase()* y *getWritableDatabase()* que abren la base de datos en modo solo lectura o lectura y escritura. En caso de todavía no existir la base de datos, estos métodos se encargarán de crearla.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.1.9 Los tres pilares de la seguridad en Android

En otras plataformas, como en iOS del iPhone, toda aplicación ha de ser validada por Apple antes de poder ser instalada en un teléfono. Esto limita a los pequeños programadores y da un poder excesivo a Apple. Se trata de un planteamiento totalmente contrario al software libre.

Android propone un esquema de seguridad que protege a los usuarios, sin la necesidad de imponer un sistema centralizado y controlado por una única empresa.

La seguridad en Android se fundamenta en los siguientes tres pilares:

Android está basado en Linux, por lo tanto, se va poder aprovechar la seguridad que incorpora este SO. De esta forma Android puede impedir que las aplicaciones tengan acceso directo al hardware o interfieran con recursos de otras aplicaciones.

Toda aplicación ha de ser firmada con un certificado digital que identifique a su autor. La firma digital también garantiza que el archivo de la aplicación no ha sido modificado. Si se desea modificar la aplicación está tendrá que ser firmada de nuevo, y esto solo podrá hacerlo el propietario de la clave privada. No es preciso (ni frecuente) que el certificado digital sea firmado por una autoridad de certificación.

Si se quiere que una aplicación tenga acceso a partes del sistema que pueden comprometer la seguridad del sistema se debe utilizar un modelo de permisos, de forma que el usuario conozca los riesgos antes de instalar la aplicación.

## 2.1.10 Usuario Linux y acceso a archivos

Para proteger el acceso a recursos utilizados por otras aplicaciones, Android crea una cuenta de usuario Linux (user ID) nueva por cada paquete ( apk) instalado en el sistema. Este usuario es creado cuando se instala la aplicación y permanece constante durante toda su vida en el dispositivo.

Cualquier dato almacenado por la aplicación será asignado a su usuario Linux, por lo que normalmente no tendrán acceso otras aplicaciones. No obstante, cuando se crea un archivo puedes usar los modos `mode_world_readable` y/o `mode_world_writeable` para permitir que otras aplicaciones puedan leer o escribir en el archivo. Aunque otras aplicaciones puedan escribir el archivo, el propietario siempre será el usuario asignado a la aplicación que lo creó.

Dado que las restricciones de seguridad se garantizan a nivel de proceso, el código de dos paquetes no puede, normalmente, ejecutarse en el mismo proceso. Para ello sería necesario usar el mismo usuario. Se puede utilizar el atributo `shareduserid` en *AndroidManifest.xml* para asignar un mismo usuario Linux a dos aplicaciones. Con esto conseguimos que a

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

efectos de seguridad ambas aplicaciones sean tratadas como una sola. Por razones de seguridad, ambas aplicaciones han de estar firmadas con el mismo certificado digital.

### 2.1.11 El esquema de permisos en Android

Para proteger ciertos recursos y características especiales del hardware, Android define un esquema de permisos. Toda aplicación que acceda a estos recursos está obligada a declarar su intención de usarlos. En caso de que una aplicación intente acceder a un recurso del que no ha solicitado permiso, se generará una excepción de permiso y la aplicación será interrumpida inmediatamente. Cuando el usuario instala una aplicación, este podrá examinar la lista de permisos que solicita la aplicación y decidir si considera oportuno instalar dicha aplicación.

Para solicitar un determinado permiso en tu aplicación, solo es incluir una etiqueta `<uses-permission>` en el archivo *AndroidManifest.xml* de tu aplicación.

```
<uses-permission android:name="string"
  android:maxSdkVersion="integer" />
```

Figura 3. Ejemplo de permisos en el Android Manifest

### 2.1.12 Licencias aplicaciones Android

**GNU GPL 2:** La GNU GPL2 establece que la persona está en libertad de usar, modificar y distribuir el software sin estar obligado a liberar el código fuente. Es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la *Free Software Foundation* (FSF) para el proyecto GNU.

**Apache 2.0:** La licencia Apache 2.0 indica que los programas derivados no necesariamente deben ser de este mismo tipo de licencia, ni siquiera exige que el software se tenga que distribuir libremente y con código abierto, pero exige que se realice un informe a los receptores que en la distribución, se ha usado código con licencia Apache.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3. METODOLOGÍA

---

A continuación se describe el proceso y recursos utilizados para el desarrollo del sistema de entretenimiento.

### 3.1 Análisis de la aplicación

#### 3.1.1 Especificaciones de la aplicación

##### 3.1.1.1 Descripción del sistema de información

Metrollín es una aplicación interactiva de entretenimiento para teléfonos inteligentes y tabletas con sistema operativo Android, que consiste en un entorno gráfico 2D, integrado por herramientas de software libre.

El objetivo de la aplicación de entretenimiento es que los usuarios vayan ingresando y sacando pasajeros de los vagones, en cada plataforma de las 19 estaciones del metro, por medio de la pantalla táctil. Cada pasajero tiene características propias, entre ellas, color, velocidad y movimiento. Para superar las estaciones, el usuario debe colocar los pasajeros en la plataforma y el vagón correspondiente de acuerdo a su color. Cada estación tendrá un límite de tiempo para el ingreso y la salida de los pasajeros. Las 19 estaciones están agrupadas en 7 niveles diferentes, en donde cada nivel cambia la velocidad, cantidad y color de los pasajeros, para hacer la aplicación más entretenida y más difícil para los jugadores.

##### 3.1.1.2 Características del sistema

- El sistema funciona exclusivamente en teléfonos inteligentes y tabletas con sistema operativo Android versión 4.0 (*Ice Cream Sandwich*) o superior.
- El sistema está desarrollado para soportar diferentes resoluciones de pantalla de teléfonos inteligentes y tabletas.
- El objetivo del juego es sacar la mayor cantidad de pasajeros de los vagones a la plataforma, así como también ingresarlos de la plataforma a los diferentes vagones del metro, en cada una de sus 19 estaciones.
- Los pasajeros se mueven a una velocidad y dirección única, generada por números aleatorios.
- Los colores de los pasajeros son 3: azul, verde y rojo.
- Los colores de los vagones son los mismos que los pasajeros: azul, verde y rojo.
- Los pasajeros de color verde, deben ser colocados en la plataforma y vagón de color verde, así como los pasajeros de azul y rojo, en sus respectivos colores.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Cuando un pasajero es colocado en el vagón o plataforma correcta, se suma 1 punto.
- Si el pasajero es colocado en el vagón o plataforma correcta, se mostrará un gráfico de +1, acompañado de un sonido de acierto, de lo contrario sonará un sonido de error, acompañado de una animación de ERROR.
- El usuario utiliza la pantalla táctil del dispositivo para seleccionar los pasajeros y llevarlos a la plataforma y a los diferentes vagones.
- La pantalla principal cuenta con el título del juego, y diferentes botones para direccionar a las diferentes secciones del juego.
- Los botones de la pantalla principal son: Jugar, instrucciones, configuración, puntuaciones, acerca de y salir.
- El botón jugar, dirige al usuario a la pantalla de selección de estaciones del metro.
- En la pantalla de estaciones del metro, se encuentra un mapa con las 19 estaciones, con sus nombres y un icono para cada estación.
- Los nombres de las estaciones son:
  - Puerta Norte, Ciudadela, La Iglesia, La Cancha, Prado Verde, Barrio Uno, 5 de Marzo, Los Lagos, Parque Alegre, Barrio Dos, El centro, Rio Claro, Ed. Uno, Ed. Central, Ed. Dos, El Bosque, Barrio Tres, Zona Poblada y Puerta Sur.
- Cuando se pulsa el icono de una estación, dirige al usuario a la pantalla Estación, que es donde el usuario interactúa con los pasajeros y los vagones del metro.
- En cada una de las estaciones, se encuentra un número aleatorio de pasajeros que se desplazan por la mitad de la pantalla superior.
- En la parte inferior de la pantalla, se encuentran los 3 vagones del metro.
- En la parte superior de la pantalla, se encuentra el nombre de la estación, la cantidad de pasajeros que el usuario ha sacado e ingresado correctamente, el tiempo restante para el cierre de puertas y un mensaje de ayuda al usuario, que le indica las acciones que debe realizar.
- El botón configuración direcciona a la pantalla configuración, donde el usuario va a poder editar su nombre, desactivar/activar el sonido, la puntuación, la estación actual, la cantidad de viajes y una opción para restaurar los datos.
- El botón puntuación direcciona a la pantalla puntuación, donde se listara los usuarios que han logrado la mayor puntuación.
- El botón acerca de direcciona a la pantalla acerca de, donde se mostrará la información del videojuego como sus desarrolladores y la licencia.
- El botón salir, cerrara la aplicación y retornara al menú del dispositivo.
- La pantalla principal, selección de estación y estación tendrán música de fondo.
- Al seleccionar un botón, activara un sonido de fondo.
- El jugador debe recorrer todas las estaciones de izquierda a derecha.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- El jugador solo puede acceder a la estación actual, no puede adelantarse ni retrocederse en la selección de estación.
- Para pasar a la siguiente estación, el jugador debe ingresar más de la mitad de los pasajeros correctamente en cada estación, de lo contrario no podrá avanzar a la siguiente estación.

## 3.1.2 Recursos utilizados para el desarrollo

### 3.1.2.1 Recurso Humano

Nombre	Cargo
Juan Fernando Cañola García	Analista Desarrollador
Edwin Daniel Álvarez Vásquez	Analista Desarrollador
Delio Augusto Aristizabal Martínez	Asesor

Tabla 2. Lista de recurso humano

### 3.1.2.2 Recursos de Hardware

Portátil ASUS

Procesador Intel core i7

8GB Memoria RAM

Disco Duro de 500 GB

### 3.1.2.3 Recursos de Software

Sistema Operativo Arch Linux

Software GIMP

Software Linux Multimedia Studio

Software Audacity

Software Notepadqq

Software Libre Office

Software Android Studio

Software Google Chrome

Base de datos MySQL

Servidor Apache PHP

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.1.3 Consultas

#### 3.1.3.1 Herramientas de software libre utilizadas

**GIMP:** GIMP es un acrónimo de GNU Image Manipulation Program (GNU programa manipulador de imágenes). Es distribuido gratuitamente para tareas como creación y edición de imágenes. Es distribuido bajo la licencia GPL. Editor de imágenes para crear todos los gráficos de personajes, fondos, botones, texto para el sistema de entretenimiento.

Página web: <https://www.gimp.org>

**Linux multimedia Studio:** Estación de trabajo de audio digital para la creación de pistas de sonido. Se utilizará para crear todos los efectos de sonido y música de fondo del sistema de entretenimiento.

Página web: <https://lms.io/>

**Audacity:** Editor de audio libre. Este programa será utilizado para modificar algunos sonidos creados con Linux multimedia Studio.

Página web: <http://audacity.es/>

**Notepadqq:** Es un editor de texto parecido a notepad++. Editor de texto para realizar anotaciones y modificaciones al código.

Página web: <http://notepadqq.altervista.org/wp/>

**Libre office:** LibreOffice es un poderoso paquete de oficina. Su interfaz limpia y sus potentes herramientas permiten dar rienda suelta a tu creatividad y hacer crecer tu productividad. LibreOffice incluye varias aplicaciones que lo convierten en el paquete de oficina libre y abierta más eficaz del mercado.

Página web: <https://es.libreoffice.org/>

**Android Studio:** Es un entorno de desarrollo integrado para la plataforma Android. Contiene editor de código, depuración, herramientas de rendimiento, sistema de generación y despliegue inmediato para la creación de aplicaciones. El integrador de todos los recursos.

Página web: <https://developer.android.com/studio/index.html>

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**Arch Linux:** Arch Linux es una distribución GNU/Linux independiente, de propósito general, desarrollada para i686/x86-64, que se esfuerza por mantenerse al día y, por lo general, ofrece las últimas versiones estables de la mayoría del software, basada en un modelo rolling-release. La instalación, por defecto, deja un sistema de base mínima, que el usuario configurará posteriormente mediante la adición de solo lo que requiere o desee para sus propósitos particulares. Sistema operativo en el que se instalaron todas estas herramientas.

Página web: <https://www.archlinux.org/>

**Google Chrome:** Navegador web rápido y gratuito para las diferentes consultas.

Página web: <https://www.google.com/chrome/browser/desktop/index.html>

**MySQL:** Es un sistema de gestión de base de datos relacional liberado bajo una licencia GPL, utilizado para el almacenamiento permanente de los datos.

Página Web: <https://www.mysql.com/>

### 3.1.3.2 Instalación de herramientas de software libre utilizadas

#### Instalación Arch Linux

Para instalar los paquetes en Arch Linux se utilizan los comandos:

```
Linux multimedia Studio
sudo pacman -S Imms
Gimp
Sudo pacman -S gimp
Google Chrome
Sudo yaourt -S google-chrome
Android Studio
Instalar los 3 paquetes
sudo pacman -S android-sdkAUR
sudo pacman -S android-sdk-platform-toolsAUR
sudo pacman -S android-sdk-build-toolsAUR
y luego
sudo pacman -S Android-studio
NotePadQQ
sudo pacman -S notepadqq-src
Audacity
Sudo pacman -S audacity
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Instalación Windows

Los programas pueden ser instalados en Windows descargándolos de su página oficial y ejecutando su instalador y siguiendo los pasos.

### 3.1.3.3 Subir aplicaciones a la Play Store

Los pasos para subir aplicaciones a la Play Store son los siguientes:

#### 3.1.3.3.1 Obtener una cuenta de programador

Obtener una cuenta de programador, con una tarjeta de crédito más 25 dólares y dirigirse al siguiente enlace <https://play.google.com/apps/publish/signup/>, pagar la tarifa de registro y completar los datos de la cuenta.

Una vez que se haya registrado para obtener una cuenta de programador, se podrá subir Apps a Google Play por medio de Google Play Developer Console.

#### 3.1.3.3.2 Subir APK a la Play Store

##### Agrega un archivo APK

Ir a: Google Play Developer Console.

Seleccionar **Todas las aplicaciones**  > **Agregar nueva aplicación.**

En el menú desplegable, seleccionar el idioma predeterminado y agregar un título para la App. Luego, escribir el nombre de la App como desee que aparezca en Google Play.

Seleccionar: **Cargar archivo APK.**

Elegir los canales de producción (Alfa o Beta) y seleccionar la opción para **cargar el archivo APK**. Para obtener más información sobre las pruebas Alfa o Beta, consultar cómo usar las pruebas Alfa o Beta, y los lanzamientos en etapas.

#### 3.1.3.3.3 Requerimientos para subir una aplicación a la Play Store

##### Tamaños de los archivos

Los archivos APK tienen un tamaño máximo, que varía en función de la versión de Android compatible con tu APK. Si un único archivo APK no es compatible con todos los dispositivos, puedes subir varios archivos APK con la misma ficha de la App, que admitan dispositivos con diferente configuración.

**100 MB:** archivos APK diseñados para Android 2.3 o versiones posteriores (API nivel 9-10, 14 o posterior).

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**50MB:** archivos APK diseñados para Android 2.2 y versiones anteriores (API nivel 8 o anterior).

**Sugerencia:** Los usuarios deben ejecutar la versión 5.2 de Play Store o una posterior para instalar archivos APK de 100 MB.

### **Firmar la App**

Android requiere que todas las Apps estén firmadas digitalmente con un certificado antes que puedan instalarse.

Si se ve un mensaje que indica que la APK está firmada con un certificado inseguro y que el tamaño de la clave debe ser de al menos 1,024 bits, seguir las instrucciones para firmar la App manualmente.

**Importante:** Guardar el almacén de claves en una ubicación segura. Si se pierde, se debe publicar la App con una clave y un nombre de paquete nuevos. Si se tiene que hacerlo, también se deberá anular la publicación de la App original y actualizar su descripción.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3.2 Diseño

### 3.2.1 Diagrama de contexto

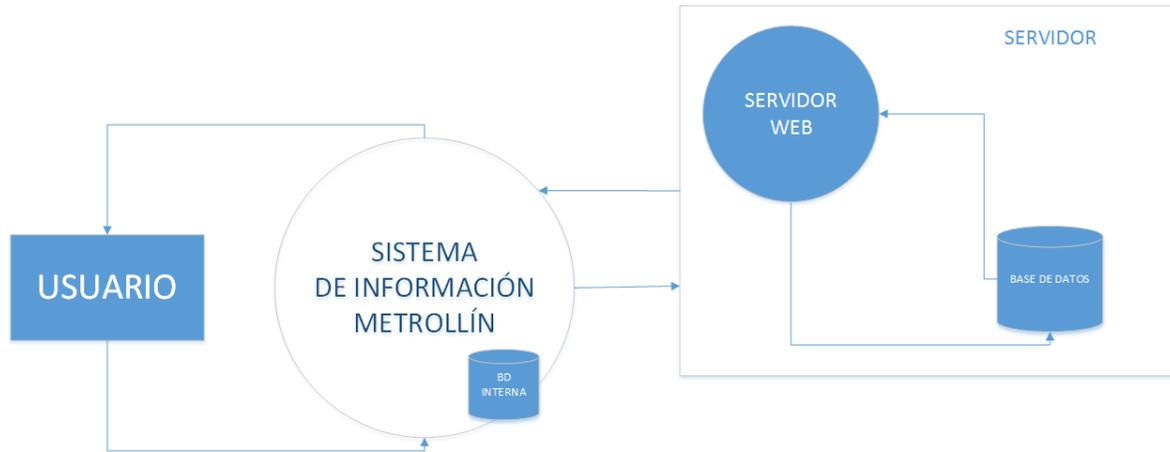
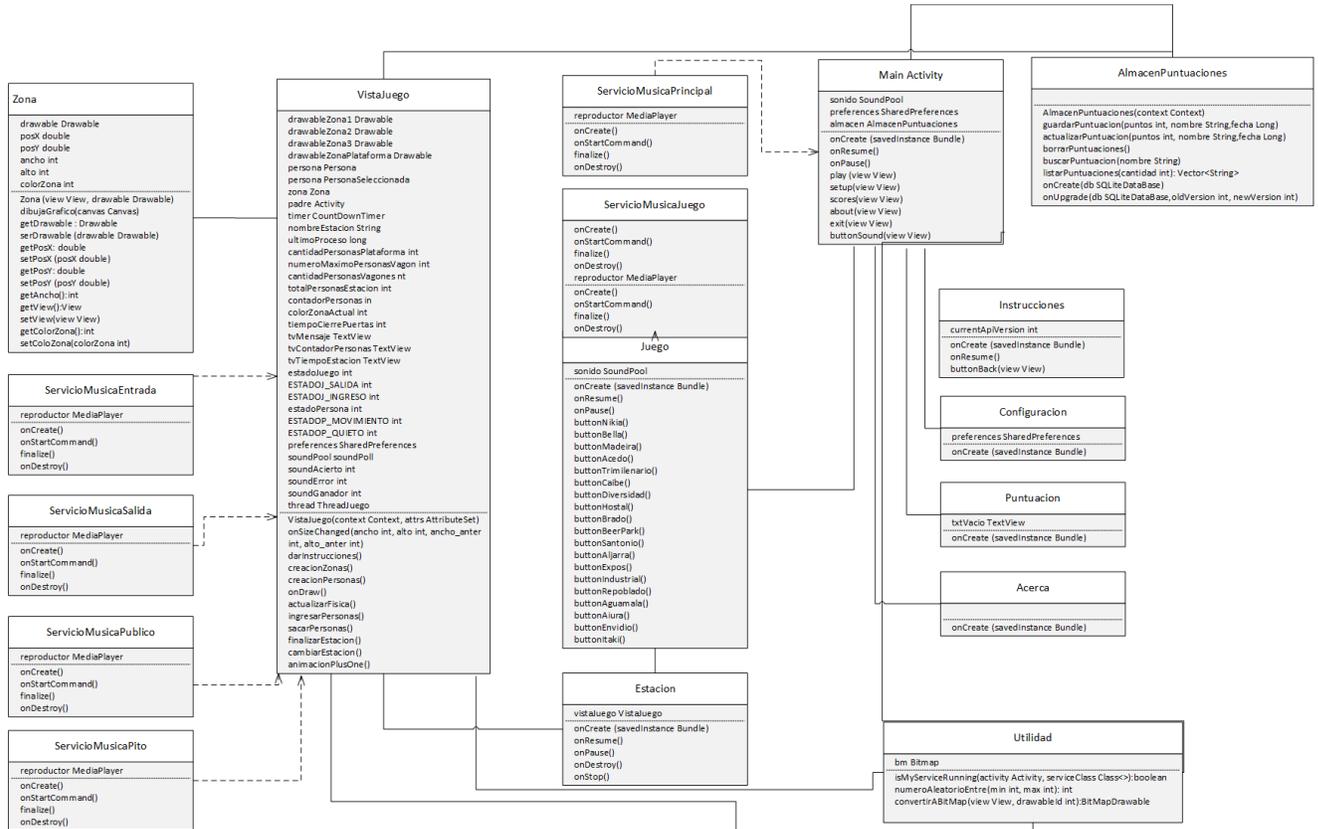
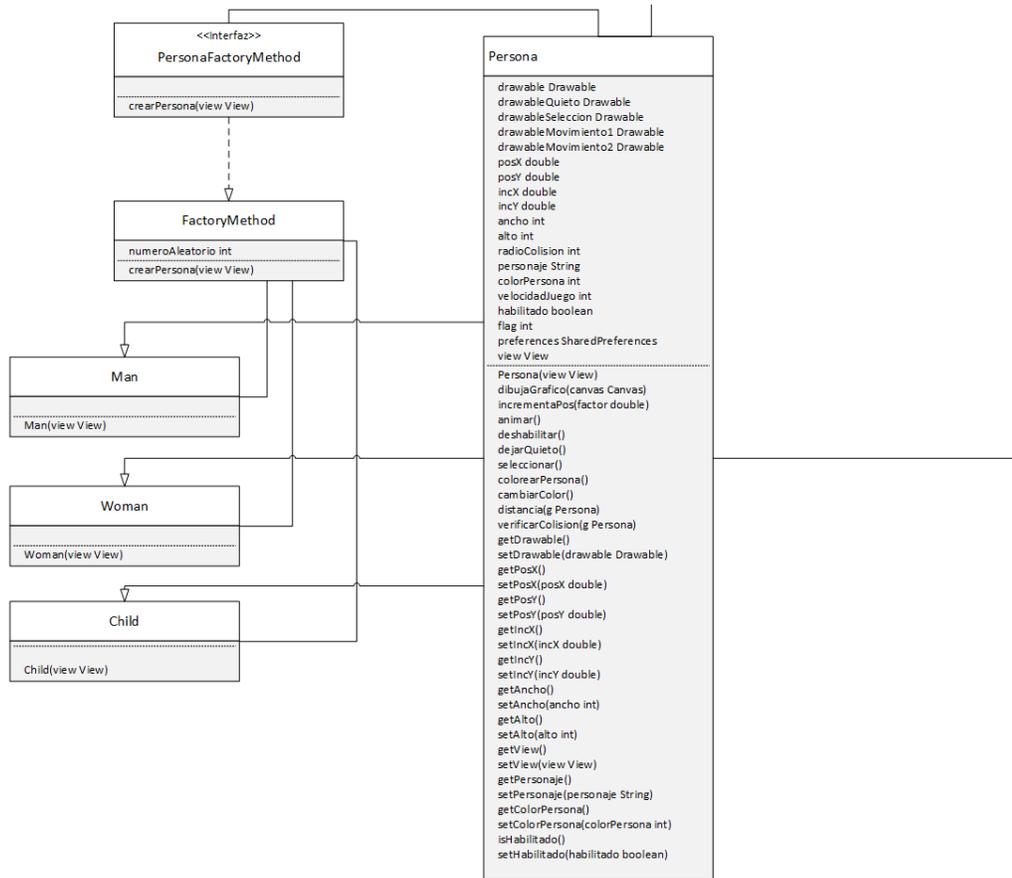


Figura 4. Diagrama de Contexto

- El usuario interactúa con el sistema de información Metrollín en su dispositivo móvil.
- El sistema de información Metrollín se comunica con una base de datos interna para el almacenamiento de datos.
- El sistema de información Metrollín se comunica con un servidor web para el envío y recibimiento de datos.
- El servidor web se comunica con una base de datos para el almacenamiento y extracción de datos.

### 3.2.2 Diagrama de Clases





	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3 Casos de Uso

#### 3.2.3.1 Ingresar a la aplicación

<b>Caso de uso:</b> Ingresar a la aplicación
<b>Actores: Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Utilizar el sistema de información para su entretenimiento.
<b>Propósito:</b> Ingresar al sistema de información para su entretenimiento.
<b>Resumen:</b> El usuario ingresa a la aplicación en su dispositivo móvil, para jugar.
<b>Precondiciones:</b> Haber instalado la aplicación.
<b>Flujo Básico:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación, a través del icono Metrollín en su dispositivo móvil.</li> <li>2. El usuario visualiza la pantalla principal de Metrollín.</li> </ol>
<b>Flujo Alterno:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación, a través del icono Metrollín en su dispositivo móvil.</li> <li>2. La aplicación se resumen en la última pantalla donde se encontraba el usuario.</li> </ol>

Tabla 3. Case de uso Ingresar a la aplicación

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.2 Presionar Jugar

<b>Caso de uso:</b> Presionar Jugar
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Ingresar a la lista de estaciones del juego.
<b>Propósito:</b> Ingresar a la pantalla jugar, para seleccionar una estación.
<b>Resumen:</b> El usuario presiona el botón jugar, para visualizar las diferentes estaciones del metro.
<b>Precondiciones:</b> Haber ingresado a la aplicación.
<b>Flujo Básico:</b>  1. El usuario ingresa a la pantalla Jugar, al presionar el botón Jugar en el menú principal. 2. El usuario visualiza las estaciones del metro.
<b>Flujo Alternativo:</b>  1. El usuario ingresa a la pantalla Jugar, al presionar el botón Jugar en el menú principal. 2. La App no responde. 3. La aplicación se cierra.

Tabla 4. Caso de uso Presionar Jugar

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.3 Seleccionar Estación

<b>Caso de uso:</b> Seleccionar Estación
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Seleccionar una estación de la lista.
<b>Propósito:</b> Ingresar a la pantalla estación para interactuar con los vagones y pasajeros del juego.
<b>Resumen:</b> El usuario ingresa a la pantalla estación, en donde visualiza la plataforma, vagones y pasajeros del juego.
<b>Precondiciones:</b> Haber ingresado a la pantalla Juego.
<b>Flujo Básico:</b>  1. El usuario ingresa a la pantalla Estación, al presionar una estación de la lista.  2. El usuario visualizar la pantalla Estación, que contiene la plataforma, vagones, y pasajeros.
<b>Flujo Alterno:</b>  1. El usuario ingresa a la pantalla Estación, al presionar una estación de la lista.  2. La App no responde.  3. La aplicación se cierra.

Tabla 5. Caso de uso Seleccionar Estación

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.4 Interactuar con el juego

<b>Caso de uso:</b> Interactuar con el juego
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Entretenerse
<b>Propósito:</b> Interactuar con los diferentes objetos de la estación.
<b>Resumen:</b> El usuario va a interactuar con los pasajeros de cada estación, para sacarlos e Ingresarlos de los vagones y sumar puntaje para avanzar a las demás estaciones.
<b>Precondiciones:</b> Haber seleccionado una estación.
<b>Flujo Básico:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la pantalla Estación, al seleccionar una estación.</li> <li>2. El usuario visualiza la pantalla Estación.</li> <li>3. El usuario saca los pasajeros de los vagones a la plataforma.</li> <li>4. El usuario ingresa los pasajeros de la plataforma a los vagones.</li> <li>5. El sistema cierra las puertas de los vagones para impedir que ingresen más pasajeros.</li> <li>6. El sistema muestra un mensaje con los resultados.</li> <li>7. El usuario ingreso la cantidad de pasajeros adecuados, así que el sistema le permite al usuario seguir con la siguiente estación.</li> <li>8. El sistema redirecciona el usuario a la pantalla Jugar, para seleccionar la siguiente estación.</li> </ol>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**Flujo Alternativo:**

1. El usuario ingresa a la pantalla Estación, al seleccionar una estación.
2. El usuario visualiza la pantalla Estación.
3. El usuario saca los pasajeros de los vagones a la plataforma.
4. El usuario ingresa los pasajeros de la plataforma a los vagones.
5. El sistema cierra las puertas de los vagones para impedir que ingresen más pasajeros.
6. El sistema muestra un mensaje con los resultados.
7. El usuario no ingreso la cantidad de pasajeros adecuados, así que el sistema no le permite al usuario seguir con la siguiente estación.
8. El sistema redirecciona el usuario a la pantalla Jugar, para que vuelva a repetir la estación.

**Tabla 6. Caso de uso Interactuar con el juego**

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.5 Presionar Instrucciones

<b>Caso de uso:</b> Presionar Instrucciones
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Ingresar a las instrucciones del juego.
<b>Propósito:</b> Ingresar a la pantalla instrucciones, para ver cómo se juega.
<b>Resumen:</b> El usuario presiona el botón instrucciones, para ver las reglas y normas del juego.
<b>Precondiciones:</b> Haber ingresado a la aplicación.
<b>Flujo Básico:</b>  1. El usuario ingresa a la pantalla Instrucciones, al presionar el botón Instrucciones en el menú principal.  2. El usuario visualiza las instrucciones.
<b>Flujo Alternativo:</b>  1. El usuario ingresa a la pantalla Instrucciones, al presionar el botón Instrucciones del menú principal.  2. La App no responde.  3. La aplicación se cierra.

Tabla 7. Caso de uso Presionar Instrucciones

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.6 Presionar Configuración

<b>Caso de uso:</b> Presionar Configuración
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Ingresar a la lista de configuraciones del juego.
<b>Propósito:</b> Ingresar a la pantalla configuración, para cambiar los ajustes.
<b>Resumen:</b> El usuario presiona el botón configuración, para ver los diferentes ajustes que puede modificar.
<b>Precondiciones:</b> Haber ingresado a la aplicación.
<b>Flujo Básico:</b>  1. El usuario ingresa a la pantalla Configuración, al presionar el botón configuración en el menú principal.  2. El usuario cambia los ajustes del juego.
<b>Flujo Alternativo:</b>  1. El usuario ingresa a la pantalla Configuración, al presionar el botón Configuración del menú principal.  2. La App no responde.  3. La aplicación se cierra.

**Tabla 8. Caso de uso Presionar Configuración**

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.7 Presionar Puntuación

<b>Caso de uso:</b> Presionar Puntuación
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Ingresar a la lista de puntuaciones del juego.
<b>Propósito:</b> Ingresar a la pantalla puntuación, para ver las puntuaciones de diferentes personas.
<b>Resumen:</b> El usuario presiona el botón puntuación, para ver los puntajes de otras personas o su propio puntaje.
<b>Precondiciones:</b> Haber ingresado a la aplicación.
<b>Flujo Básico:</b>  1. El usuario ingresa a la pantalla Puntuación, al presionar el botón puntuación en el menú principal.  2. El usuario ve un listado de nombres con sus puntajes.
<b>Flujo Alterno:</b>  1. El usuario ingresa a la pantalla Puntuación, al presionar el botón Puntuación del menú principal.  2. La App no responde.  3. La aplicación se cierra.

Tabla 9. Caso de uso Presionar Puntuación

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.3.8 Presionar Acerca De

<b>Caso de uso:</b> Presionar Acerca De
<b>Actores:</b> <b>Principal:</b> Usuario
<b>Interesados:</b> Usuario
<b>Intereses:</b> Ingresar a la información del juego.
<b>Propósito:</b> Ingresar a la pantalla acercade, para ver la información del juego.
<b>Resumen:</b> El usuario presiona el botón acerca de, para ver los desarrolladores del juego y más información sobre el desarrollo del juego Metrollín.
<b>Precondiciones:</b> Haber ingresado a la aplicación.
<b>Flujo Básico:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la pantalla AcercaDe, al presionar el botón acercaDe en el menú principal.</li> <li>2. El usuario ve información sobre el desarrollo del juego como los desarrolladores, asesor, software utilizado para su desarrollo y más información.</li> </ol>
<b>Flujo Alterno:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la pantalla AcercaDe, al presionar el botón acercaDe en el menú principal.</li> <li>2. La App no responde.</li> <li>3. La aplicación se cierra.</li> </ol>

Tabla 10. Caso de uso Presionar Acerca De

### 3.2.4 Diagrama de casos de uso

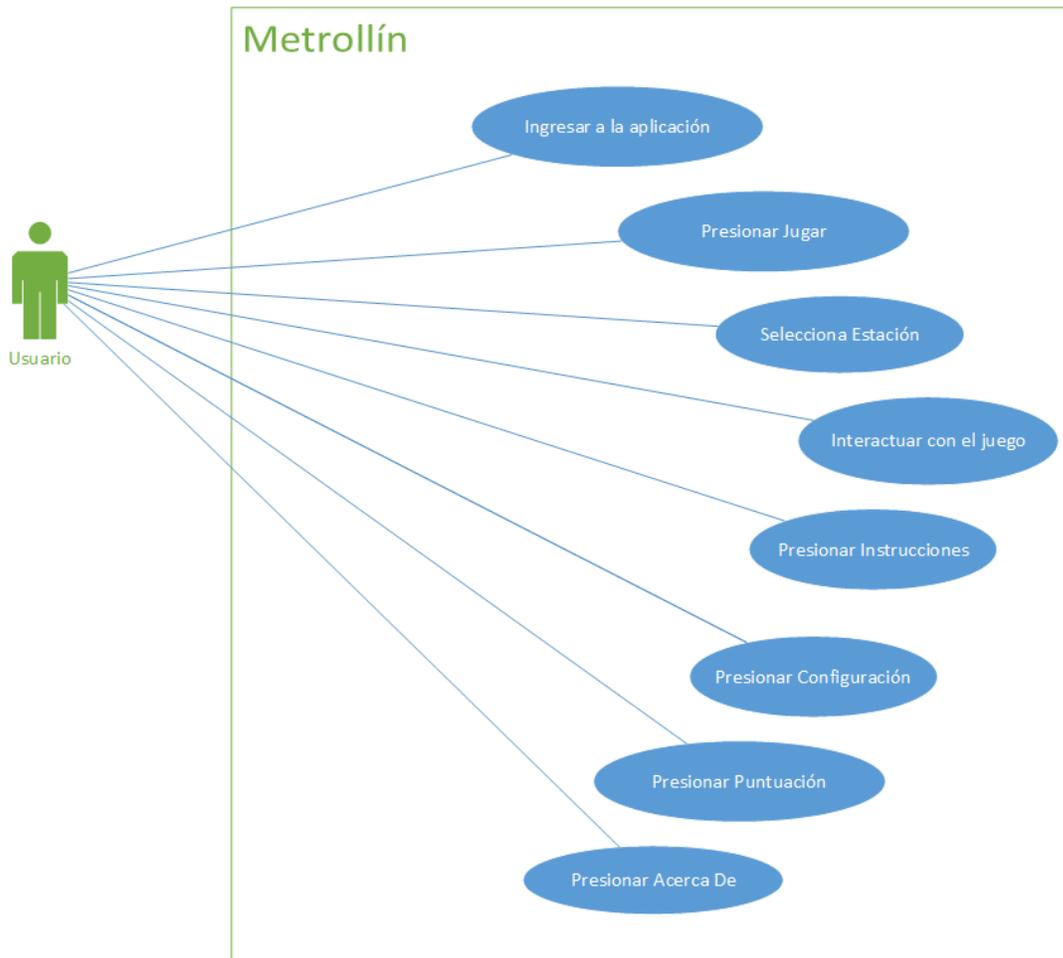


Figura 5. Diagrama de casos de uso

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.5 Arquitectura de la aplicación

A continuación se describe el método fábrica utilizado en una parte del desarrollo de la aplicación.

#### 3.2.5.1 Definición Método Fábrica

Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos. Libera al desarrollador sobre la forma correcta de crear objetos. Define la interfaz de creación de un cierto tipo de objeto, permitiendo que las subclases decidan qué clase concreta necesitan instancias. Muchas veces ocurre que una clase no puede anticipar el tipo de objetos que debe crear, ya que la jerarquía de clases que tiene requiere que deba delegar la responsabilidad a una subclase. Es similar al *Abstract Factory*, pero sin el énfasis en las familias. A este método se lo utiliza con el método plantilla.

Este patrón debe ser utilizado cuando:

Una clase no puede anticipar el tipo de objeto que debe crear y quiere que sus subclases especifiquen dichos objetos.

Hay clases que delegan responsabilidades en una o varias subclases. Una aplicación es grande y compleja y posee muchos patrones creacionales.

Diagrama UML

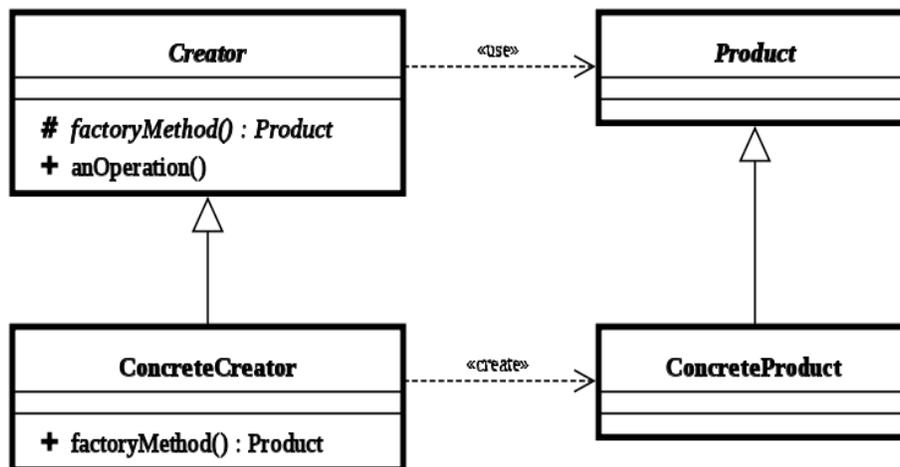


Figura 6. Diagrama UML del método Fábrica

**Creator:** declara el método de fabricación (creación), que devuelve un objeto de tipo Product.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**ConcretCreator:** redefine el método de fabricación para devolver un producto.

**ProductoConcreto:** es el resultado final. El creador se apoya en sus subclases para definir el método de fabricación que devuelve el objeto apropiado.

### 3.2.5.2 Método implementado en la aplicación

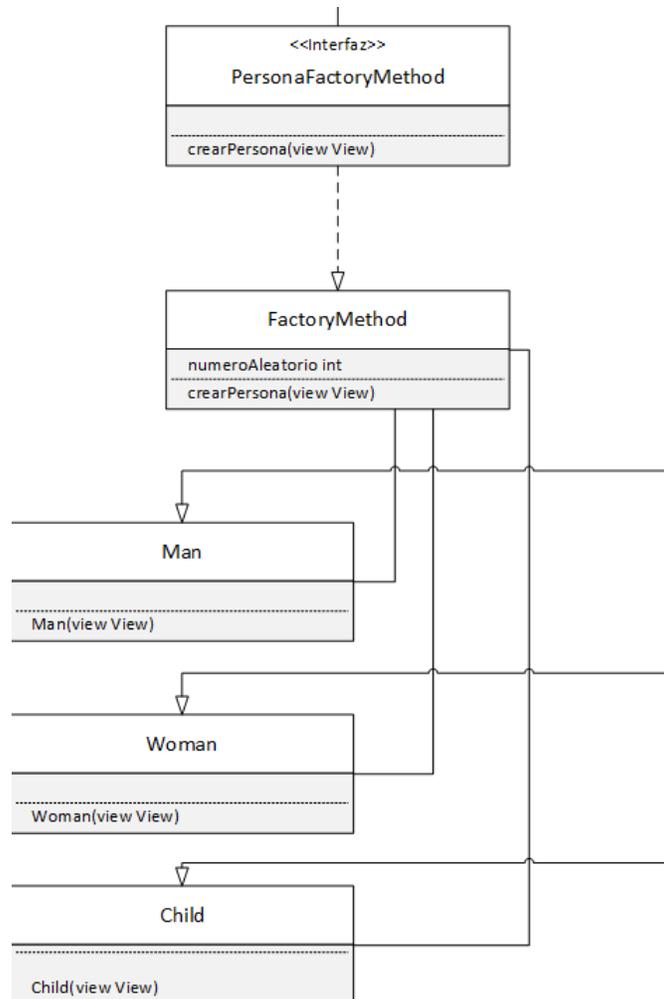


Figura 7. Diagrama UML del método Fábrica implementado

En el diagrama de clases se puede apreciar que solo es necesario instanciar un objeto de la clase *Factory Method*. Luego se ejecuta el método *crearPersonas* de esta clase, quien se encarga de crear cualquiera de los 3 tipos de personajes, liberando al desarrollador de tener que crear una persona y asignarle su velocidad, color, posición y otros atributos.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3.2.6 Diseño de Layouts

### 3.2.6.1 Pantalla principal

La pantalla principal es la primera interfaz gráfica que se le muestra al usuario una vez que ingresa a la aplicación. La pantalla principal de Metrollín cuenta con 6 botones en su centro.

Los 6 botones re direccionan a una nueva actividad, que son:

**Botón jugar:** Abre la actividad Estación, donde se le muestra al usuario las 19 estaciones del metro.

**Botón instrucciones:** Abre la actividad Instrucciones, donde se le muestra al usuario el objetivo, uso y reglas de la aplicación.

**Botón configuración:** Donde el usuario puede editar las preferencias como su nombre y la reproducción de sonido.

**Botón puntuaciones:** Donde el usuario va a ver el listado de las personas que jugaron y subieron sus registros de puntuaciones.

**Botón acerca de:** Donde el usuario va a visualizar información sobre los creadores de la aplicación, licencia e información sobre Metrollín.

**Botón salir:** Para cerrar la aplicación.



Figura 8. Menú principal propuesto



Figura 9. Menú principal final

### 3.2.6.2 Pantalla Jugar

En la **pantalla jugar** existirán dos actividades.

La primera actividad juego1 muestra la lista de estaciones del metro que deberá recorrer el usuario que vaya jugando, y que deberá seguir en el sentido izquierda derecha. El usuario deberá pasar cada una de las estaciones, de lo contrario no podría avanzar. Cuando el usuario seleccione una estación, abrirá la actividad juego2.

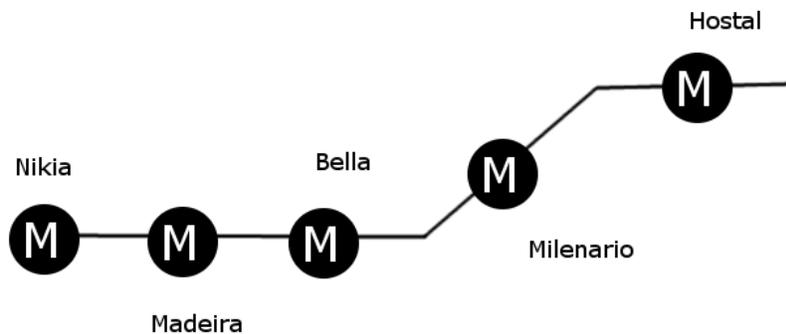


Figura 10. Selección de estaciones propuesto

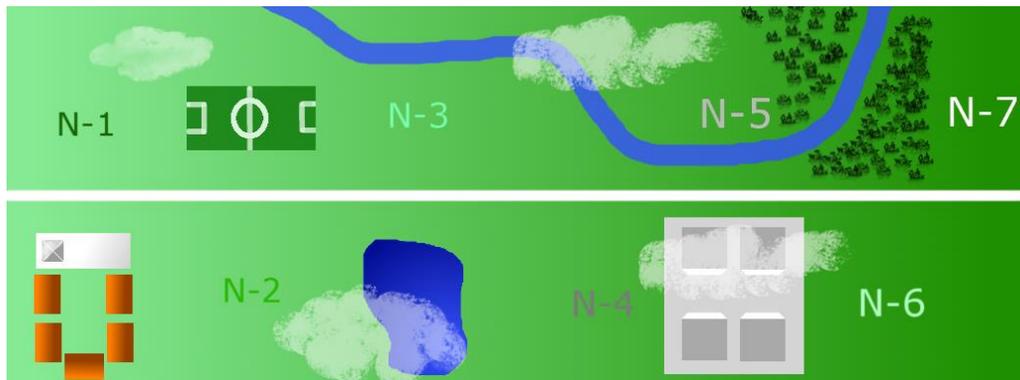


Figura 11. Selección de estaciones final

### 3.2.6.3 Pantalla Juego Estación

La segunda pantalla juego 2 mostrará la plataforma, vagones, y pasajeros de la estación que seleccione el usuario.

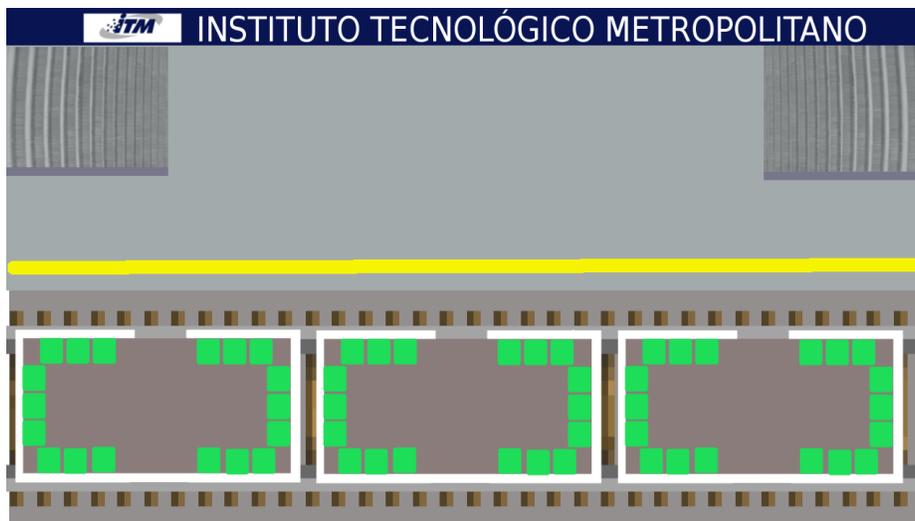


Figura 12. Pantalla Estación Propuesto

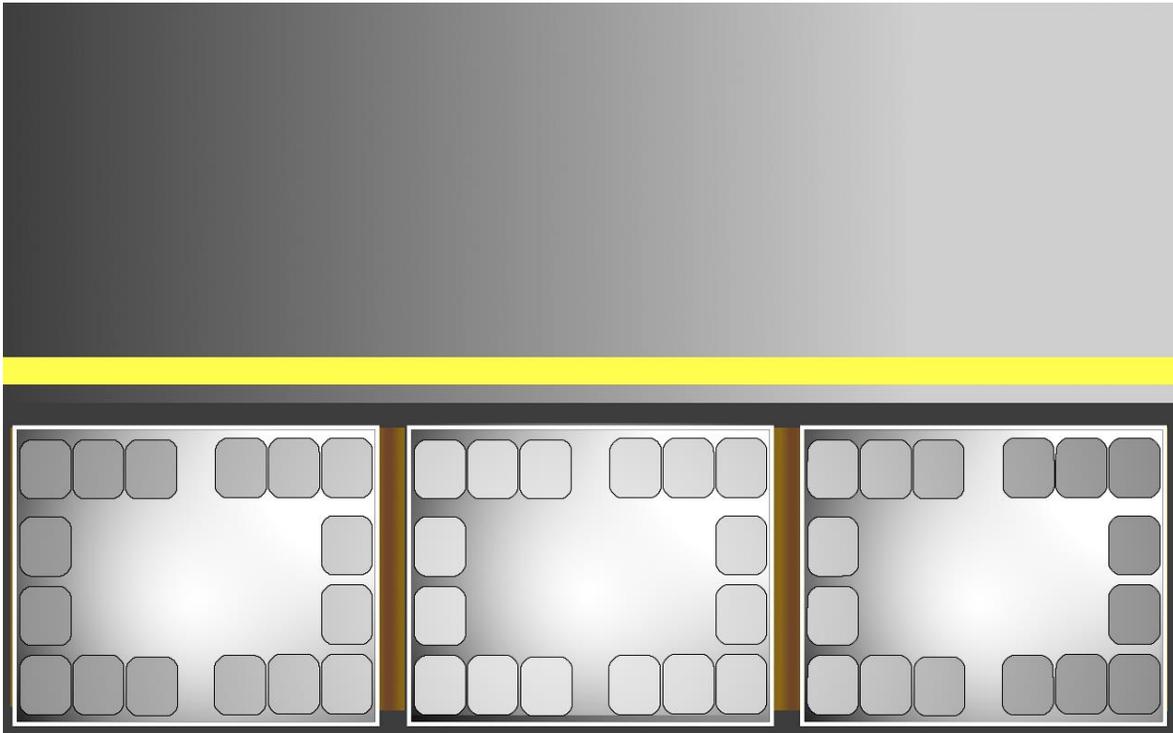


Figura 13. Pantalla Estación Final

Además de los componentes del juego, esta pantalla tiene en la parte superior, la estación en la que se encuentra el usuario, el tiempo para ingresar los pasajeros y el puntaje realizado.

### 3.2.6.4 Pantalla Instrucciones

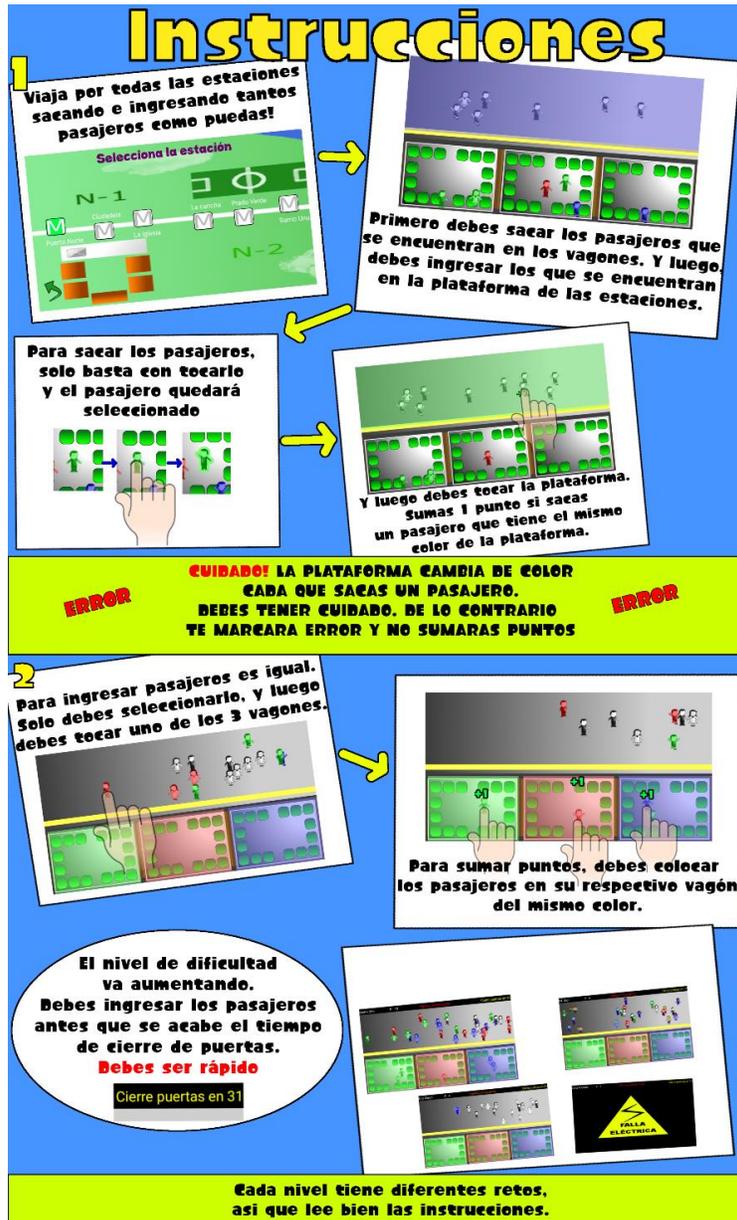


Figura 14. Imagen con las instrucciones

Pantalla que contiene en forma de historieta, las instrucciones del juego sobre su objetivo, uso, reglas y descripción sobre como jugar.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.6.5 Pantalla configuración

Pantalla donde se encuentran las preferencias del usuario como su nombre, la cantidad de pasajeros, la estación actual y un botón para activar y desactivar sonido. Además tiene una opción para restaurar los datos del juego.

Esquema propuesto:

**Nombre de usuario: Jugador**

**Reproducir sonido: SI/NO**

**Estación Actual: Puerta Norte**

**Puntuación: 100**

**Cantidad de viajes: 1**

**Restaurar Datos: SI/NO**

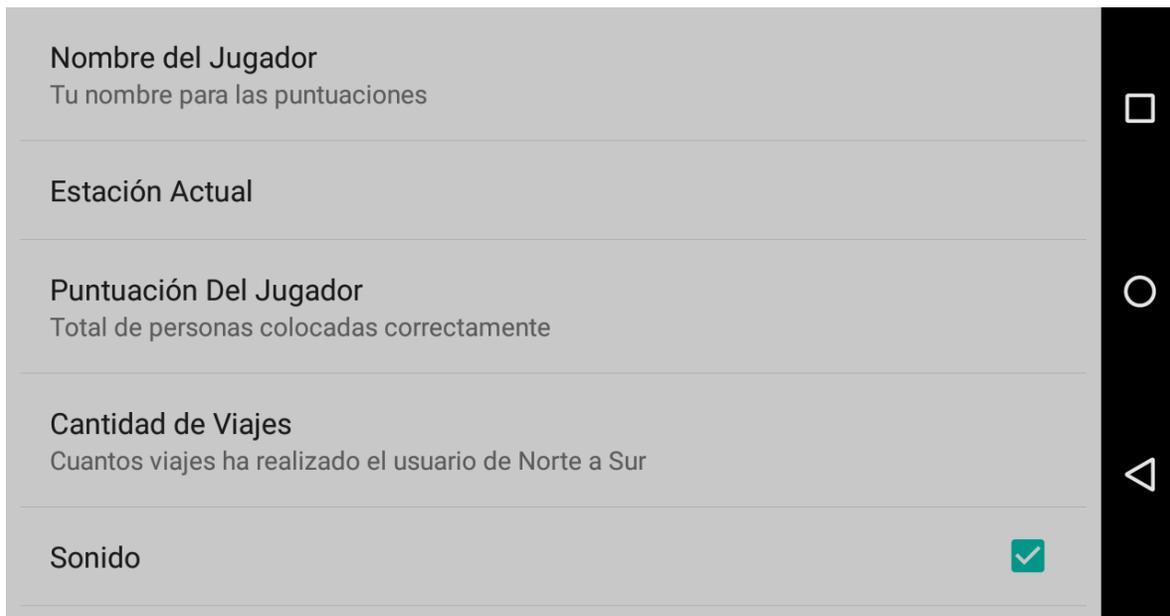


Figura 15. Pantalla Configuración final

### 3.2.6.6 Pantalla Puntuaciones

Pantalla que muestra la lista de los 10 primeros puntajes con mayor cantidad de pasajeros sacados e ingresados, realizados por diferentes personas, organizados de mayor puntaje a menor puntaje.

Esquema propuesto:

Nombre	Estación	Puntaje
Juan	Puerta Norte	1000
Edwin	Hostal	1500

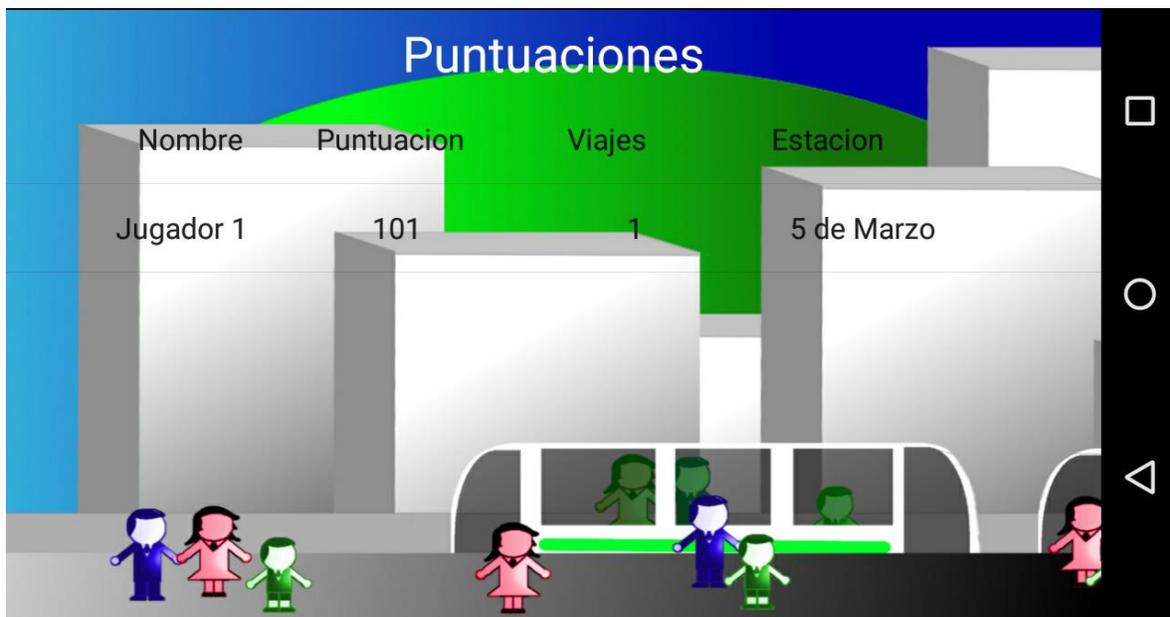


Figura 16. Pantalla Puntuaciones final

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.6.7 Pantalla Acerca de

Pantalla que contiene información de la aplicación como los desarrolladores, descripción y la licencia de la aplicación.

**ACERCA DE**

**Metrollín es un videojuego creado por estudiantes del ITM, para su proyecto de grado.**  
**Todas las imágenes son de la autoría de los estudiantes.**  
**Las referencias musicales se encuentran listadas a continuación:**

Figura 17. Pantalla Acerca De propuesto

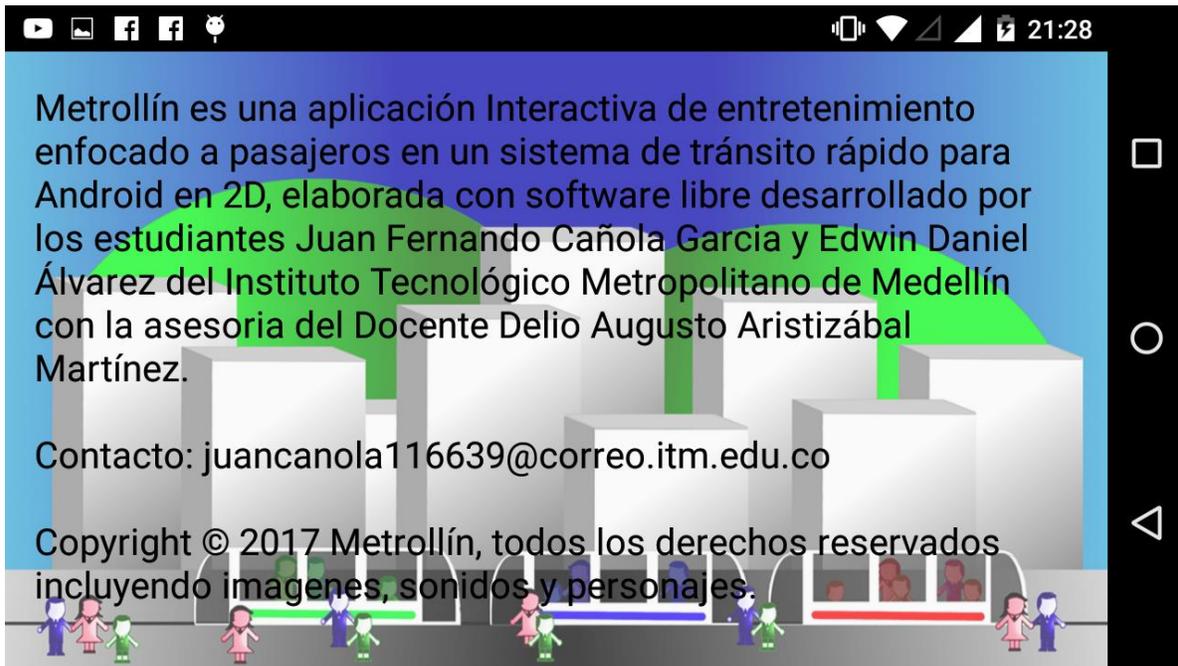


Figura 18. Pantalla Acerca De Final

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.6.8 Pantalla Transición

Pantalla que contiene una imagen, la cual es animada y utilizada para el ingreso y salida de los vagones. Para esta pantalla se creó un código que desplaza la imagen, haciendo creer al usuario que los vagones se desplazan hacia la derecha o hacia la izquierda.



Figura 19. Pantalla transición de vagones

### 3.2.7 Diseño de imágenes

**GIMP** es el editor de imágenes libre por excelencia. Con este programa se crearon todas las imágenes necesarias para el sistema de entretenimiento.

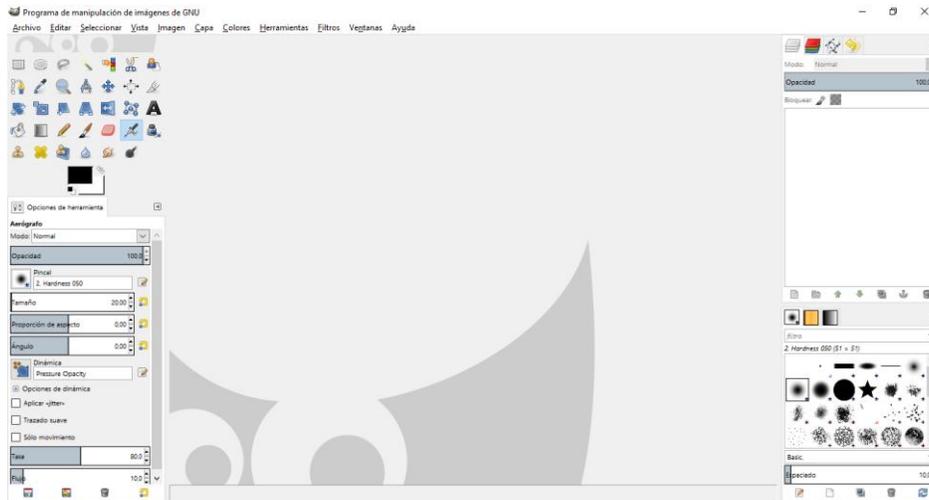


Figura 20. Interfaz gráfica de GIMP

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.7.1 Icono

El icono de Metrollín consiste de una montaña verde en el fondo, el edificio de la alcaldía de Medellín y el edificio Coltejer, edificios simbólicos de la ciudad de Medellín, rodeados de una vía férrea del metro. Los edificios ocultando la montaña, forman la letra M de Metrollín.



Figura 21. Icono de la aplicación propuesto



Figura 22. Icono de la aplicación final

### 3.2.7.2 Botones

#### 3.2.7.2.1 Botones del menú principal



Figura 23. Botones del menú principal

Los botones del menú principal son de color amarillo, para que contraste con el paisaje de fondo de la pantalla principal.

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.7.2.2 Botón volver

El botón volver para regresar a la pantalla anterior, que al ser presionado, cambiar a color amarillo.



Figura 24. Imagen botón volver

### 3.2.7.2.3 Botón sonido

Los botones para activar y desactivar la música.



Figura 25. Imagen Botón activar y desactivar sonido

### 3.2.7.2.4 Botón estación

El botón para seleccionar la estación es verde y blanco, es animado y cambia al color azul al ser presionado. El de las demás estaciones es de color blanco con gris, que indica que están deshabilitadas.



Figura 26. Imagen botón animado selección de estación

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.7.3 Personajes

#### 3.2.7.3.1 Personaje Man

El personaje man, tiene traje elegante, sus movimientos son movimiento de cabeza y brazos. Al ser seleccionado tiene un borde que resalta.



Figura 27. Drawables Personaje Man

#### 3.2.7.3.2 Personaje Woman

El personaje woman, tiene traje elegante con falda. Sus movimientos son de cabeza y cadera. Al ser seleccionada tiene un borde que resalta.



Figura 28. Drawables Personaje Woman

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.7.3.3 Personaje Child

El personaje child tiene pantalón corto. Sus movimientos son de brazos y piernas. Al ser seleccionado contiene un borde que resalta.

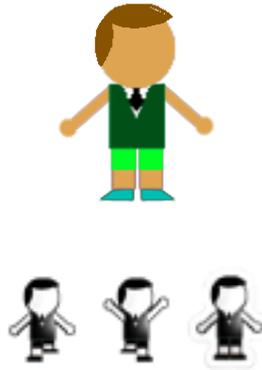


Figura 29. Drawables Personaje Child

### 3.2.7.4 Otros Objetos

#### 3.2.7.4.1 Zonas

Las zonas se componen de 1 pixel, con un 50% de transparencia y luego en una de las clases de Android, se editan y crean las zonas de acuerdo al tamaño de la pantalla del dispositivo móvil.



Figura 30. Drawable Zona

#### 3.2.7.4.2 PlusOne

Animación mostrada cuando es colocado un pasajero en una zona que corresponde con su color.



Figura 31. Gif Animado Plus One

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.7.4.3 Error

Animación mostrada cuando es colocado un pasajero en una zona que no corresponde con su color.



Figura 32. Gif Animado Error

### 3.2.7.4.4 Problema eléctrico

Imagen que aparece en el sexto nivel del juego, para simular problemas de electricidad en el juego.



Figura 33. Imagen Falla Eléctrica

## 3.2.8 Efectos de sonido

La música del juego es desarrollada con el software Linux multimedia Studio. LMS es un software de creación de audio digital de software libre y multiplataforma para la creación de música y sonidos.

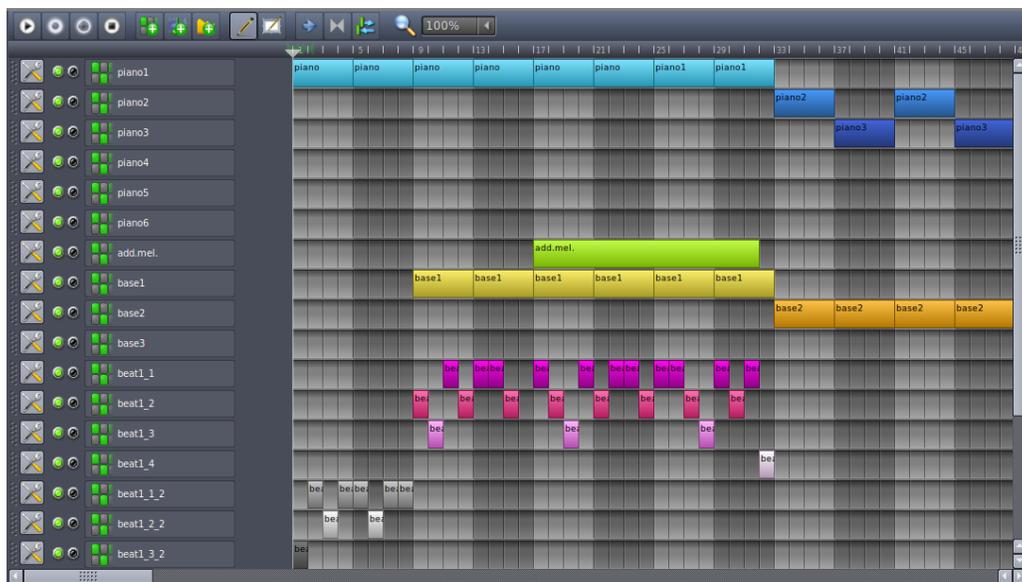


Figura 34. Interfaz gráfica Linux Multimedia Studio

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.8.1 Música

El videojuego cuenta con dos canciones de fondo para la pantalla Principal y la pantalla de Estación.

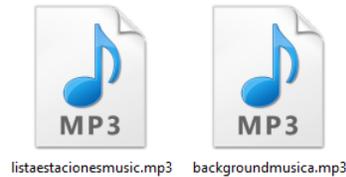


Figura 35. Imagen Archivos de música

### 3.2.8.2 Efectos de sonido

El videojuego cuenta con 14 sonidos para los efectos:

1. Sonido de selección al presionar botones.
2. Sonido al colocar una persona correctamente.
3. Sonido al colocar una persona incorrectamente.
4. Sonido al finalizar una estación satisfactoriamente.
5. Sonido al finalizar una estación incorrectamente.
6. Sonido al presionar un personaje Man.
7. Sonido al presionar un personaje Woman.
8. Sonido al presionar un personaje Child.
9. Sonido de ingreso de Metro.
10. Sonido de Salida de Metro.
11. Sonido de público en la estación.
12. Sonido de cierre de puertas.
13. Sonido de cambio de color de personajes.
14. Sonido de falla eléctrica.

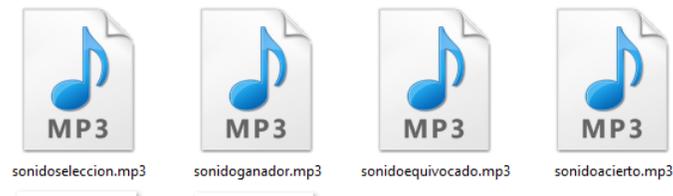


Figura 36. Imagen archivos efectos de sonido

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.2.9 Diseño de Bases de Datos

A continuación se describe el Modelo Relacional, donde se almacenaran todos los puntajes de los diferentes jugadores.

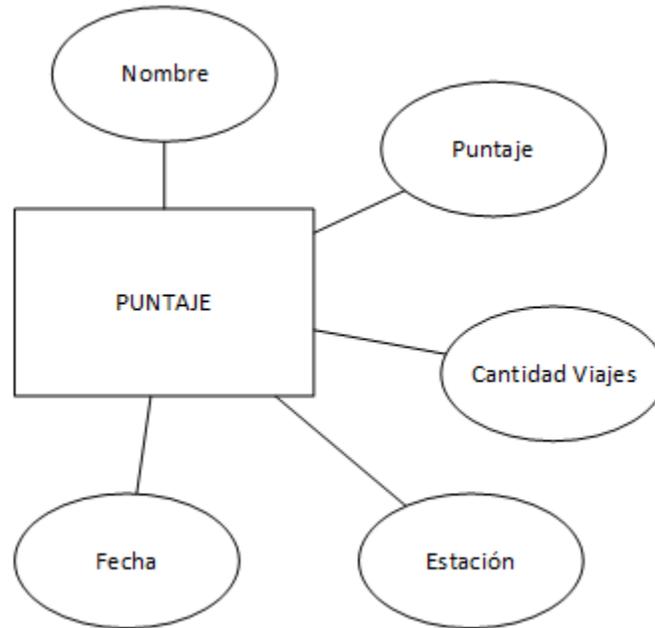


Figura 37. Entidad Puntaje

El sistema de información Metrollín consiste de 2 bases de datos. Una interna del dispositivo móvil y otra externa.

La base de datos interna de Android se compone de 1 tabla PUNTUACIONES con 6 columnas:

- **id** para identificar un usuario y que no se repita.
- **puntos** guardar la cantidad de pasajeros ingresados y sacados correctamente.
- **nombre** del jugador.
- **viajes** para guardar la cantidad de viajes que el jugador ha hecho desde la primera hasta la última estación.
- **estación** para guardar la estación actual del usuario.
- **fecha** para guardar el día y hora que se guardó la puntuación.

#### Script de creación de la entidad

```
CREATE TABLE puntuaciones(_id INTEGER PRIMARY KEY AUTOINCREMENT, puntos INTEGER, nombre TEXT, fecha LONG, viajes INTEGER, estacion TEXT))
```

La otra base de datos se encuentra en un servidor, gestionada con software MySQL la cual almacena los puntajes de todos los jugadores de Metrollín.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Esta base de datos se compone de 1 tabla PUNTUACIONES con 6 columnas:

- **id:** Contiene el ID de cada dispositivo móvil.
- **puntos:** Para la cantidad de pasajeros ingresados y sacados correctamente.
- **nombre:** Para el nombre del jugador.
- **viajes:** Para la cantidad de viajes que el jugador ha realizado desde la primera hasta la última estación.
- **estación:** para la estación actual del usuario.
- **fecha:** para el día y hora que se guardó la última puntuación.

#### Script de creación de la entidad

```
CREATE TABLE `db_metrollin`.`puntuaciones` (
  `id` VARCHAR(50) NOT NULL COMMENT "",
  `puntos` INT NOT NULL COMMENT "",
  `nombre` VARCHAR(50) NOT NULL COMMENT "",
  `fecha` DATETIME NOT NULL COMMENT "",
  `viajes` INT NOT NULL COMMENT "",
  `estacion` VARCHAR(50) NOT NULL COMMENT "",
  PRIMARY KEY (`id`) COMMENT "")
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci
COMMENT = 'Entidad Puntuaciones para android';
```

## 3.3 Desarrollo, Integración y Publicación

### 3.3.1 Estructura final del proyecto

Se utilizaron 23 clases y una interface. Las 5 primeras clases más 1 interface pertenecen a la arquitectura del método fábrica para la creación de personajes.

La clase Estacion y VistaJuego son las clases más complejas ya que contienen todas las instrucciones, integración y funcionamiento del sistema.

La clase Zona permite crear los objetos de las 4 zonas: zona verde, zona azul, zona roja y zona plataforma.

La carpeta servicios contiene los efectos de sonido del metro y música de fondo.

La carpeta útil contiene la clase para la creación de la base de datos, una clase para la animación y una clase Utilidad que contiene métodos útiles como retornar un número aleatorio.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Las últimas clases le pertenecen a cada una de las actividades del juego (una actividad por Layout: *Main Activity, Juego, Instrucciones, Configuración, Puntuación* y *Acerca De*).

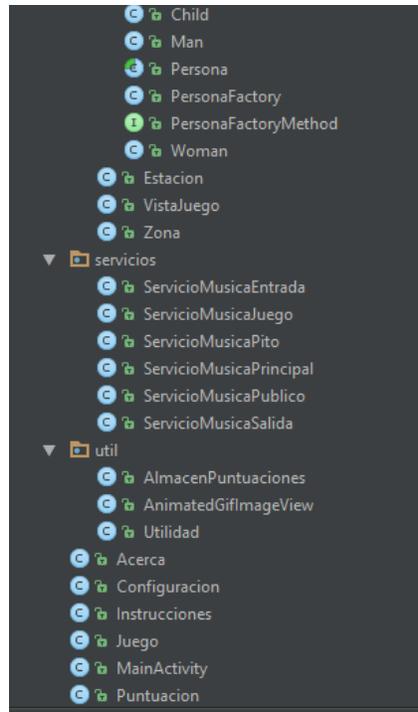


Figura 38. Diagrama de clases final

### 3.3.2 Layouts

La figura 40, muestra los Layouts necesarios para la ejecución del sistema de entretenimiento. Cada Layout se encuentra en 7 diferentes carpetas, para que pueda ser adaptado correctamente a diferentes resoluciones de pantallas: ldpi, mdpi, hdpi, xhdpi, xxhdpi y xxxhdpi y por defecto.

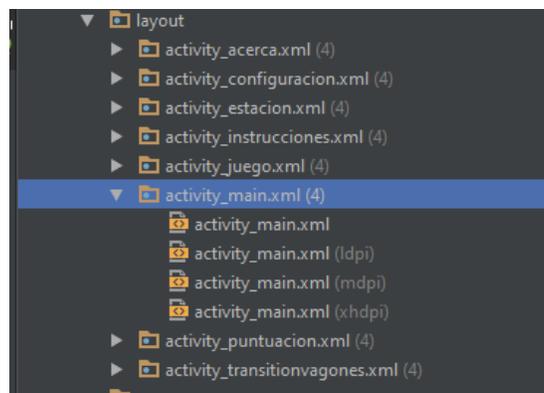


Figura 39. Lista de layouts final

### 3.3.3 Imágenes sistema de entretenimiento finalizado



Figura 40. Pantalla selección estación final 1

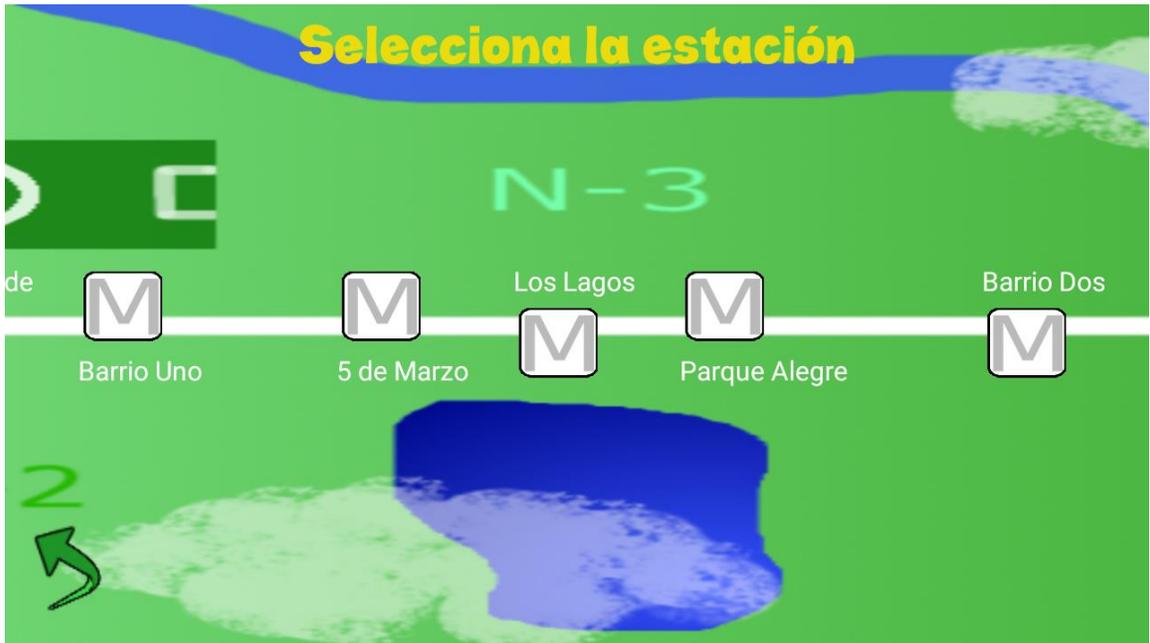


Figura 41. Pantalla selección estación final 2

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

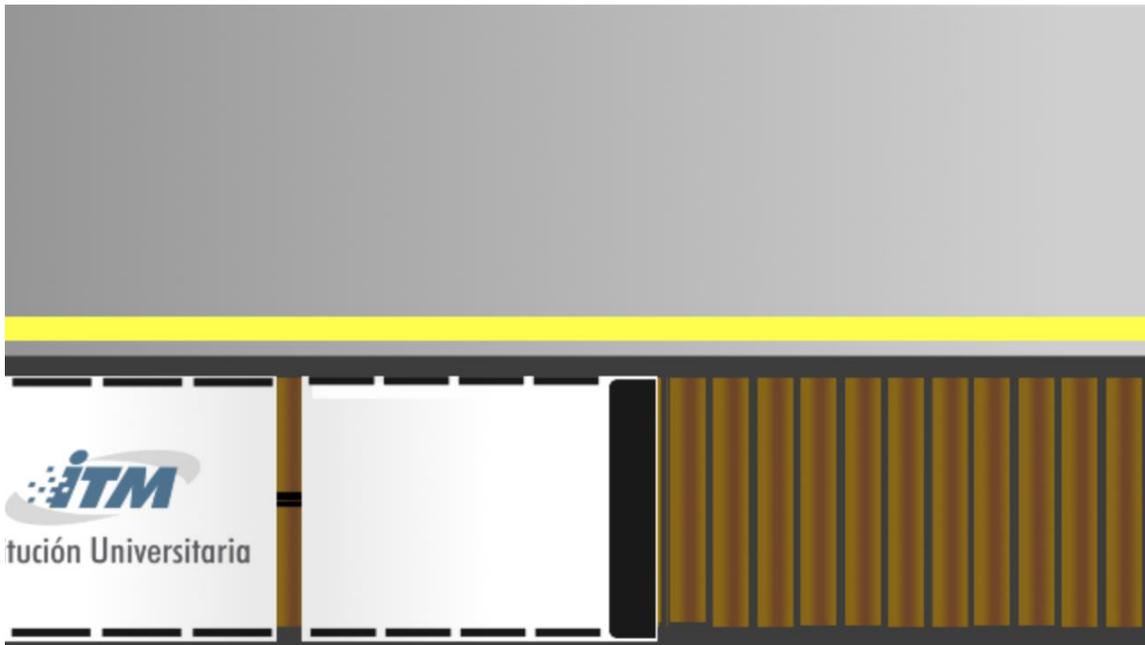


Figura 42. Pantalla estación final 2

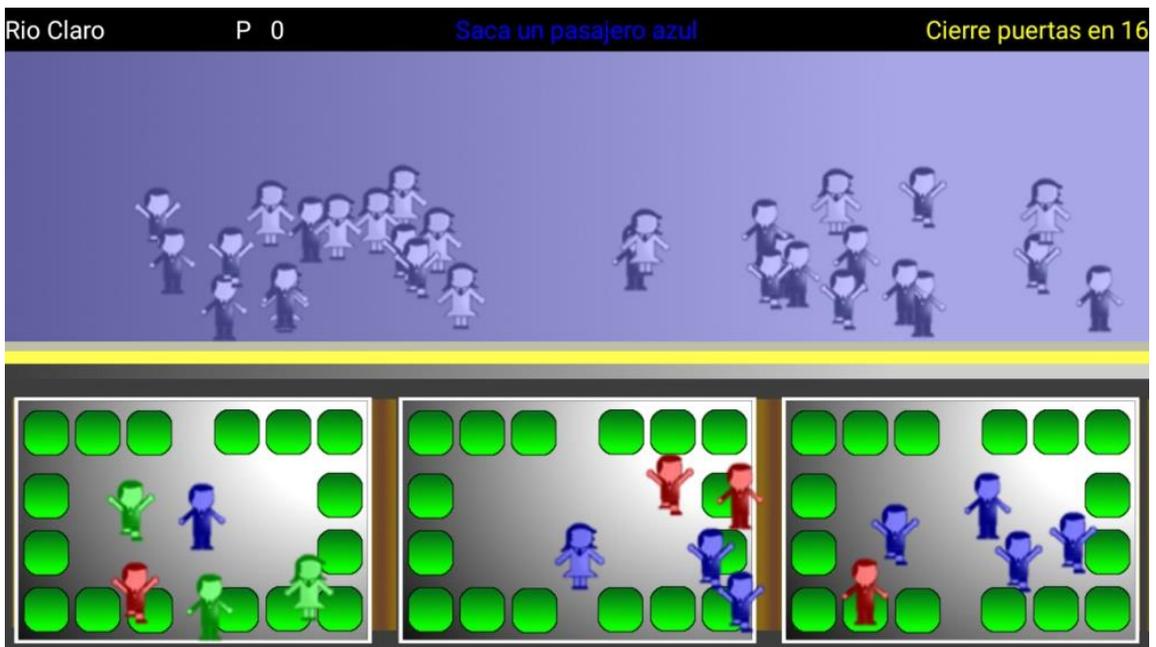


Figura 43. Pantalla estación final 3

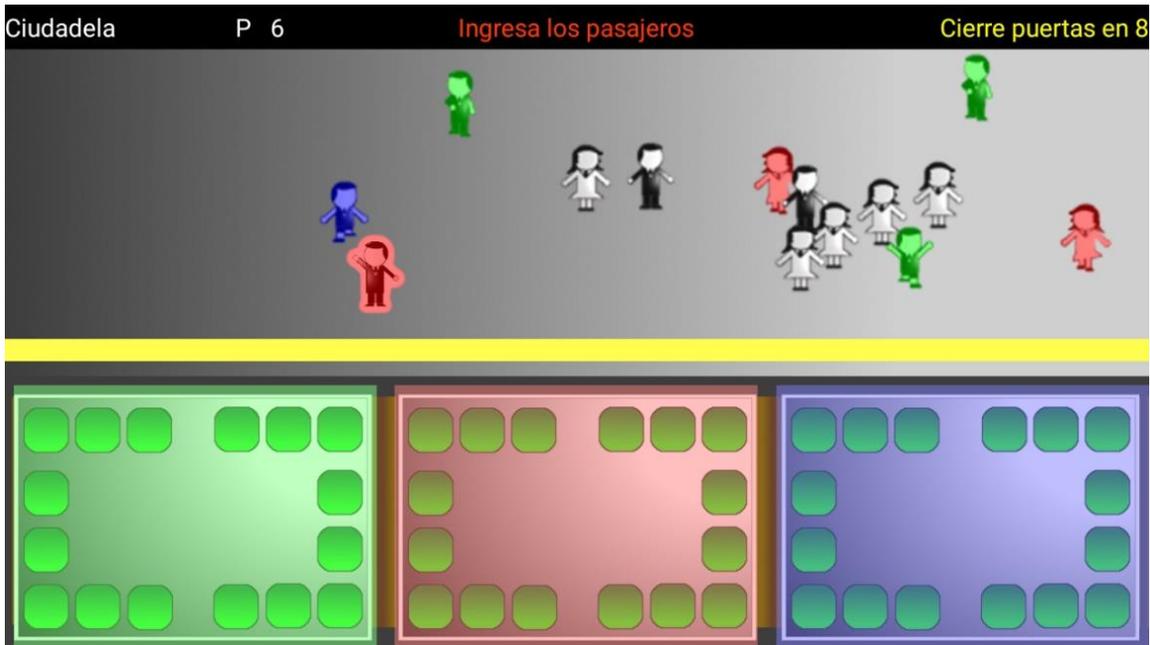


Figura 44. Pantalla estación final 4



Figura 45. Pantalla estación final 5

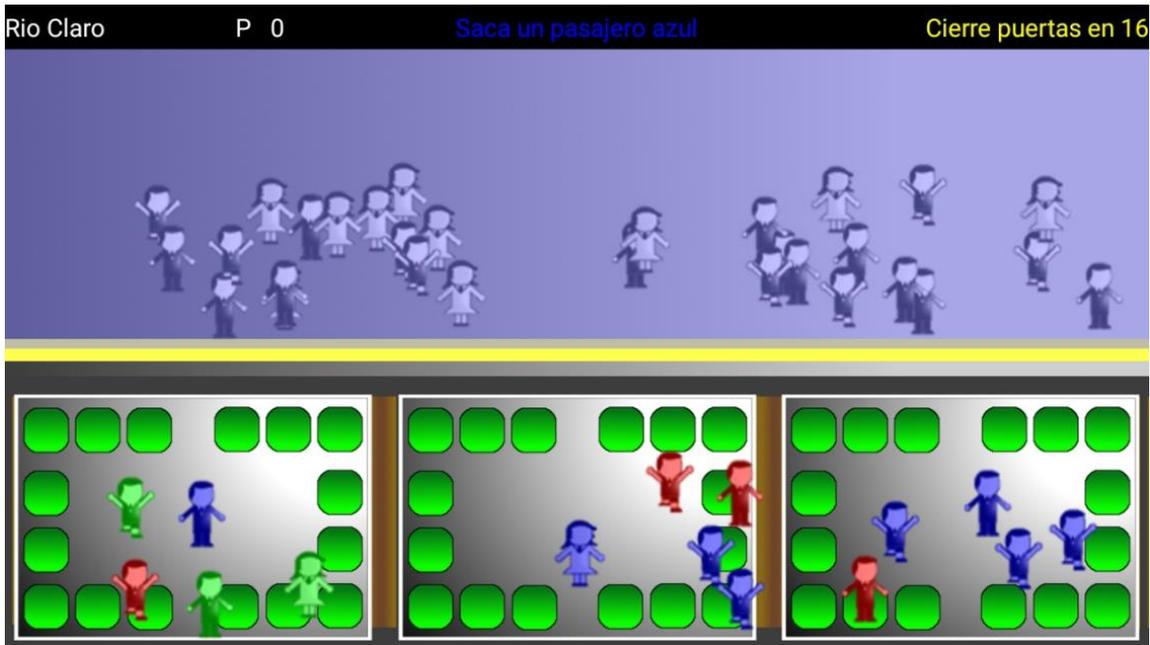


Figura 46. Pantalla estación final 6



Figura 47. Pantalla estación final 7

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

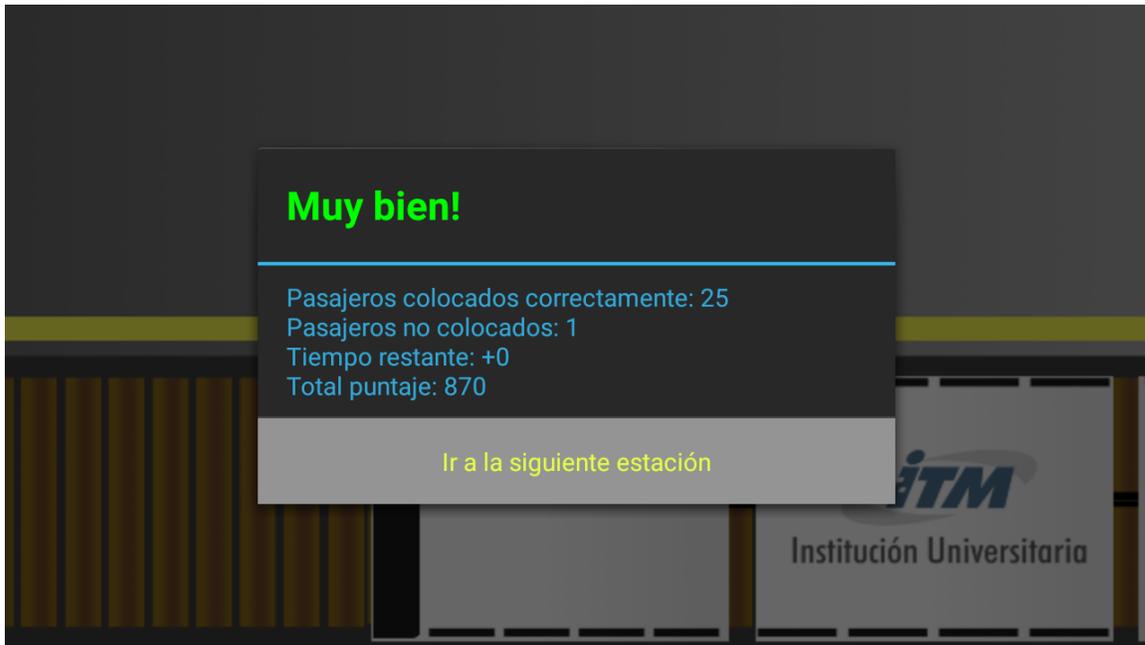


Figura 48. Pantalla estación final 8



Figura 49. Pantalla estación final 9

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3.4 Servicios Web

Un servicio web es un conjunto de reglas y tecnologías para el intercambio de la información a través de la red. Una de las principales ventajas de esta tecnología es que el intercambio de la información se realiza a través de un formato único que puede ser xml o *JSON*, de acuerdo a la tecnología que se use, sea *SOAP* o *REST*.

En la consulta sobre herramientas de software libre a utilizar, encontramos que era muy sencillo realizarlo con un servidor web para *PHP* y la base de datos *MySQL*, porque son las tecnologías más utilizadas en el desarrollo de páginas web, de las cuales tenemos conocimiento. Lo único que se necesitó, fue pagar un hosting que proporcionara este servicio, por un valor de 60 dólares.

El formato para el intercambio de información elegido fue *JSON*, el cual se caracteriza por su simplicidad y rápida transmisión de datos.

### 3.4.1 Web Service ingresarUsuario

Web Service encargado de insertar los datos del usuario en la base de datos.

El web Service se encarga de recopilar los datos del usuario y enviarlos a través de un objeto *JSON* al servidor *PHP*, en el siguiente formato:

```
[{"nombre":"Juansmash","puntos":"126","viajes":"1","estacion":"Los Lagos"}]
```

El servidor web se encarga de recoger dato por dato y realizar la inserción a la base de datos.

### 3.4.2 Web Service generarUsuarios

Web Service encargado de retornar la lista de los 20 puntajes más altos, almacenados en la base de datos.

El web Service lista el puntaje, nombre, viajes y estación en el siguiente formato:

```
[{"nombre":"Juansmash","puntos":"126","viajes":"1","estacion":"Los Lagos"},
{"nombre":"Jugador 1","puntos":"83","viajes":"1","estacion":"5 de Marzo"},
{"nombre":"Joan Smash","puntos":"66","viajes":"1","estacion":"Barrio Uno"},
{"nombre":"SMaria","puntos":"7","viajes":"1","estacion":"Ciudadela"},
{"nombre":"Jugador 1","puntos":"4","viajes":"1","estacion":"Ciudadela"}]
```

Cuando esta información es recibida por el dispositivo móvil, es almacenada en una variable para luego listarla en la Actividad Puntuaciones.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 3.6 Aplicación en la Play Store

### 3.6.1 Subir aplicación a la tienda de Apps Play Store

Para subir la aplicación a la tienda de aplicaciones, se requiere una cuenta de programador, la cual tiene un costo de \$25 dólares, los cuales son pagados con una tarjeta de crédito.

Una vez que la cuenta es activada y verificada, la página redirecciona a la consola de desarrollador de Android, en donde se encuentran todas las opciones y herramientas para subir la aplicación.

Para subir la aplicación, se necesita el apk, que es generado por Android Studio.

Para generar el apk, se hace mediante la opción *Generate Signed APK*, la cual, nos pide que firmemos la aplicación con un certificado digital, el cual es generado mediante el asistente de *Android Studio*.

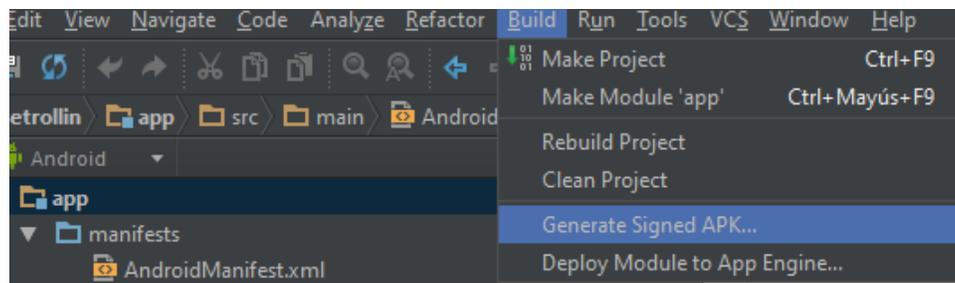


Figura 50. Opción para generar el APK

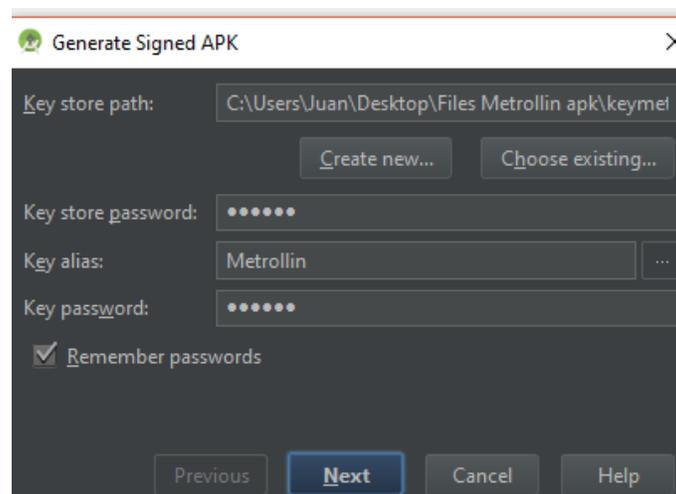


Figura 51. Asistente para la generación del APK

Y una vez completado los pasos, se tendrá el apk generado en la carpeta App del proyecto.

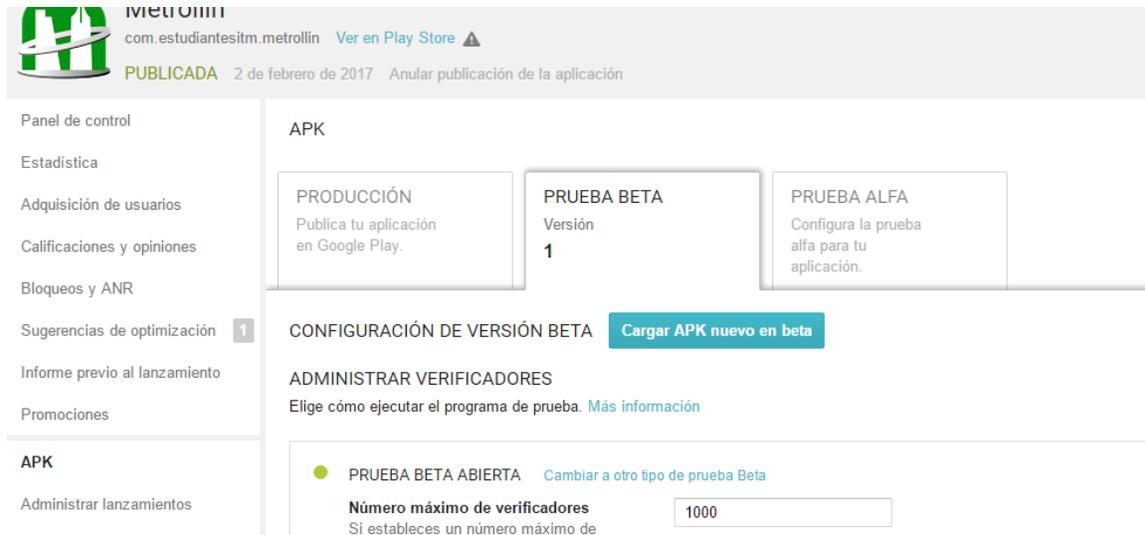
 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Nombre	Fecha de modifica...	Tipo	Tamaño
build	30/01/2017 11:48 a...	Carpeta de archivos	
libs	11/08/2016 4:46 p...	Carpeta de archivos	
src	11/12/2016 1:50 p...	Carpeta de archivos	
.gitignore	11/08/2016 4:46 p...	Archivo GITIGNORE	1 KB
app.iml	01/02/2017 2:10 a ...	Archivo IML	9 KB
app-release.apk	03/02/2017 1:11 a ...	Archivo APK	9.170 KB
build.gradle	01/02/2017 2:10 a ...	Archivo GRADLE	1 KB
proguard-rules.pro	11/08/2016 4:46 p...	Archivo PRO	1 KB

Tipo: Archivo APK  
 Tamaño: 8,95 MB  
 Fecha de modificación: 03/02/2017 1:11 a. m.

Figura 52. Apk generado en el explorador

Y con el apk generado, se procede a hospedarlo en la *Play Store*, donde la consola nos ofrece las opciones de subir una versión *Alpha*, *Beta* o publicar la aplicación final. También aparecen las opciones para subir el icono e imágenes para promocionar la aplicación.



The screenshot shows the Google Play Console interface for the application 'metrolini' (com.estudiantesitm.metrollin). The application is marked as 'PUBLICADA' (Published) on February 2, 2017. The interface includes a sidebar with navigation options like 'Panel de control', 'Estadística', and 'APK'. The main content area shows the 'APK' section with three options: 'PRODUCCIÓN' (Publish to Google Play), 'PRUEBA BETA' (Beta Testing) with version '1', and 'PRUEBA ALFA' (Alpha Testing). Below this, there is a 'CONFIGURACIÓN DE VERSIÓN BETA' section with a 'Cargar APK nuevo en beta' button, and an 'ADMINISTRAR VERIFICADORES' section with a 'Número máximo de verificadores' set to 1000.

Figura 53. Interfaz gráfica de la consola de desarrollador

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3.6.2 Pruebas a la aplicación en la Play Store



Figura 54. App Metrollín Beta en la Play Store

Con la aplicación en la *Play Store*, se proceden a realizar las pruebas, pidiéndoles a nuestros amigos y cercanos, que descarguen la aplicación para visualizar el comportamiento y funcionamiento de la aplicación en diferentes dispositivos móviles.

### 3.6.3 Actualización y mantenimiento

Una vez que la aplicación ha sido descargada y probada en diferentes dispositivos, se corrigen y se realizan algunas modificaciones.

Algunos problemas encontrados fueron:

- Imágenes más grandes o más pequeñas por las diferentes densidades.
- Imágenes que no se trasladaban o no se veían por las dimensiones de la pantalla.
- Detención inesperada en algunos teléfonos inteligentes, por tener pantallas con una densidad y tamaño muy pequeños.
- Los puntajes no aparecían correctamente en la pantalla Puntajes, debido a un elemento de Android, llamado listview que no se visualizaba en algunas ocasiones.
- Corrección de colores para personas que sufren de daltonismo.

### 3.6.4 Anuncio y divulgación de la aplicación

La divulgación de la aplicación se hace a través de las diferentes redes sociales y servicios ya sea facebook, WhatsApp, Instagram y twitter, compartiendo el enlace de la aplicación.

<https://play.google.com/store/apps/details?id=com.estudiantesitm.metrollin>

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. RESULTADOS Y DISCUSIÓN

El resultado final es el sistema de entretenimiento propuesto, alojado en la *Play Store*, en donde puede ser descargado sin costo alguno por cualquier persona con teléfono inteligente o tableta con sistema operativo Android 4.0 (*Ice Cream Sandwich*) o superior.



### INFORMACIÓN ADICIONAL

**Actualizada**  
2 de febrero de 2017

**Requiere Android**  
4.0 y versiones superiores

**Calificación del contenido:**  
Todos  
[Más información](#)

**Permisos**  
[Ver detalles](#)

**Informar**  
[Marcar como inadecuado](#)

**Ofrecida por**  
Juan Fernando Cañola Garcia

**Programador**  
[Visitar el sitio web](#)  
Enviar correo electrónico a [juancanola5942@gmail.com](mailto:juancanola5942@gmail.com)

**Figura 55. Metrollín en la Play Store**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Con la integración de herramientas de software libre para la creación de un entorno gráfico 2D, se implementó un sistema de información interactivo tipo videojuego para usuarios con teléfonos inteligentes y tabletas con sistema operativo Android, llamado Metrollín.

Con los conocimientos adquiridos en la carrera de Ingeniería de Sistemas del ITM y consultas sobre el desarrollo de aplicaciones para Android, se logró alojar en la Play Store la aplicación móvil completamente funcional, adaptada a varias resoluciones de pantalla, con movimientos de personajes autónomos, con gráficos en 2D que puede ser descargada gratuitamente por los usuarios.

Cuando se comenzó a crear la aplicación, no se tenía previsto que los pasajeros se desplazaran y movieran por la plataforma. Gracias a las consultas realizadas y al esfuerzo, se logró que los personajes adquirieran adicionalmente velocidad y animaciones propias, para darle más dinámica al juego.

Gracias a la modificación en las propiedades de velocidad y el color de los pasajeros, se pudo clasificar las 19 estaciones en 7 niveles diferentes, para aumentar gradualmente la dificultad y la dinámica del videojuego. Además, el jugador puede continuar con el entretenimiento del videojuego, una vez finalizada las 19 estaciones por cuanto puede seguir compitiendo en puntaje con otros jugadores y además porque la velocidad de los pasajeros aumenta en cada viaje finalizado por el jugador.

Se logró almacenar los 20 mejores puntajes obtenidos por los jugadores en tiempo real, de forma externa a los dispositivos móviles, utilizando un servidor en la nube, centralizando dicha información, para que los jugadores puedan visualizarlos y conocer su puntaje comparados con el de otros jugadores, lo cual no estaba contemplado inicialmente.

Cuando se comenzó a animar los pasajeros, el sistema presentaba un retardo y el rendimiento no era el esperado, afectando el funcionamiento. Esto pudo ser solucionado, gracias al alojamiento de los *Drawables* en memoria RAM, al momento de cambiar la animación de los personajes, y así se logró mejorar considerablemente el rendimiento de la aplicación.

Gracias al alojamiento de la aplicación en su versión de prueba en la *Play Store*, se logró realizar una buena cantidad de pruebas por parte de algunos usuarios evaluadores, que conllevó al mejoramiento de la aplicación en 8 versiones iniciales, y que gracias a sus aportes, sugerencias y recomendaciones, se fue modificando hasta obtener la versión final con el cual se entrega este proyecto.

Una de las dificultades que fue muy singular en la evaluación de la aplicación, tuvo que ver con 2 evaluadores que sufren de daltonismo. Gracias a ellos, se logró implementar una opción que permite cambiar el color rojo por amarillo, y de esa manera permitir que las

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

personas con esta discapacidad, pudieran jugar a Metrollín. Otro caso particular se dio con una evaluadora que en las características de su dispositivo móvil no permitía la instalación, aun sabiendo que estaba dentro del rango de las características de la aplicación, aportando a la modificación específica en la configuración de la aplicación, con respecto a la densidad de la pantalla, dándose la posibilidad de ampliar el espectro de dispositivos en los que se puede ejecutar la aplicación.

Cuando un jugador ganaba o perdía, solamente aparecían los mensajes de “Muy bien” o “Muy mal”. Esto se cambió por mensajes que motivaran más al jugador a seguir, como: “Puntaje Perfecto”, “Lo lograste”, “Casi, te faltó solo uno” o “No te rindas”.

Como discusión para el juego, se consideró el uso de diferentes perfiles en un mismo dispositivo móvil y de esa manera lograr que varias personas pudieran almacenar su puntaje en un mismo dispositivo. Esto no es posible, ya que el identificador para cada jugador, es el ID de cada dispositivo, el cual es único, y que permite diferenciar si dos personas tienen el mismo nombre en el juego.

Finalmente la pregunta que nos hacen muchos jugadores, es ¿por qué no se puede repetir una estación que ya se superó? El objetivo principal de Metrollín es que los jugadores sumen la mayor cantidad de puntos en cada una de las estaciones y sus diferentes niveles, y así puedan superar el puntaje de los demás jugadores en todo el mundo. Si un jugador pudiera repetir una estación, no tendría sentido quedarse haciendo puntos en una misma estación, cuando el objetivo de Metrollín es sumar puntaje, sacando e ingresando pasajeros en cada una de las estaciones del juego, superando los diferentes niveles y lograr estar entre los mejores puntajes.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

Se logró integrar, desarrollar y dar a conocer a todo el mundo, un sistema de entretenimiento para Android, creado única y exclusivamente con software libre.

El desarrollo de una aplicación de entretenimiento para Android, tiene gran acogida a nivel mundial debido a la cantidad de dispositivos móviles que funcionan con el sistema operativo Android. Su desarrollo puede ser un poco tedioso debido a la cantidad de resoluciones de pantalla a las que una aplicación debe adaptarse, pero vale el esfuerzo por la gran cantidad de dispositivos en que pueda ser instalada la aplicación.

Aunque en el desarrollo de un sistema de entretenimiento, intervienen muchas profesiones como diseñadores gráficos, productores de sonido, programadores, guionistas, entre muchas otras, se puede lograr crear una aplicación completamente funcional, entretenida y con poco presupuesto, gracias a todas las herramientas libres y gratuitas que pueden ser encontradas en internet.

Se puso en práctica la mayoría de conocimientos adquiridos en la carrera de Ingeniería de Sistemas como lógica de programación, matemáticas para la informática, lenguaje de programación java, base de datos, arquitectura de software, desarrollo de software, programación distribuida, ingeniería del software, introducción al software libre, proyecto integrador en software libre, sistemas operativos, entre otros conocimientos necesarios para el desarrollo de esta aplicación interactiva de entretenimiento.

Como recomendaciones para el desarrollo de aplicaciones móviles podemos decir que:

- Si se quiere desarrollar un sistema de entretenimiento parecido a Metrollín, con gran cantidad de recursos y movimiento, se debe crear sobre un objeto *View* y utilizar las librerías como *Drawable* y *Canvas*, para dibujar sobre este objeto. No cometer el error de crear cada personaje u objeto heredado de la clase *View*, por el contrario, todo hace parte de una misma *View*.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Gimp es el software ideal para la creación y edición de imágenes y animaciones. Posee todas las herramientas como crear, recortar, filtros, cambiar color, tono de luz, e insertar texto, entre muchas otras, para la creación de recursos.
- Linux Multimedia Studio es el software ideal, para la creación de música y efectos de sonido. Contiene una gran cantidad de sonidos de muestra, instrumentos y herramientas para crear tu propia música y efectos.
- El software libre requiere más tiempo y consulta para su instalación y uso, pero no se debe temer a este proceso. Una vez dominado un software, se aprende a sacar provecho de todas sus funcionalidades y resulta más sencillo cada vez que se usa.
- Se deben realizar los prototipos que sean necesarios para desarrollar lo que se desee. En nuestro caso al comenzar el desarrollo de la aplicación Android, no tuvimos en cuenta el movimiento de los personajes. Lo que sucedió fue que tuvimos que volver a crear la aplicación desde cero, debido que los personajes los creamos sobre un Layout y no sobre un objeto View como bien lo recomendaba el gran libro de Android (Android).
- Para la optimización de recursos como imágenes y sonido, siempre se deben mantener cargados en memoria RAM asignados en variables. Tuvimos la experiencia de presentar retardos en la actualización del movimiento de personajes, y los gráficos se veían lentos, lo cual no era muy amigable para el jugador. Esto fue debido a que siempre que queríamos animar los personajes, codificábamos el juego de manera que siempre buscara la imagen en la memoria interna del dispositivo móvil y no en la memoria RAM que es donde el procesador tiene acceso más rápido, con lo cual fue solucionado.
- Siempre buscar ayuda en la web, para eso está la comunidad de software libre. Para la aplicación requeríamos la utilización de gif animados y Android no posee esta característica por defecto. Pero consultando en la web, descubrimos una librería que permite realizar animaciones para archivos en formato .gif, lo cual facilitó mucho el desarrollo de la aplicación.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- El proceso para el desarrollo de un sistema de entretenimiento como este, requiere mucho tiempo, mucha paciencia y mucha dedicación, por lo que siempre debes estar dispuesto a nunca rendirse.

Como futuro trabajo, se puede hacer la aplicación en 3D, con más personajes y que pueda ser la aplicación oficial de un sistema de transporte masivo tipo metro y las variantes que pueda tener, como por ejemplo el Metro de la ciudad de Medellín.

También que exista una versión de pago de Metrollín, para las personas que deseen tener una mayor experiencia en el juego y obviamente que se pueda obtener ganancias a partir de esta aplicación.

Falta desarrollar la aplicación Metrollín para otros sistemas operativos móviles como iOS y Windows Phone, ya que no todas las personas poseen Android.

Y también implementar las diferentes redes sociales como facebook, twitter, Instagram, entre otras, en la aplicación, para que las personas puedan compartir sus puntajes, sus logros y mostrar sus experiencias con el juego a otras personas.

Se desea aportar a la ciudad que se considera más innovadora; Medellín, y mostrarle a Colombia y el mundo, un producto de buena calidad, realizado por dos estudiantes de Ingeniería de Sistemas del ITM, como aporte a dicha pretensión.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# REFERENCIAS

---

## Bibliografía

- Android. (s.f.). *Android Developers*. Recuperado el 20 de Abril de 2016, de <http://developer.android.com/intl/es/training>
- Arch Linux. (s.f.). *Arch Linux*. Recuperado el 30 de Enero de 2016, de <https://www.archlinux.org/>
- Audacity. (s.f.). *Audacity*. Recuperado el 30 de Enero de 2016, de <http://audacity.es/>
- Eco City Games. (22 de Junio de 2015). *Play Store*. Recuperado el 20 de Abril de 2016, de <https://play.google.com/store/apps/details?id=com.danielviktorin.tramsimulador>
- Gimp. (s.f.). *GNU Image Manipulation Program*. Recuperado el 30 de Enero de 2016, de <https://www.gimp.org>
- Girónes, J. T. (2012). *El gran libro de Android*. Barcelona, España: Marcombo S.A.
- Google. (s.f.). *Google chrome*. Recuperado el 30 de Enero de 2016, de <https://www.google.com/chrome/browser/desktop/index.html>
- LMMS. (s.f.). *Linux MultiMedia Studio*. Recuperado el 30 de Enero de 2016, de <https://lmms.io/>
- Maximiliano, J. (31 de Mayo de 2011). *Mi granito de Java*. Recuperado el 10 de Octubre de 2014, de <http://migranitodejava.blogspot.com/2011/05/factory-method.html>
- NotePadQQ. (s.f.). *Notepadqq*. Recuperado el 30 de Enero de 2016, de <http://notepadqq.altervista.org/wp/>
- SG Software. (22 de Octubre de 2014). *Play Store*. Recuperado el 19 de Abril de 2016, de <https://play.google.com/store/apps/details?id=com.sgsoftware.sgrailroad2d>
- The Document Foundation. (s.f.). *Libre Office*. Recuperado el 30 de Enero de 2016, de <https://es.libreoffice.org/>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# APÉNDICE

## Código del método fábrica aplicación Metrollín

### Interface PersonaFactoryMethod

```

package com.estudiantesitm.metrollin.juego.factory;
import android.view.View;
/**
 * Created by Juan on 12/10/2016.
 */
public interface PersonaFactoryMethod {
    Persona crearPersona(View view);
}

```

### Clase PersonaFactoryMethod

```

package com.estudiantesitm.metrollin.juego.factory;
import android.view.View;
import com.estudiantesitm.metrollin.util.Utilidad;
/**
 * Created by Juan on 12/10/2016.
 */
public class PersonaFactory implements PersonaFactoryMethod {
    int numAleatorio;
    @Override
    public Persona crearPersona( View view) {
        numAleatorio = Utilidad.numeroAleatorioEntre(1, 3);
        switch (numAleatorio) {
            case 1:
                return new Man(view);
            case 2:
                return new Woman(view);
            case 3:
                return new Child(view);
            default:
                return new Child(view);
        }
    }
}

```

### Clase persona

```

package com.estudiantesitm.metrollin.juego.factory;
import android.content.SharedPreferences;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.drawable.Drawable;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.View;
import com.estudiantesitm.metrollin.R;
import com.estudiantesitm.metrollin.util.Utilidad;
/**
 * Created by Juan on 12/10/2016.
 */
public abstract class Persona {
    public Drawable

```

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

drawable, drawableQuieto, drawableSeleccion, drawableMovimiento1, drawableMovimiento2
;
private double posX, posY; //Imagen de la persona
private double incX, incY; //Posicion
private double incX, incY; //Velocidad desplazamiento
private int ancho, alto; //Dimensiones de la imagen
private int radioColision; //Para determinar colision
private String personaje; //Para saber el tipo de personaje
private int colorPersona; //Color para el personaje
private int velocidadJuego; //Conocer la velocidad del juego
private boolean habilitado; //Habilitar personajes
private int flag; // Bandera para la animacion
//Preferencias
SharedPreferences preferences;
//Donde dibujaremos el grafico (usaba en view.invalidate)
private View view;
public Persona(View view){
    this.view = view;
    drawable = Utilidad.convertirABitmap(view, R.drawable.personajewoman);
    // Solamente para coger las medidas de la imagen
    ancho = drawable.getIntrinsicWidth(); //Retorna el ancho de la imagen
    alto = drawable.getIntrinsicHeight(); //Retorna el alto de la image
    radioColision = (alto+ancho)/6;
    //Leer la cantidad y aumenta la cantidad de viajes
    preferences =
PreferenceManager.getDefaultSharedPreferences(view.getContext());
    velocidadJuego = Integer.parseInt(preferences.getString("Viajes", "1"));
    //Posicion persona
    double x, y;
    x = Utilidad.numeroAleatorioEntre(100, (view.getWidth() - getAncho()));
    y = Utilidad.numeroAleatorioEntre (100, view.getHeight()/2 - getAlto());
    setPosX((int) x);
    setPosY((int) y);
    //Velocidad personas Aleatorio entre 0 y 10
    setIncY(Utilidad.numeroAleatorioEntre(0, 10)); //Movimiento persona
    setIncX(Utilidad.numeroAleatorioEntre(0, 10));
    //Color
    colorearPersona();
    //habilitamos
    habilitado = true;
}

public void dibujaGrafico(Canvas canvas){
    canvas.save();
    drawable.setBounds((int)posX, (int)posY, (int)posX+ancho, (int)posY+alto);
    //dibujar rectangulo , coordenadas (int left, int top, int right, int bottom)
    drawable.draw(canvas);
    canvas.restore();
}

public void incrementaPos (double factor){
    posX+=incX*factor* velocidadJuego;
    if (posX<-ancho/8)
    {posX = -ancho/8; incX=-incX;}
    if (posX>view.getWidth()-ancho)
    {posX=view.getWidth()-ancho; incX=-incX;}
    posY+=incY*factor* velocidadJuego;
    if (posY<-alto/8) {posY = -alto/8; incY=-incY;}
    if (posY>view.getHeight()/2-100) {posY=view.getHeight()/2-100; incY=-incY;}
}

```

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

public void animar() {
    if(habilitado) {
        flag = Utilidad.numeroAleatorioEntre(0, 1);
        if (flag == 0) {
            drawable = drawableMovimiento1;
            flag = 1;
        } else {
            drawable = drawableMovimiento2;
            flag = 0;
        }
        drawable.setColorFilter(cambiarColor(), PorterDuff.Mode.SRC_ATOP);
    }
}

public void deshabilitar() {
    drawable=drawableQuieto;
    drawable.setColorFilter(Color.parseColor("#10FFFFFF"),
PorterDuff.Mode.SRC_ATOP);
}
public void dejarQuieto() {
    drawable=drawableQuieto;
    drawable.setColorFilter(cambiarColor(), PorterDuff.Mode.SRC_ATOP);
}
public void seleccionar() {
    drawable=drawableSeleccion;
    drawable.setColorFilter(cambiarColor(), PorterDuff.Mode.SRC_ATOP);
}
public int colorearPersona(){
    int numAleatorio = Utilidad.numeroAleatorioEntre(1, 3);
    int colorPersona;
    switch (numAleatorio) {
        case 1:
            colorPersona = Color.parseColor("#8000FF00"); // #AARRGGBB
            setColorPersona(0x00FF00);
            break;
        case 2:
            colorPersona = Color.parseColor("#80FF0000"); // #AARRGGBB
            setColorPersona(0xFF0000);
            break;
        case 3:
            colorPersona = Color.parseColor("#800000FF"); // #AARRGGBB
            setColorPersona(0x0000FF);
            break;
        default:
            colorPersona = Color.parseColor("#80FFFFFF"); // #AARRGGBB
            setColorPersona(0xFFFFFFFF);
            break;
    }
    return colorPersona;
}
protected int cambiarColor(){
    int color=0;
    switch (colorPersona) {
        case 0x00FF00:
            color = Color.parseColor("#8000FF00"); // #AARRGGBB
            break;
        case 0xFF0000:
            color = Color.parseColor("#80FF0000"); // #AARRGGBB
            break;
        case 0x0000FF:
            color = Color.parseColor("#800000FF"); // #AARRGGBB
            break;
        case 0xFFFFFFFF:

```



 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### Clase Man

```

package com.estudiantesitm.metrollin.juego.factory;
import android.view.View;
import com.estudiantesitm.metrollin.R;
import com.estudiantesitm.metrollin.util.Utilidad;
/**
 * Created by Juan on 12/10/2016.
 */
public class Man extends Persona {
    public Man(View view) {
        super(view);
        drawableQuieto = Utilidad.convertirABitmap(view,
R.drawable.personajeman);
        drawableSeleccion= Utilidad.convertirABitmap(view,
R.drawable.personajemanselected);
        drawableMovimientol = Utilidad.convertirABitmap(view,
R.drawable.personajeman1);
        drawableMovimiento2 = Utilidad.convertirABitmap(view,
R.drawable.personajeman2);
        drawable=drawableQuieto;
        setPersonaje("man");
    }
}

```

### Clase Child

```

package com.estudiantesitm.metrollin.juego.factory;
import android.view.View;
import com.estudiantesitm.metrollin.R;
import com.estudiantesitm.metrollin.util.Utilidad;
/**
 * Created by Juan on 12/10/2016.
 */
public class Child extends Persona {
    public Child(View view) {
        super(view);
        drawableQuieto = Utilidad.convertirABitmap(view,
R.drawable.personajechild1);
        drawableSeleccion= Utilidad.convertirABitmap(view,
R.drawable.personajechildselected);
        drawableMovimientol = Utilidad.convertirABitmap(view,
R.drawable.personajechild1);
        drawableMovimiento2 = Utilidad.convertirABitmap(view,
R.drawable.personajechild2);
        drawable=drawableQuieto;
        setPersonaje("child");
    }
}

```

### Clase Woman

```

package com.estudiantesitm.metrollin.juego.factory;
import android.view.View;
import com.estudiantesitm.metrollin.R;
import com.estudiantesitm.metrollin.util.Utilidad;
/**
 * Created by Juan on 12/10/2016.
 */
public class Woman extends Persona {
    public Woman(View view) {
        super(view);
        drawableQuieto = Utilidad.convertirABitmap(view,

```

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

R.drawable.personajewoman);
    drawableSeleccion= Utilidad.convertirABitmap(view,
R.drawable.personajewomaselected);
    drawableMovimiento1 = Utilidad.convertirABitmap(view,
R.drawable.personajewoman1);
    drawableMovimiento2 = Utilidad.convertirABitmap(view,
R.drawable.personajewoman2);
    drawable=drawableQuieto;
    setPersonaje("woman");
    }
}

```

### Para crear una persona

```

PersonaFactoryMethod factory;
factory = new PersonaFactory();
Persona persona = factory.crearPersona(this);

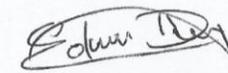
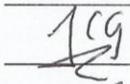
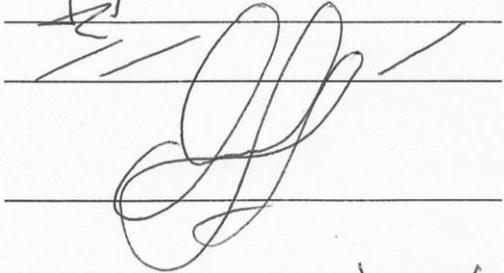
```

El método retorna un objeto persona sea child, man o woman, dejando a la clase Fábrica encargada de la creación de personajes, sin tener que especificar color, velocidad, personaje, posición y demás atributos.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Proyecto  
 Metrolin  
 ya entregado

FIRMA ESTUDIANTES

FIRMA ASESOR

FECHA ENTREGA: Feb 13/17

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO \_\_\_\_\_      ACEPTADO \_\_\_\_\_      ACEPTADO CON MODIFICACIONES \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_