

**Universidad Nacional Autónoma de Nicaragua  
RURD-Managua  
Facultad de ciencias e ingeniería  
Departamento de Computación**



**Seminario de Graduación**

**Tema:**

Programación de Videojuegos.

**Sub-Tema:**

Programación del videojuego Tours Nica, en ambiente gráfico para tablet, bajo plataforma android en el primer semestre del año 2013.

**Autores:**

Brch. Abraham Aguirre González  
Brch. Jimmy Díaz Parrales

**Tutor:**

Msc. Juan de Dios Bonilla Anduray



## **DEDICATORIA**

Este proyecto de seminario de graduación se lo dedicamos primeramente a Dios por habernos prestado el tiempo, la vida y la sabiduría.

A nuestras familias, que han sido el motor, que nos han impulsado seguir adelante, para cumplir nuestros sueños, metas e intereses.

A nuestro tutor Msc. Juan de Dios Bonilla Anduray, que nos ayudó y motivo a realizar este proyecto.

A Nuestro profesores que con dedicación, compartieron sus conocimientos a lo largo de nuestra carrera, con el único propósito de que llegásemos hacer excelente profesionales.

Bach: Abraham Aguirre González

Bach: Jimmy Díaz Parrales

## DEDICATORIA

*Este proyecto está dedicado primeramente a Dios que me ha dado la sabiduría, el tiempo y la vida, y a mi familia, quienes han estado física y espiritualmente apoyándome constantemente.*

*A mi Madre Juana Mercedes (qed) y mi Padre Juan Aguirre, quienes no escatimaron recurso alguno, para darme una mejor educación.*

*A mis hermanos, por el apoyo brindado en todo el trayecto de mi carrera, siendo un ejemplo a seguir para ellos.*

*A mi dos hijos, Elías Josué (qed) y Ricardo Abraham, quienes me han enseñado hacer un mejor Padre y personas, ya que son el principal motivo que me impulsa cada día a superarme académicamente.*

*Y finalmente a mi esposa Reyna Martínez, considerada mi compañera de estudio por estar siempre a mi lado, apoyándome contantemente y a la vez pidiéndole disculpa por el tiempo robado que le pertenecía a ella, pero lo dedique a este proyecto.*

*Brch: Abraham Aguirre.*

## DEDICATORIA

*Este seminario se lo dedico a Dios que ha estado siempre a mi lado, protegiéndome del mal, llenándome de fuerza y perseverancia antes mis fracasos, por saber de mis sueños y ser parte de él.*

*A mi familia por robarles tiempo y ser lo que soy.*

*A mis padre por apoyarme siempre con sus consejos, por indicarme siempre el camino correcto y por haberme enviado a la escuela e inducirme en este camino.*

*A mi tía abuela Felicita Parrales (qed) que siempre me deseo lo mejor y siempre cuando pudo me apoyó.*

*A Maritza Mendieta y Ena parrales que siempre tuvieron al pendiente*

*Al hogar de Fermín Bermúdez Baltodano y Gloria Elena García Parrales, por haberme acogido en su hogar y tratarme como un hijo más, por lo que siempre estaré eternamente agradecido y guardo mucho aprecio.*

*Finalmente, a los dos regalos más bellos que Dios me ha dado, mis hijos Jimmy Elliam Díaz Maradiaga y Kendra Jimena Díaz Chacon.*

*Brch: Jimmy Díaz Parrales*

## **AGRADECIMIENTO**

Agradecemos primeramente a nuestro Dios por brindarnos la vida, el tiempo y la sabiduría, para la realización de este proyecto de seminario de graduación.

A nuestras familias, quienes no dudaron de su apoyo incondicional, para sentirnos capaces, de asumir nuevos retos, en nuestra formación profesional.

A nuestros profesores, ya que sin sus conocimientos, experiencia y enseñanzas, no fuera posible nuestro proyecto.

Brch: Abraham Aguirre González

Brch: Jimmy Díaz Parrales

## **RESUMEN DEL PROYECTO**

El desarrollo del videojuego asido todo un reto, debido que es un tema nuevo poco explorado. Por lo que hemos dedicado mucho tiempo de investigación para tener un mayor alcance en el desarrollo de nuestro proyecto.

Se ha elaborado un guión, en el que se detalla la historia y en base a esta, desarrollar la programación. También se ha elaborado un documento de diseño, en donde se detalla la mecánica de nuestros personajes, escenarios y entornos de desarrollo, con el fin de que el usuario o jugador, tenga una mayor comprensión.

Como objetivo principal, presentar una producción propia ajustada a nuestra realidad, simulando el recorrido del protagonista, partiendo desde Metro centro (carretera Masaya) hacia a Granada.

Para el desarrollo utilizamos las siguientes herramientas:

Eclipse como ide, el ADT (plugging oficial), kit de desarrollo para android, JDK java kit de desarrollo de java, Libgdx como framework para el desarrollo de nuestra aplicación, photoshop CS5 para la edición de cada una de las imágenes y pinnacle studio vrs 12 para edición del sonido y música y java como lenguaje de programación.

En el enfoque orientado a objeto se desarrollaron diagramas caso de uso y de clases.

CONTENIDO

INTRODUCCIÓN.....	11
JUSTIFICACIÓN.....	12
OBJETIVO GENERAL.....	13
OBJETIVO ESPECÍFICO.....	13
MARCO CONCEPTUAL.....	14
I.PROGRAMACIÓN DE VIDEOJUEGOS.....	14
1.1 Breve historia de los videojuegos.....	14
1.2 ¿Qué es un videojuego?.....	21
1.3 Anatomía de un video Juego.....	21
1.3.1 Entrada.....	21
1.3.2 Visualización.....	21
1.3.3 Sonido.....	21
1.3.4 Comunicaciones.....	21
1.4 Programación móvil.....	22
1.4.1 Sistemas Operativos Móviles.....	22
1.4.1.1 Symbian.....	22
1.4.1.2 Windows Mobile.....	22
1.4.1.3 iPhones OS.....	23
1.4.1.4 Android.....	23
2- AMBIENTE DE DESARROLLO DE LOS VIDEOJUEGOS.....	24
3- ANDROID.....	25
3.1 Historia y Éxito de Android.....	25
3.2- Estructura de Android.....	28
3.3-Estructura de las Aplicaciones de Android.....	29
3.4 Estructura de un Proyecto Android.....	30
3.5 Entorno de desarrollo de android.....	31
4. ECLIPSE Y ANDROID.....	32
4.1 Eclipse.....	32
4.2 Historia de Eclipse.....	32
4.3 Instalación de Android SDK y Eclipse.....	33
4.4 Plugin ADT para Eclipse.....	33
4.5 Creación de una Máquina Virtual.....	34
5 LENGUAJE DE PROGRAMACIÓN JAVA.....	34
5.1 Historia.....	34
5.2 ¿Porque Utilizar Java Como Lenguaje de Programación en Android?.....	35
5.2.1 Características y Propiedades de Java.....	35
6. MOTOR LIBGDX PARA ECLIPSE Y ANDROID.....	36
6.1 ¿Que es Libgdx?.....	36
6.1.1 Ventajas:.....	37
6.1.2 Desventaja:.....	37
6.2 Ciclo de Vida De una Aplicación LIBGDX.....	37
6.2.1 La aplicación.....	37

DISEÑO METODOLÓGICO .....	38
Tipo De estudio .....	38
Descripción del Juego.....	38
Procedimiento de desarrollo del videojuego .....	39
Estudio de Factibilidad .....	42
Factibilidad Técnica.....	42
Factibilidad Económica.....	43
Factibilidad operacional .....	44
Cronograma de Actividades del Proyecto Tours Nica. ....	45
Entorno de trabajo haciendo uso de eclipse y el plugin oficial de android el ADT .....	50
DESARROLLO.....	53
Elaborar el guión del videojuego “Tours Nica”. ....	53
Documento de Diseño .....	54
Caso de uso, diagrama de clase .....	60
Digrama de Secuencia de Iniciar Juego .....	62
Diagrama de clase .....	63
RECOMENDACIONES .....	72
BLIOGRAFÍA .....	73



## INDICE DE FIGURAS

<b>Descripción</b>	<b>Número de página</b>
Figura #1 Ipad Touch	14
Figura #2 TouchSmart	15
Figura #3 Netbook tablet	15
Figura #4 xTablet T700	15
Figura #5 Ipad	16
Figura #6 Galaxy S	16
Figura #7 Motorola Xoom	17
Figura #8 Ipad 2	17
Figura #9 Galaxy S2	18
Figura #10 Toshiba AT300	18
Figura #11 Iphone	18
Figura #12 Surface	19
Figura #13 Nexus 7	19
Figura #14 Estructura de Android	28
Figura #15 Estructura Proyecto Eclipse	30
Figura #16 Carpeta SRC eclipse	31
Figura #17 Estructura Libgdx	36
Figura #18 Proceso del desarrollo del videojuego	40
Figura #19 Cotización Tablet	43
Figura #20 Ambiente de Programación Eclipse 1	51
Figura #21 Ambiente de Programación Eclipse 2	51
Figura #22 Ambiente de Programación Eclipse 3	53
Figura #23 Texture Atlas	55
Figura #24 Carro Protagonista	55
Figura #25 Carro Enemigo	56
Figura #26 Obstáculos	56
Figura #27 Baúl de Vidas	56
Figura #28 Pantalla Principal	57
Figura #29 Pantalla Ayuda	57
Figura #30 Escenario del Videojuego	58
Figura #31 Pantalla Game Over	58
Figura #32 Pantalla de fin de Partida	59
Figura #33 Caso de Uso	61
Figura #34 Diagrama de Secuencia	63
Figura #35 Diagrama de Clase	64
Figura #36 Clase Assets	65
Figura #37 Clase de Mygame	66
Figura #38 Clase menú principal	66
Figura #39 Clase Gamescreen	67
Figura #40 Clase Gameoverscreen	67
Figura #41 Clase InfiniteScrollBGActor	68
Figura #42 Clase CarroProtagonista	68

**INDICE DE FIGURAS**

Figura #43 Clase animalesActor	69
Figura #44 Clase VidaActor	69
Figura #45 Clase Puntuación Actor	70
Figura #46 Clase AgruparActoresEscenariogame	71

**INDICE DE TABLA**

<b>Descripción</b>	<b>Número de página</b>
Tabla #1 Factibilidad Técnica	41
Tabla #2 Factibilidad Recursos Humanos	43
Tabla #3 Cronograma de trabajo 1	44
Tabla #4 Cronograma de trabajo 2	45
Tabla #5 Cronograma de trabajo 3	46
Tabla #6 Cronograma de trabajo 4	47
Tabla #7 Cronograma de trabajo 5	48
Tabla #8	49

## INTRODUCCIÓN

Actualmente los teléfonos móviles y las Tablet son ya parte esencial en nuestra forma de vida. Cada día son más los usuarios que están conectados a estos dispositivos.

La evolución de estos dispositivos ha hecho que la gran mayoría de las personas que se dedican al desarrollo de sistemas para PC, migren hacia esta nueva plataforma, con el único objetivo de satisfacer las demandas de los portadores de estos. Por lo que están en la mira de ser los próximos ordenadores que sustituirán a las PC, debido a su comodidad de operación y portabilidad.

La gran demanda de estos dispositivos han causado un desbordamiento de intereses económicos por parte de sus creadores, siendo así que se han venido desarrollando sistemas operativos más amigables, seguros y de mejor rendimiento, tal es el caso del sistema operativo Android, siendo el más popular hoy en día por ser un software de comercialización gratuita, y brinda a los desarrolladores un sinnúmero de herramientas para desarrollar aplicaciones, siendo popularmente el desarrollo para videojuegos.

Los videojuegos hoy en día, van ganando mayor popularidad en todo el mundo, la mayoría de las personas dedicadas a esta comercialización, las comparan con la industria del cine, debido a que muchos de estos, que se crearon para consolas como Atari, Nintendo64, Playstation, Nintendo wi, han venido hacer reemplazados o ya sea codificados nuevamente para plataformas de dispositivos móviles.

Cabe señalar que el desarrollo de videojuegos en Nuestro país, aún no asido experimentado, ya que no existe empresas, instituciones, personas que se dediquen a la comercialización y producción de estos.

Es por tales motivos, que nos hemos aventurado en explorar y desarrollar un videojuego, indagándonos desde su creación hasta su evolución. Para llevar a cabo este proyecto (video juego 2d), nos hemos visto en la necesidad de investigar el entorno de desarrollo, el cual consiste en las siguientes herramientas de trabajo:

1. ADT(android development tools) es el kit de desarrollo de android que trae por defecto los siguientes paquetes:
  1. El SDK: las distinta librerías necesarias para que nuestra aplicación sea compatibles con todas las versiones de android disponible.
  2. Eclipse: el ide ó interfaz de desarrollo.
  3. El JRE: la versión del lenguaje java .
  4. JDK de java: al igual que el ADT es el kit desarrollo de java , el cual contienen el JRE y java virtual Machine, encargados de compilar y ejecutar nuestra aplicación.

## **JUSTIFICACIÓN**

En Nicaragua actualmente se desconoce a empresas o Personas que se dediquen, a la industria de desarrollo de video juegos, desaprovechando de esta manera, los recursos tecnológicos que se han venido dando. Siempre hemos estado consumiendo aplicaciones de video juegos extranjeros, que no se ajustan a la realidad ni a la cultura de nuestro país.

Hemos sido testigos que los videojuegos de PC, Internet, redes sociales o móviles ya sean en ambientes de consolas o gráficos (2d, 3d), han tomado posiciones muy importantes en el mundo de la tecnología.

De tal manera que las empresas desarrolladores de videojuegos, los comparan con la industria del cine, debido a la gran cantidad de divisas que esta industria genera.

Debido ala necesidad, de tener una producción propia, que nos promueva como país, hemos decidido aplicar nuestro conocimiento, para el desarrollo de un videojuego, en el cual podamos mostrar parte del territorio y de la cultura, que llene las expectativas de las personas o jugadores, basándonos en el lema “PRODUCIENDO Y CONSUMIENDO LO NUESTRO, IREMOS AVANZANDO CADA DÍA EN EL DESARROLLO DE NUESTRO PAIS”.

Para este proyecto hemos propuesto desarrollar un video juego que permita al jugador una interacción directa, mediante toques táctiles para alcanzar una meta, donde se divierta y a la vez conozca más de nuestro País.

Hemos escogido una Tablet como dispositivo para su implementación, debido a la portabilidad y comodidad del usuario, y por su gran demanda en la actualidad.

### **OBJETIVO GENERAL**

Desarrollar un video juego con el nombre TOURS NICA en ambiente gráfico para Tablet bajo plataforma android en el primer semestre del 2013.

### **OBJETIVO ESPECÍFICO**

1. Elaborar el guión del videojuego “Tours Nica”.
2. Crear el documento de diseño asociado al videojuego “Tours Nica”.
3. Utilizar el paradigma Orientado a Objeto en la programación del juego “Tours Nica”.

## MARCO CONCEPTUAL

### PROGRAMACIÓN DE VIDEOJUEGOS

#### 1.1 Breve historia de los videojuegos

El primer país que comenzó en el desarrollo de videojuegos fue Estados Unidos con una empresa conocida popularmente llamada **SEGA** (Service Games) era una empresa Americana que en los años 40 se dedicaba a la comercialización de primitivas máquinas de monedas basadas en sistemas mecánicos y que en la década de los 50 fue trasladada a Japón. En 1968 alcanzaron su primer éxito con la recreativa Periscopio, un simulador de submarinos, crearían juegos clásicos como Out Run, Shinobi, la saga de Sonic, Daytona, etc. Seguidamente fue **ATARIS** también de Origen Americano creado por **Nolan Bushnel** en 1972, con su popular juego PONG que fue una copia de **tennis for Two** de su autor original **BILL NIGHINBOTTHAM**. Luego **APPLE** de origen americano también que en 1976 junto con Atari contaba entre sus filas con un tal Steve Jobs y un tal Steve Wozniak que desarrollaron el juego Breakout, primer juego machaca-bloques de la historia, que era una nueva variación de Pong. Ambos, usando componentes de Atari crearon el primer computador Apple, esto le dio la pauta para crear su empresa y dedicarse hoy en día al mundo de las computadoras, iPod, Tablet, etc. y por consiguiente la empresa japonesa **Nintendo** fue fundada en 1889 por Fusajiro Yamauchi y hasta los años 70 del siglo XX se dedicó, sobre todo, a la fabricación de naipes para juegos. **Nintendo** no perdió de vista la aparición de tecnologías electrónicas para el ocio y, ya a principios de los años 70 presentó algún dispositivo de juegos para salas recreativas, basados en principio en la reproducción de video. En 1977 presenta su consola doméstica COLOR TV GAME 6, en los 80 sacó la SNES y en 1996 la Nintendo64.[3]

El mundo de los videojuegos para Tablet comienza con la popular empresa llamada Apple que es el año 2008 donde, Lanza el Ipod touch, un reproductor mp3 con Wi-Fi, con características de asistente digital y consola de juegos. Fue el primer Ipod con capacidad de conectarse a la tienda virtual iTunesStore. Corria con iOS.[16]



Figura #1 Ipod touch.

Luego en ese mismo año la 2008 empresa HP presento, la HP TouchSmart tx12 series, estas fueron una serie dispositivos que podían ser usados como laptop, PC o tablet, funcionaba con distintas versiones de procesadores Intel o AMD y corría Windows Vista o 7. Poseía una pantalla de 19 pulgadas, memoria RAM de 2 Gb, procesador AMD Turion 64 X2. [16]



Figura #2 TouchSmart

Posterior en el 2009 la empresa Asus presenta la netbook tablet, la EEE PC T91, era una combinación de Linux OS con peso ligero, disco duro de estado sólido (SSD), relativamente bajo costo, luego aparecieron variaciones de este dispositivo con Windows y con disco duro (HDD). Con un costo de 500 euros con un procesador Celeron, Atom o AMD. [16]



Figura #3 Netbook tablet

Y el 2010, El año de la tablet se convierte en un ícono comercial. **Mobile Demand** Lanza al mercado la xTablet T7000 Rugged Tablet PC que corría con Windows e integraba un teclado numérico. Además incluía un scanner de código de barras, teclado QWERTY, lector de tarjetas de crédito, cobro de pagos en el sitio en el que se encuentre el dispositivo, además de GPS opcional con posibilidad de establecer una ruta apropiada.



Figura #4 xTablet T700

Seguidamente Apple lanza su Apple iPad es el tablet que Apple comercializa en la actualidad y que según Jobs, es el fruto de varios años de trabajo. El iPad es un dispositivo electrónico tipo tableta desarrollado por Apple Inc. La primera generación es anunciada el día 27 de enero de 2010.



Figura #5 Ipad

Mas tarde aparece la compañía Samsung con el Galaxy Tab, una Tablet basada en Android fabricado por Samsung que debutó el 2 de septiembre de 2010 en la IFA de Berlín.

Este tablet fue el primero en plantar cara y mejorar el ipad de Apple; a diferencia de éste, Samsung opto por mejorar ligeramente las características del Galaxy S para luego implantarlas en un dispositivo de 7", del potente procesador de 1Ghz del Galaxy S, El sistema operativo original de este dispositivo es Android 3.0 actualizado de forma automática a 3.2.0.



Figura #6 Galaxy S



La empresa Motorola de buta con su Tablet Motorola Xoom es una tableta de Motorola basada en Android 3.0 Honeycomb, presentada en el CES 2011, el 5 de enero de 2011. Es la primera tableta que lleva instalado Android 3.0. Fue lanzada el 24 de febrero de 2011.

Dispone una variedad de sensores, incluido un sensor de luz ambiental para auto ajustar el brillo de la pantalla, un acelerómetro y un giroscopio de 3 ejes, un magnetómetro (brújula), y un barómetro. Utiliza el chip Nvidia Tegra 2 T20, con un CPU doble núcleo a una frecuencia de 1 GHz. [16]



Figura #7 Motorola Xoom

Mientras que el 2 de marzo de 2011 (última presentación de Steve Jobs) apareció la segunda generación. Se sitúa en una categoría entre un "teléfono inteligente" (smartphone) y una computadora portátil, enfocado más al acceso que a la creación de aplicaciones y temas.



Figura #8 Ipad 2

Samsung introdujo el 8 de junio de 2011 un modelo de mayor tamaño al original, y con mejores características técnicas. Cuenta con pantalla multitáctil LCD IPS de 10.1 pulgadas, resolución de pantalla WXGA de 1280×800 píxeles y densidad de 149 ppi (píxeles por pulgada), capacidad Wi-Fi, procesador NVIDIA Tegra 2 dual-core de 1 GHz. Está equipado con una cámara trasera de 3.1 Megapíxeles con flash de LED y una frontal para videollamadas de 2.0 Megapíxeles. Memoria RAM 1 GB DDR2, memoria flash 16, 32, o 64 GB. 3G. aGPS. Bluetooth. Está equipado con Android 3.1 (Honeycomb) actualizable a Android 3.2 con TouchWiz UX. [16]



Figura #9 Galaxy S2

Se anuncia la Toshiba Tablet con una pantalla de 10 pulgadas y Android 3.0.

El nuevo modelo responde al nombre de Toshiba AT300, y se trata de un modelo de 10.1 pulgadas (1280×800 píxeles) en el que destacan un par de aspectos: Nvidia Tegra 3 en la parte hardware, y Ice Cream Sandwich como versión del sistema operativo.



Figura #10 Toshiba AT300

La tercera versión del dispositivo que pasó a llamarse de nuevo iPad,2 fue presentada el 7 de marzo de 2012. Esta fue la primera presentación de productos del CEO Tim Cook, tras el fallecimiento de Jobs.

Las funciones son similares al resto de dispositivos portátiles de Apple, como es el caso del iPhone o iPod touch, aunque la pantalla es más grande y su hardware más potente. [16]



Figura #11 Iphone

Se espera la llegada de la Surface, en Octubre de 2012, la tablet con la que Microsoft espera hacer frente a los dos principales fabricantes de tablets, Apple y Samsung.

Este agradable tablet de 10.6 pulgadas vendrá en dos versiones: una con Windows 8 Pro para los que buscan productividad (además de un procesador intel) y otra con Windows 8 RT, más pequeña, liviana y con procesador ARM que creemos le puede hacer competencia a iPad. [16]



Figura #12 Surface

Nexus 7, el dispositivo con el que Google quiere competir en el mercado de las tabletas, inició su comercialización en España, Francia y Alemania.

La tablet pesa 340 gramos, tiene una pantalla de alta resolución de 7 pulgadas, un procesador de cuatro núcleos y cámara frontal de 1.2 megapíxeles. Además, está equipada con la última versión del sistema operativo móvil Android, la 4.1 o Jelly Bean.



Figura #13 Nexus 7

La tablet como cualquier dispositivo a la vanguardia ha sido limitado por la tecnología en la que se desarrolló, no obstante vemos que se ha desarrollado desde hace décadas, siempre ha tenido la visión de un dispositivo que permita llevar a cabo tareas básicas y necesarias, donde un computador no sea necesario, puesto que este último es muy pesado y nada fácil en cuanto a portabilidad. La tablet nace de la necesidad de un equipo que sea versátil y funcional a la vez, donde vaya a la par de las necesidades de la sociedad.

A pesar de todo los obstáculos económicos para que este equipo en particular sea de difusión mundial hace que no llegue a todos los niveles sociales pues un Ipad posee un costo elevado, por eso otras compañías como Google apuntan a mercados de costos bajos, específicamente espera llegar allí con su nuevo dispositivo, el Nexus, el cual busca minimizar las brechas económicas en cuanto a adquisición de una tecnología como lo es la tablet. [16]

Esta plataforma ha levantado mucha expectación en la industria y la admiración de los usuarios, cada vez hay más gente que siente la curiosidad de desarrollar videojuegos para esta plataforma. Si bien es cierto muchos países se han sumado a la creación y producción de Videojuegos como hoy en día tenemos a países centroamericanos que se han hecho participe de estos como son los pioneros Panamá y Costa Rica que busca abrirse camino en la industria de los videojuegos en Estados Unidos con la participación de cinco empresas en la feria Game Connection las cuales son “Fair Play Labs”, “Psion Tech”, “Tree Interactive”, “Happy Dog Games” y “Móvil Multimedia” que según los promotores de comercio exterior de Costa Rica afirman que “la industria de videojuegos en el país es muy joven, pero ha demostrado una gran capacidad, creatividad y calidad en sus proyectos”. Por su parte Honduras desde el 06 /03/2010 ya posee su primera consola de video juegos se trata de un joven de tan solo 16 años de edad, incursionó con gran éxito en el campo de la programación y diseño de videojuegos. Se trata de Luis Fernando Cruz Rodríguez, joven hondureño, creador de la primera consola de videojuegos desarrollada completamente en el país. Y también este país ha ido integrando a jóvenes en varios programas de creaciones de Videojuegos como alternativa a la violencia. Posteriormente el salvador a principios del año 2011 en el mes de mayo la compañía **The Domaginarium**, empresa salvadoreña de desarrollo de videojuegos, preparo el lanzamiento de **Parasite**, título con el que debutaron en el sector. Sergio Rosa, diseñador del juego y director creativo, explicó que **The Domaginarium** es la empresa hermana de DoDigital, compañía dedicada a otras ramas del entretenimiento, como películas y series de televisión.

## **1.2 ¿Qué es un videojuego?**

Un videojuego es un medio de entretenimiento que involucra a un usuario, denominado jugador, en una interacción constante entre una interfaz y un dispositivo de video. Los videojuegos recrean entornos y situaciones virtuales, en los que el jugador puede controlar uno o varios personajes para alcanzar objetivos por medio de determinadas reglas [2].

## **1.3 Anatomía de un video Juego**

Para dar una visión general y sin entrar en detalles de implementación de la anatomía de un videojuego. Vamos a ver qué partes lo componen y como se relacionan.

### **1.3.1 Entrada**

Un videojuego necesita comunicarse con el jugador a través de un dispositivo de entrada, Algunos jugadores preferirán usar el teclado al ratón y viceversa. Si además queremos que nuestro juego tenga un toque profesional, hay que dar la posibilidad al jugador de que defina las teclas con las que quiere jugar, además de dar soporte a los diferentes tipos de teclados internacionales. [1]

### **1.3.2 Visualización**

Uno de los elementos principales de un buen juego es un buen motor gráfico, capaz de mover una gran cantidad de Sprites en un juego 2D o una gran cantidad de polígonos (triángulos) en el caso de un juego en 3D es lo que se conoce como visualización [1].

### **1.3.3 Sonido**

El sonido es un aspecto al que, a veces, no se le da la importancia que merece. Tan importante como unos buenos gráficos es un buen sonido. Tanto el sonido como la banda sonora de un juego son capaces de transmitir una gran cantidad de sensaciones al jugador, y eso es, al fin y al cabo, lo que se busca. Tanto como la calidad, la buena sincronización con lo que va sucediendo en pantalla es de suma importancia [1].

### **1.3.4 Comunicaciones**

Cada vez más, los juegos en red van ganando adeptos. La comunicación se realiza normalmente sobre redes TCP/IP. Esto nos permite jugar contra otros jugadores situados en nuestra propia red local o en Internet. La comunicación entre máquinas se lleva a cabo mediante la programación de sockets. Afortunadamente SDL nos provee de herramientas muy potentes para este tipo de comunicación, aunque esto es algo que queda fuera del alcance de este libro [1].

## 1.4 Programación móvil

La generalización en los últimos años de teléfonos móviles, Smartphone, PDAS, etc., ha generado una importante necesidad de aplicaciones para este tipo de dispositivos. Las prestaciones de los dispositivos móviles se incrementan día a día, posibilitando la implementación de aplicaciones muy interesantes como son [3]:

- ✓ Acceso a Internet y conectividad mediante Bluetooth
- ✓ Envío de mensajes cortos
- ✓ Aceleración 2D y 3D
- ✓ Reproducción de video y sonido
- ✓ Cámara integrada
- ✓ Información de localización GPS
- ✓ Censores de orientación
- ✓ etc.

### 1.4.1 Sistemas Operativos Móviles

Son sistemas operativos específicos para este tipo de dispositivos que tienen en cuenta las limitaciones que hemos visto antes.

#### 1.4.1.1 Symbian

Es el resultado de una alianza entre varias empresas de renombre en el mercado tales como Nokia, Sony Ericsson, Samsung, Siemens, Motorola y otras. Symbian posee un núcleo de tiempo real, es un sistema operativo con micro kernel y capacidad Multithreading. Soporta las arquitecturas de los últimos CPU e incluso soporta hardware “Single-Chip” o de un solo chip. Cuenta con un sistema de archivo de alto rendimiento que soporta las últimas memorias NOR, NAND, SD Y MMC. Entre los servicios genéricos que brinda el sistema operativo, se encuentra una base de datos SQL, seguridad integrada contra malware y virus, y soporte para varias plataformas de desarrollo como C++, J2ME, C y MIDP 2.0. En la actualidad, la multinacional Nokia es la que provee mayor cantidad de dispositivos móviles equipados con Symbian, seguida por Sony Ericsson, Motorola, Samsung, Panasonic y otros. Symbian continúa innovando en el mercado de las comunicaciones móviles con tecnologías de última generación. Es un Sistema Operativo Instalado en mayor número de dispositivos móviles en la actualidad, sobretodo en teléfonos móviles [4].

#### 1.4.1.2 Windows Mobile

Windows Mobile es un SO de la familia Windows CE, desarrollado por Microsoft .A pesar de llevar el nombre Windows, no es un sistema derivado ni es una versión recortada del mismo, sino que es un nuevo sistema diseñado específicamente para dispositivos móviles. Los primeros dispositivos que se comenzaron a fabricar con lo que sería el sistema Windows Mobile datan del año 2000. Para ese entonces, fue lanzado como Pocket PC 2000 y estaba basado en Windows CE 3.0 [4].

Este sistema, está estrechamente vinculado a otros productos de la misma marca (servicios Live, Office Mobile, Internet Explorer Mobile, etc.) y cuenta con una interfaz gráfica de muy buena calidad, y muy similar a la de los sistemas operativos Windows.

Sus características están basadas en tener:

- ✓ Kernel unificado  
El kernel de Windows CE puede manejar más de 32000 procesos simultáneos, cada uno con 2GB de memoria virtual compartida.
- ✓ El filesystem soporta archivos de hasta 4GB y encriptación de dispositivos de almacenamiento externo.
- ✓ Variadas arquitecturas  
Trabaja con procesadores de arquitecturas x86, ARM, SH4 y MIPS.

Actualmente, este sistema se encuentra en una buena posición en el mercado, ganando terreno lentamente. Más específicamente, Microsoft tuvo un total de 12% del mercado entre PDAs y smartphones en el primer cuarto de 2006. En primer lugar estuvo Symbian (54,4%) y le siguió Linux con un 21,8%. [4]

#### **1.4.1.3 iPhones OS**

La historia del iPhone OS comienza conjuntamente con el nacimiento del conocido iPhone, en el 2008. Aunque, esta aseveración es discutible, ya que este sistema operativo que corre en el iPhone es en realidad una versión adaptada del OS X. Por lo cual, en este sentido, este sistema ya tiene años en el mercado y ha sido puesto a prueba. El sistema ha sido adaptado, removiendo todos los componentes que no son críticos para un dispositivo móvil, y se le adicionan funcionalidades que si están relacionadas con el mundo de la telefonía móvil. [4]

Sobre la versión modificada del kernel de MAC OS X que corre el iPhone, se encuentran las capas de servicios que componen el teléfono móvil. Existe una gran inclinación en el desarrollo del SO a la interfaz de usuario y las cuestiones de usabilidad. Sin duda el iPhone es el SO para dispositivos móviles que brinda una mejor experiencia de usuario, con un modo de manejo revolucionario basado en su TouchScreen (Pantalla Táctil). [4]

#### **1.4.1.4 Android**

Android OS es el más reciente de los sistemas operativos para móviles del mercado. Está siendo desarrollado por the open Handset Alliance, un grupo de más de 30 empresas de tecnología. Entre ellas la principal participante es google.

Sus características se detallan a continuación.

- ✓ Se trata de un SO abierto, multi-tarea
- ✓ Permite a los desarrolladores acceder a las funcionalidades principales del Dispositivo mediante APIs.
- ✓ “Todas las aplicaciones son iguales”
- ✓ El SO no diferencia entre las aplicaciones básicas del teléfono y las Aplicaciones de terceros.
- ✓ Cualquier aplicación puede ser reemplazada libremente, incluso las que trae Por defecto el SO.

- ✓ Navegador Web integrado basado en el motor Web Kit.
- ✓ Soporte para gráfico 2D y 3D basado en la especificación OpenGL 1.0
- ✓ Base de datos SQLite
- ✓ Soporte multimedia para audio, video e imágenes en varios formatos.
- ✓ Conectividad Bluetooth, EDGE, 3G y Wifi.

Se basa en el kernel de Linux versión 2.6 para las principales funciones como seguridad, manejo de memoria, manejo de procesos, networking y modelo de driver. Android hace público un SDK (Software Development Kit) para que los desarrolladores que lo deseen puedan programar aplicaciones que corran en el SO. El lenguaje de programación utilizado es Java. Las aplicaciones corren sobre una máquina virtual diseñada para ser usada de forma embebida, denominada Dalvik, la cual se ejecuta sobre un kernel de Linux.

## **2- AMBIENTE DE DESARROLLO DE LOS VIDEOJUEGOS**

Como ya sabemos que la gran mayoría de personas están expuesta a todo tipo de tecnología y la gran demanda existente en estos aparatos móviles, decimos emplear nuestros conocimientos y habilidades, en lo que se conoce como Android un Sistema Operativo creado especialmente para Dispositivos móviles y Tablet, es por lo tanto que hemos decidido trabajar bajo esta plataforma por una de las más demandada en el mundo de comunicaciones móviles y utilizando como dispositivos una Tablet para crear nuestro video juego que funcionara en modo grafico 2D.

Antes de que nos introduzcamos a desarrollar nuestro ambiente de desarrollo para las aplicaciones o juegos sobre android, se pretende hacer un plan. Se tendrás que tomar varias decisiones antes de empezar a codificar el juego y cada una de ellas es importante. Decisiones como si quieres el juego para uno o múltiple jugadores, en que idiomas, etc.

Antes de empezar a programar juegos en android es necesario que nos hagamos las siguientes preguntas para empezar a diseñarlo.

- ¿Qué tipo de juego quieres hacer?
- ¿Cuál es el tipo de audiencia al que va dirigido?
- ¿Qué rango de dispositivos móviles voy a soportar?
- ¿Quiero enfocar mi juego a tabletas?
- ¿Quiero hacer mi juego disponible para el mayor número de dispositivos android o solo los más populares?
- ¿Cómo navegara el jugador en el juego?
- ¿Cómo controlara el jugador el juego?
- ¿Quiero hacer dinero con este juego?
- ¿Cuál es la forma más apropiada generar dinero con este juego?

Como observamos que una vez respondidas estas preguntas tendrás un paso adelante en la programación del juego y una idea más clara hacia donde nos vamos a dirige.



### 3- ANDROID

Android es un conjunto de software que constituye un ecosistema para las aplicaciones móviles. Dentro de este conjunto se incluye un sistema operativo móvil, lo que significa que Android está dirigido principalmente a teléfonos inteligentes (o *Smartphone*) y a *tablets*. [6]

#### 3.1 Historia y Éxito de Android

Android es un software desarrollado por Google posteriormente a la compra de la empresa Android Inc. El objetivo de Google a la hora de crear y mantener Android es crear un ecosistema móvil estándar y abierto, para satisfacer las necesidades de fabricantes de dispositivos móviles y tablets. Desde su salida al mercado en el año 2008, Android ha ido escalando puestos en el ranking de ventas, hasta llegar a ser el sistema operativo que llevan el 52.5% de los teléfonos inteligentes vendidos en el tercer cuarto de 2012. Seguido de lejos por Symbian, sistema operativo de la empresa Nokia, el cual solo representa el 16.9%. Todo esto según un estudio realizado por la empresa.

El secreto de este crecimiento tan rápido se debe a las características de las que presume el ecosistema Android. En primer lugar, como ya se ha explicado, Android está pensado para satisfacer los requerimientos de algunas de las empresas fabricantes de smartphones y tablets.

Estas empresas necesitan un ecosistema estándar, que sea ampliamente utilizado por todas las empresas fabricantes de dispositivos móviles que lo deseen. Y es que esto les permite obtener dos beneficios importantes:

- ✓ Por un lado, las empresas fabricantes del hardware, no necesitan embarcarse en el proceso de desarrollo de un sistema operativo propio, el cual no es un proceso sencillo ni barato. Además, estas empresas tienen disponible un S.O. creado y mantenido por la reconocida empresa Google, la cual dota de calidad a todas sus creaciones.
- ✓ Por otro lado, en un ecosistema móvil es muy importante el número de aplicaciones que estén disponibles, pues son estas aplicaciones las que dan utilidad al dispositivo.

En este sentido, Android es además un sistema muy abierto, lo que se traduce en muchas facilidades para los desarrolladores, factor que incrementa aún más el número de aplicaciones que podemos encontrar disponibles para el ecosistema. Pero empresas importantes como Samsung, LG o HTC no se conforman con eso, pues estas quieren tener la opción de forma fácil y barata, personalizar la parte visual de Android, con tal de darle un toque único que se adapte a las características que estos quieren ofrecer a sus usuarios Android. Al ser un sistema muy abierto, permite con facilidad esta personalización, tanto por parte de fabricantes de dispositivos móviles, como por parte de otros desarrolladores que quieran crear su propia interface.

Hay que decir, además, que el sistema operativo Android es de código abierto lo que aun facilita más la labor de las empresas que quieran implementar este S.O. en sus terminales, puesto que estos pueden acceder al código. Eso sí, para obtener la compatibilidad con Android, el fabricante no puede modificar el código completamente a su antojo, hay una serie de requisitos que deberá cumplir. En caso de no cumplirlos, no se pasará el test de compatibilidad y no se estará implementando el ecosistema Android.

Por si fuera poco, además Android es un sistema operativo gratuito, así que el fabricante no tiene que pagar ninguna licencia. Google recibe beneficios económicos a través de la tienda de aplicaciones y de sus servicios, los cuales se financian principalmente mediante publicidad.

En resumen, el único trabajo que deberá llevar a cabo el fabricante que quiera implementar el ecosistema, es el desarrollo de los drivers necesarios para que Android sepa comunicarse con el hardware del dispositivo, el resto queda en manos de si el fabricante quiere personalizar más o menos el sistema. Por tanto, la inversión en software de estos fabricantes puede reducirse mucho gracias a Android.

Esta versatilidad de Android y la idea de tener un ecosistema común es la que ha hecho que un gran número de fabricantes añadan el ecosistema a sus dispositivos móviles y que, por tanto, este S.O. goce de gran popularidad. Otro punto importante que justifica el crecimiento tan rápido que ha experimentado el S.O. tiene que ver con el desarrollador de aplicaciones.

Para empezar, publicar una aplicación en el Android Market que cambio por el nombre google store (la tienda de aplicaciones de Google) es muy barato, únicamente necesitamos una licencia que tiene un coste de 25\$ (unos 20€) para poder publicar aplicaciones durante un tiempo ilimitado. A parte de este pago, Google se lleva el 30% de las ganancias obtenidas con la aplicación, si es que esta es de pago. En caso de ser gratuita, no será necesario pagar nada más. Además, el entorno de desarrollo se puede montar tanto en Windows como en Linux e incluso Mac O.S.

Todas estas facilidades hacen que desarrollar para Android sea bastante más barato que hacerlo para otros ecosistemas móviles. Las aplicaciones además, no sufren ningún tipo de control a la hora de publicarse. Desde el punto de vista de la libertad del desarrollador y de los usuarios esto es una ventaja, pues en el Android Market encontramos aplicaciones que en otras tiendas de aplicaciones de la competencia son censuradas, simplemente por ir en contra de la política de la empresa o por tener contenidos para adultos. Pero también tiene su parte negativa, pues cualquier desarrollador puede publicar una aplicación malintencionada.

Para evitar que el usuario instale una aplicación con fines maliciosos sin darse cuenta, se ha desarrollado el sistema de permisos explicado con anterioridad y que recordamos a continuación. Cualquier aplicación que se instale en el dispositivo y quiera llevar a cabo una serie de accesos a la información externa a la propia aplicación, deberá solicitar al usuario los diferentes permisos necesarios para llevar a cabo estas acciones.

Así que, si por ejemplo, el usuario ve que un videojuego está pidiendo permiso para acceder a la información de los contactos o de las llamadas realizadas, podrá decidir no instalarlo, pues es sospechoso que este tipo de aplicación requiera tales permisos. En cambio, si instala el videojuego estará aceptando estos permisos y, por tanto, la aplicación podrá acceder a dicha información.

Como vemos se trata de un sistema que confía en la prudencia de los usuarios y, al mismo tiempo, en la buena fe de los desarrolladores. Por el momento, este sistema tan abierto está provocando algunas quejas por parte de algunos usuarios que lo consideran poco seguro.

Pero es que el termino abierto, en el sistema operativo Android, se extiende más allá de lo citado anteriormente, pues tampoco hay ninguna restricción que impida instalar aplicaciones externas al Android Market. Por tanto, las aplicaciones, se pueden instalar mediante ejecutables introducidos directamente dentro de la memoria del dispositivo o mediante otras tiendas de aplicaciones gestionadas libremente por otras empresas. Eso sí, la política de permisos se extiende más allá del Android Market, pues afecta a cualquier aplicación, que en caso de no obtener los permisos necesarios no podrá realizar las acciones deseada.

Podemos concluir, por tanto, que el S.O. Android ha cosechado éxito gracias a haberse adaptado a lo que los fabricantes demandaban, así como gracias a la facilidad brindada a los desarrolladores que puedan realizar sus aplicaciones para el ecosistema de forma barata y libre. Ecosistema libre, abierto y gratuito son, por tanto, los principales adjetivos que definen al ecosistema Android.

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android, como pueden ser la App Store de Amazon o la tienda de aplicaciones Samsung Apps de Samsung.[13]

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (*surface manager*), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.[1]

### 3.2- Estructura de Android

El conjunto de software denominado Android incluye: un sistema operativo, software intermedio que trabaja al servicio de las aplicaciones que se encuentran por encima y algunas aplicaciones claves que vienen incluidas desde el principio con el sistema operativo. Un ejemplo de aplicación por defecto es el Android Market, la tienda desde donde podemos comprar o descargar gratuitamente las aplicaciones que los desarrolladores ofrecen a través de ella. Para ver la estructura de todo el paquete Android tenemos la siguiente Figura.

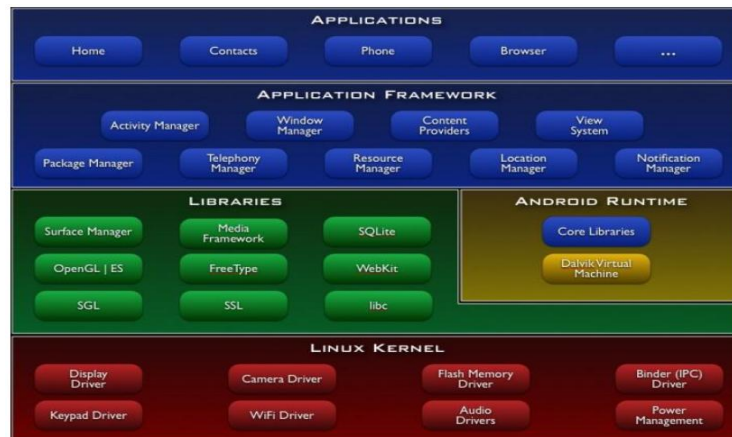


Figura #14

#### Estructura de Android

Como podemos ver en la figura #14, el núcleo del sistema operativo es una modificación del núcleo de Linux. En él se encuentran los drivers que permiten comunicarse con el hardware específico de cada dispositivo que implementa Android. Por encima tenemos una capa con todas las librerías, accesibles a la hora de programar aplicaciones, las cuales hacen uso de los drivers implementados. Entre estas librerías encontramos OpenGL ES, SQLite, SSL, etc.

Dentro del sistema operativo también encontramos la famosa máquina Dalvik. Y es que, en Android, las aplicaciones se ejecutan en una instancia de la máquina Dalvik. Cada instancia es independiente y, por tanto, ejecuta una aplicación de forma cerrada. Este es un buen mecanismo de seguridad, pues nadie puede llegar a entrometerse en la ejecución de una aplicación. De igual forma, los recursos de cada aplicación se encuentran en un fragmento de memoria privada e inaccesible desde fuera de la aplicación.

Y hasta aquí el sistema operativo, ya que en la siguiente capa, encontramos una serie de componentes utilizados por las aplicaciones para realizar funciones determinadas. Entre estos componentes, por ejemplo, se encuentra el Notification Manager, el cual recibe notificaciones de las aplicaciones y las presenta al usuario a través de la barra de notificaciones. Otro ejemplo es el Activity Manager, el cual se encarga de la gestión de las actividades de cada aplicación

Como estos componentes que conforman el framework manejan una información sensible para el usuario, pues en un dispositivo móvil normalmente el usuario introduce bastante información personal, se ha desarrollado un mecanismo de permisos por tal de mantener la información de forma segura.

De esta forma, cuando la aplicación se instala, esta solicita al usuario permisos para acceder a los componentes del framework que sean necesarios por tal de llevar a cabo las acciones pertinentes. El usuario deberá aceptar estos permisos si quiere instalar la aplicación, ya que, en caso contrario, esta aplicación no podrá utilizar los poderes que el permiso solicitado otorga.

Si seguimos con la estructura de Android, un nivel por encima tenemos las aplicaciones, que pueden estar creadas por desarrolladores externos o por la propia Google. De entre las creadas por la empresa Google destacan unas cuantas que, como hemos dicho antes, son clave y vienen incluidas en el paquete.

Aunque, como hemos visto Android es más que un S.O., cuando se habla del sistema operativo Android, se suele incluir todo el paquete de software. Así que durante el resto de la memoria nos dirigiremos de esta forma a la totalidad del software que se engloba dentro del nombre Android.

### **3.3-Estructura de las Aplicaciones de Android**

Las aplicaciones de Android se estructuran en componentes, cada componente de una aplicación juega un papel específico dentro de esta y existe por sí mismo. Eso sí, puede haber componentes que dependan unos de otros. Es decir, a la hora de desarrollar una aplicación para Android no tendremos una sola función principal (o función main) la cual lanzará las diferentes pantallas de la aplicación. Sino que tendremos varios componentes independientes cuya comunicación será llevada a través del S.O.

Además, estos componentes, lejos de tener cada uno una sola función main la cual nos permitiría llevar a cabo una ejecución secuencial de la aplicación, implementan una serie de funciones que serán llamadas por el S.O. cuando se cumpla la condición necesaria para llamar a cada función. Por tanto, las aplicaciones en Android funcionan de forma asíncrona y es el S.O. el que ante las peticiones del usuario van llamando a una y otra función de los diferentes componentes de la aplicación, según convenga.

A través de la API de Android podemos declarar cuatro tipos de componentes: Actividades (Activities), Servicios (Services), Proveedores de Contenido (Content Providers) o Receptores de Transmisión (Broadcast Receivers).

Las **Actividades** son los componentes más importantes, de hecho los videojuegos desarrollados se componen exclusivamente de este tipo de componentes. Una actividad es una pantalla de la aplicación, con su interface de usuario, es decir, lo que en el desarrollo de aplicaciones se conoce habitualmente como vista.

Por ejemplo, en una aplicación que implementa un cliente de correo electrónico, la pantalla que nos permite rellenar un mensaje de texto y nos presenta el botón de envío, junto con el mecanismo para dar respuesta a las diferentes acciones que el usuario pueda llevar a cabo en esta pantalla, sería una actividad. Otra actividad de esta aplicación sería la pantalla que nos presenta una lista con todos los correos que nos han llegado, nuevamente junto con la lógica necesaria para atender las peticiones del usuario.

Un **Servicio**, en cambio, es un componente que no tiene interface propia, sino que se ejecuta en segundo plano y lleva a cabo tareas largas que consuman muchos recursos. Por ejemplo, en caso de querer implementar un reproductor de música, este tendrá un servicio que reproduce esta música. Al mismo tiempo, la aplicación tendrá una actividad que permite que el usuario escoja la canción deseada. La actividad se comunicará con este servicio para pausar, reanudar o cambiar de canción.

En cuando al **Proveedor de Contenido**, este es un tipo de componente que permite almacenar una información para que después pueda ser accedida desde varias aplicaciones. Según cómo configuremos al Proveedor limitaremos dicho contenido a unas aplicaciones muy concretas o no. Se trata del único mecanismo para compartir información entre diferentes aplicaciones que existe en Android.

Por último, se encuentra el **Receptor de Transmisión**, este crea una notificación que se presentará en la Barra de Notificaciones del sistema. Normalmente, una notificación contiene una información acerca de lo que se pretende informar y un apuntador a una actividad o servicio que será lanzado cuando se presione sobre la notificación.

Hay que tener en cuenta que estos componentes son representados por clases Java, lo que significa que, cuando la aplicación se ejecuta, se crean instancias de estos, pudiendo crear más de una instancia del mismo componente. Dichas instancias además son independientes.

### 3.4 Estructura de un Proyecto Android

Para empezar a comprender cómo se construye una aplicación Android vamos a echar un vistazo a la estructura general de un proyecto. Cuando creamos un nuevo proyecto Android en Eclipse se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura será común a cualquier aplicación, independientemente de su tamaño y complejidad. En la siguiente imagen vemos los elementos creados inicialmente para un nuevo proyecto Android:

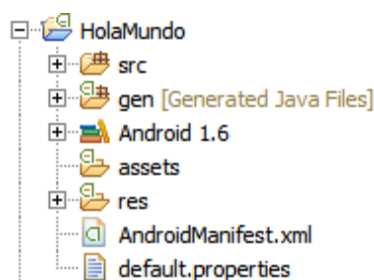


Figura #15 Estructura Proyecto Eclipse

En los siguientes apartados describiremos los elementos principales de esta estructura.

**Carpeta /src/:** Contiene todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc. Inicialmente, Eclipse creará por nosotros el código básico de la pantalla (**Activity**) principal de la aplicación, siempre bajo la estructura del paquete java definido.

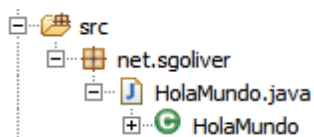


Figura #16 Carpeta SRC eclipse

**Carpeta /res/:** Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc.

**Carpeta /gen/:** Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de compilación de Android genera por nosotros una serie de ficheros fuente en java dirigido al control de los recursos de la aplicación.

**Carpeta /assets/:** Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc. [9]

**Fichero AndroidManifest.xml:** Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, etc...), sus componentes (pantallas, mensajes), o los permisos necesarios para su ejecución. [9]

### 3.5 Entorno de desarrollo de android

Hay que destacar que una de las ventajas de este entorno es que se puede instalar en Windows, en Linux y en Mac O.S. X. Lo único que necesitamos para desarrollar una aplicación es el SDK (Software Development Kit) de Android. Un kit que incluye una serie de herramientas como son: el Traceview, herramienta que nos permite extraer estadísticas de rendimiento de la aplicación; el LogCat, que nos presenta los mensajes que se imprimen desde el código durante la ejecución de una aplicación; y herramientas para generar los instaladores de las aplicaciones que desarrollemos; entre otras herramientas de utilidad.

Además, el SDK incluye la API (Application Programming Interface) de Android, unas librerías que contienen todas las clases y operaciones que debemos utilizar para poder comunicarnos con el S.O. y de esta forma poder, por ejemplo, definir una actividad. En cuanto al IDE (Integrated Development Environment) o entorno que integra las herramientas necesarias para desarrollar aplicaciones, podemos utilizar cualquiera. Incluso podríamos utilizar un editor de textos normal para programar la aplicación y no utilizar un IDE. Aun así, se recomienda encarecidamente el uso de Eclipse. Y es que existe un *plugin* oficial llamado ADT (Android Development Tools) que una vez instalado en Eclipse nos permite acceder a la API y a las diferentes

herramientas que el SDK nos brinda directamente desde la interface de Eclipse, facilitando enormemente el desarrollo de las aplicaciones.

Por último, en cuanto al lenguaje de programación, la API se encuentra en Java y el lenguaje recomendado para programar es este. Java es un lenguaje menos eficiente que C y sus variantes, por lo tanto no se suele utilizar para desarrollar videojuegos. Pero también tiene una gran ventaja y es que, como se ha comentado con anterioridad, es un lenguaje interpretado, lo que significa que se interpreta a través de una máquina virtual en tiempo real.

De esta forma podemos evitar tener que compilar nuestra aplicación para diferentes arquitecturas, pues con una sola compilación, la aplicación funcionará en todos los dispositivos sin problemas.

## **4. ECLIPSE Y ANDROID**

### **4.1 Eclipse**

La plataforma Eclipse consiste en un Entorno de Desarrollo Integrado (IDE) abierto y extensible. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con en un editor de código, un compilador/intérprete y un depurador. [10]

Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad.

### **4.2 Historia de Eclipse**

Gran parte de la programación de Eclipse fue realizada por IBM antes de que se creara el proyecto Eclipse como tal. El antecesor de Eclipse fue VisualAge y se construyó usando Smalltalk en un entorno de desarrollo llamado Envy. Con la aparición de Java en la década de los 90, IBM desarrolló una maquina virtual válida tanto para Smalltalk y Java. La rápida expansión de Java y sus ventajas con miras a una Internet en plena expansión obligaron a IBM a plantearse el abandono de esta maquina virtual dual y la construcción de una nueva plataforma basada en Java desde el principio. El producto final resultante fue Eclipse, que ya había costado unos 40 millones de dólares a IBM en el año 2001. [10]



### 4.3 Instalación de Android SDK y Eclipse

Cada vez que Google publica una nueva versión de su sistema operativo y muchos antes que esta llegue a cualquiera de sus teléfonos de forma oficial, los de Mountain View publican el sdk ( software Development kit) del sistema, que contiene una imagen de este y todos los desarrolladores para adaptarlas a sus aplicaciones a nueva versión antes de que esta llegue a l gran público.

Para poder gestionar todo este sistema de versiones y subversiones en un sistema operativo vivo, como es el caso de Android, resulta necesario un software que se en cargue tanto de la sincronización de los paquetes e imágenes ya instaladas. Para ello google utiliza el SDK Manager, un programa que es sincroniza con los servidores google y que nos informa tanto de la disponibilidad de nuevos paquetes como el estado ,esta aplicación está a disponible tanto para entornos Windows como MAC y Linux.

Cuando este archivo sea descargado se procede a la instalación del mismo ejecutando el instalador SDK Manager.exe, hay algunos sistemas operativos como es el caso de Windows que se necesita configurar una variable de entorno para que pueda funcionar apropiadamente denominado Android\_SDK\_HOME, cuyo contenido ha de ser la ruta completa al directorio de instalación del SDK.

La instalación de eclipse es muy sencilla y solo consiste en descomprimir el contenido de un archivo ZIP, que contiene la aplicación y las librerías, en el directorio de nuestra elección (recomendamos C:\Eclipse).

### 4.4 Plugin ADT para Eclipse

EL ADT (Android Development tools) es el plugin desarrollado por google para permitir la integración del SDK de Android en eclipse, permitiendo el desarrollo de programas android de forma nativa desde eclipse.

La instalación de realiza manualmente desde el propio eclipse siguiendo los siguientes pasos.

1. Seleccionar Ayuda→ instalar nuevo software
2. Hacer clic en Add, en la parte superior derecha de la ventana
3. En el dialogo indicar “ADT Plugin” como nombre y <https://dl-ssl.google.com/android/eclipse> como localización.
4. Hacer click en ok
5. En la ventana que aparezca, marcar la casilla junto a development tolos y hacer clic en next.
6. Hacer clic en el botón next.
7. Leer y aceptar el acuerdo de licencia y hacer clic en finish.
8. Y por último Reiniciar eclipse.

## 4.5 Creación de una Máquina Virtual

El gestor de sdk incluye un emulador Android que podemos ejecutar para probar nuestras aplicaciones. Para crear la máquina virtual correspondiente a una versión concreta de android deberemos abrir el gestor de paquetes SDK y acceder a manager AVDs, dentro del menú tools.

Para crear una instancia del emulador haremos clic en el botón new en la ventana de gestiones de la máquina virtual, procedimiento a rellenar los campos correspondientes del formulario con los valores siguientes.

Name:ICS

Target: android 4.0.1- API Level 15

CPU/ABI: ARM(armabi-v7a)

## 5 LENGUAJE DE PROGRAMACION JAVA

### 5.1 Historia

Java es un lenguaje de programación originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. el lenguaje deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede correr en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora. Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados. [13]

## 5.2 ¿Porque Utilizar Java Como Lenguaje de Programación en Android?

Primeramente cabe señalar que java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles, juegos, contenido basado en web y software de empresa. Tiene alrededor de 9 millones de desarrolladores en todo el mundo, java le permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios.

Java es por ello que ha sido fundamental en la programación de aplicaciones de sistemas operativos móviles como es en nuestro caso Android que fue adquirido para ellos teniendo todos los beneficios legales para su utilización.

Java es toda una tecnología orientada a la programación de software con el cual podemos realizar cualquier tipo de programas. Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE. Pero Java no se queda ahí, ya que en la industria para la programación dispositivos móviles también usa en gran medida este lenguaje. La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma nos referimos a la máquina virtual de Java (Java Virtual Machine). Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código fuente funcione sobre cualquier plataforma de software y hardware, es decir, un mismo programa puede ejecutarse en varios sistemas sin tocar el código fuente. Por ejemplo: Unos programas en java escritos para Linux pueden ser ejecutados en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras. Finalmente concluimos que Java nos brinda una solución para la programación de en todo tipo de plataformas.

### 5.2.1 Características y Propiedades de Java

- Escribir software en una plataforma y ejecutarla virtualmente en otra
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, productos de consumo y prácticamente cualquier otro dispositivo electrónico

## 6. MOTOR LIBGDX PARA ECLIPSE Y ANDROID

### 6.1 ¿Que es Libgdx?

LibGDX es un framework multiplataforma basado en Java y OpenGL, pensado para la creación de videojuegos.

Lo más destacable de éste es que crea un núcleo jugable (core project) y el solito se encarga de “transformarlo” según la plataforma a la que va destinado. De esta forma, creas un juego y lo puedes ejecutar para Android, Windows/Linux/macOS, navegador y recientemente iOS, es por ello que hemos escogido este motor para nuestro videojuego, por la gran documentación que este posee, ya que tiene hasta su propia página web para consultas y otros recursos.[15]

Está escrito en Java con una mezcla de C/C++ para dar soporte y rendimiento a tareas relacionadas con el uso de la física y procesamiento de audio.

LibGDX está compuesto por una serie de componentes que serán comunes a todas las aplicaciones.

Marco de **Aplicación**, que manejará el bucle principal y además estará encargado del ciclo de vida, es decir, los eventos de creación, destrucción, pausa y resume de la misma.

Un componente de **Gráficos** que nos permitirá gestionar la representación de imágenes y objetos gráficos en la pantalla.

Un componente de **Audio**, que nos facilitará el acceso a los sonidos y música de la aplicación.

Un componente de **Entrada y Salida** para leer y escribir los diferentes ficheros de datos como por ejemplo, imágenes, archivos de configuración, sonidos, música, texturas,...

Un componente de **Entrada** que gestionara la entrada a través del teclado, pantalla táctil o acelerómetro.

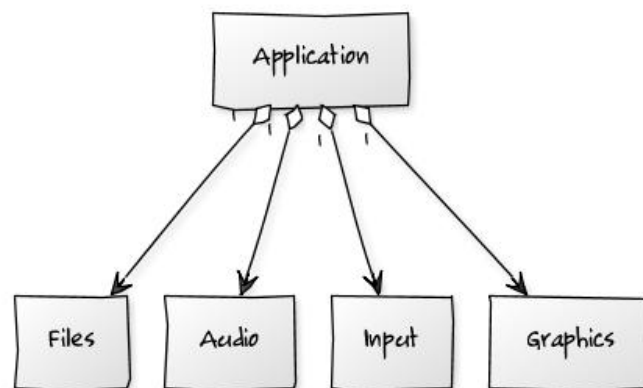


Figura #17 Estructura Libgdx

### 6.1.1 Ventajas:

- Soporte 2d full (bajo y alto nivel)
- Mucha documentación, tutoriales, ejemplos de código
- Releases en forma periódica la última fue en setiembre y anterior en Mayo
- Se puede probar en desktop (antes de subir al Mobil)
- Maneja Audio, input (usuario), física, matemática, archivos
- Más poderoso que Andengine
- libGDX te da un acceso más fácil a Bajo nivel
- Posibilidad de tomar un juego hecho en java jar 2d o 3d y adaptarlo a libgdx para q funcione nativo en android, eso hicieron con el juego Droid Invaders 3d.

### 6.1.2 Desventaja:

- El soporte de alto nivel en 3-d está en construcción actualmente

## 6.2 Ciclo de Vida De una Aplicación LIBGDX

Este apartado describe cómo se estructura una aplicación libgdx.

### 6.2.1 La aplicación

Es el principal punto de entrada para cualquier aplicación libgdx. La interfaz Application determina la plataforma y los gráficos de fondo que serán utilizados por la aplicación. La interfaz también proporciona los métodos para acceder a **gráficos, audio, archivos de entrada y módulos de E/S**. También da acceso a un módulo de **Logging** el cual funciona en todas las plataformas.

**LibGDX** es compatible actualmente con dos back-ends de aplicaciones de escritorio (lwjgl y JOGL) y otra para Android.

Para crear una aplicación **libGDX**, se ha de implementar la interfaz **ApplicationListener** primero.

## DISEÑO METODOLÓGICO

### Tipo De estudio

El tipo de estudio que hemos aplicado en este proyecto ha sido de tipo de **investigación aplicada** con periodo de tiempo, según el tiempo de investigación, nuestro trabajo es un estudio **transversal**, ya que se realizó el análisis de la información recolectada en un periodo limitado, para luego proceder a la elaboración de la aplicación.

### Descripción del Juego

“**ToursNica**”: nombrado así por el objetivo perseguido en nuestro videojuego, ToursNica, viene representando al turismo de Nicaragua. La Abreviatura empleada en la palabra Toursnica no significa nada, simplemente lo tomamos de la traducción de nuestro Videojuego en Ingles “Nicaragua tour” cuya traducción es “Paseo por Nicaragua”.

La mecánica de nuestro video juego tiene como personaje Principal (ficticio) a un turista español, al que llamaremos Andrés, este viaja en un auto rentado desde Managua hacia Granada y como personaje secundario a Caramanzai, un espirito que asido despertado y que tratara por todos sus medio alcanzar al protagonista en su carro fantasma para robársele el alma.

El protagonista viaja en una carretera de doble carril el cual permitirá aventajar y esquivar los obstáculos (vacas, caballos y perros) que se le presentaran en el transcurso de su viaje y que son frutos de las maniobras de ataques de Caramanzai,

Cada obstáculo esquivado se le sumara un punto al protagonista y el carro enemigo retrocederá 25 pixeles de distancia acorde al ejes de las x y cada obstáculo colisionado se le restará un punto, y hará que el carro enemigo se acerque 50 pixeles de distancia de acorde al eje de las x con respecto al protagonista, hay que estar bien atento por que los obstáculo aparecerán aleatoriamente en cual quiera de los dos carriles y de la misma manera cada cierto tiempo aparecerá un baúl que contiene las vidas del protagonista. Si caramanzai alcanza al protagonista y este no ha ganado ninguna vida se le roba el alma y el juego finaliza, si el tiempo del juego finaliza y el protagonista sigue con vida el juego finalizará exitosamente y el protagonista ha ganado la partida.

## **Procedimiento de desarrollo del videojuego**

El Desarrollo de Videojuegos es la actividad por la cual se diseña y se crea un Video juego, desde el concepto inicial hasta el video juego en su versión final. Esta es una actividad que involucra profesionales de la informática, el diseño, el sonido, la actuación, etc...

Se realizó un estudio y análisis de procesos de desarrollo de videojuegos y dicha investigación nos llevó a concluir que la industria se aferró muchos años a utilizar la metodología cascada (Keith, Waterfall Gameevelopment, 2009) y muchas compañías siguen creando productos de esta manera; del estudio mencionado se desprende que no existen procesos específicos para el desarrollo de videojuegos que sean públicos; posiblemente existan algunos procesos cerrados en el sentido de que su información, modelos, plantillas y herramientas no están disponibles al público en general y por lo tanto puede considerarse que existe la necesidad de modelos de procesos que puedan ser utilizados por la industria en general y que en su momento contribuyan al desarrollo de este tipo de aplicaciones. [2]

El desarrollo de un Video Juego Generalmente Sigue el siguiente proceso.

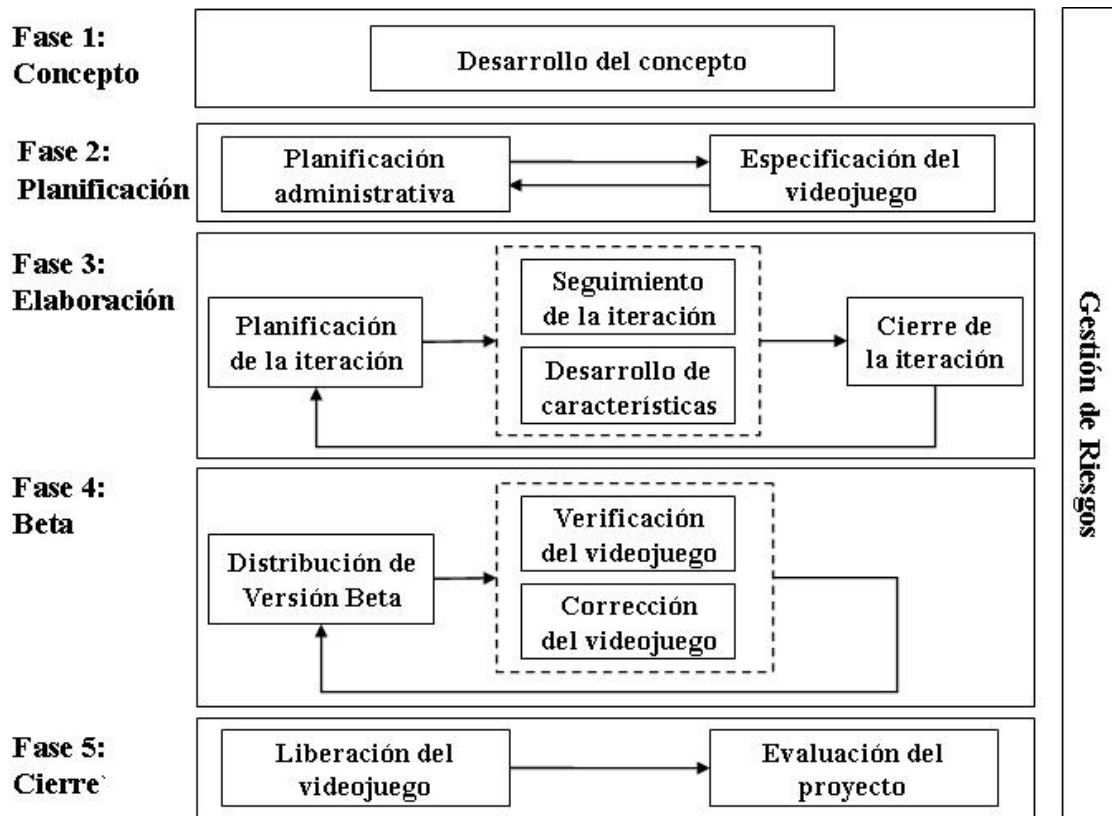


Figura 18 Proceso del desarrollo del videojuego

**Concepto:** tiene como objetivo principal definir el concepto del videojuego lo que implica definir los aspectos de negocio (público objetivo, modelo de negocio), de elementos de juego (principales características, gameplay, personajes e historia entre otros) y técnicos (Lenguajes y Herramientas para el desarrollo).

**Historia de Nuestro juego**

**Personajes:** Andres, Karamanzai

**Herramientas de desarrollo:** Eclipse,

**ADT**(android development tools) es el kit de desarrollo de android que trae por defecto los siguientes paquetes:

El **SDK**: las distintas librerías necesarias para que nuestra aplicación sea compatibles con todas las versiones de android disponible.

**JDK** de java: al igual que el **ADT** es el kit desarrollo de java , el cual contienen

**Lenguaje de programación:** Java

**Planificación:** La fase tiene como objetivo principal planificar las restantes Fases del proyecto. Para ello es necesario definir el cronograma del proyecto Junto con sus principales hitos, conformar el equipo para la fase de elaboración de acuerdo a las necesidades técnicas del proyecto, determinar y tercerizar las tareas que el equipo no pueda cumplir, definir el presupuesto y especificar las características. Esto último consiste en describir, estimar y priorizar cada una de las características funcionales y no



funcionales que definen el videojuego. Una característica funcional representa una funcionalidad del videojuego desde el punto de vista del usuario final, mientras que, una característica no funcional representa una propiedad o cualidad que el videojuego debe presentar. La planificación que se obtiene en esta fase es flexible ya que en cada iteración de la fase de elaboración se puede modificar para adaptarse a los cambios y reflejar la situación actual del proyecto.

Ver tabla #3 a la tabla #9.

**Elaboración:** El objetivo de esta fase es implementar el videojuego. Para ello se trabaja en forma iterativa e incremental para lograr una versión ejecutable del videojuego al analizar cada iteración. Estas se dividen en tres etapas, en la primera se planifican los objetivos a cumplir, las métricas a utilizar en el seguimiento, las características a implementar y las tareas necesarias para ello. En la segunda se desarrollan las características planificadas a través de la ejecución de las tareas que la componen. Al mismo tiempo se realiza el seguimiento para mantener la visión y el control de la iteración en base a los objetivos planteados. La tercera y última implica la evaluación del estado del videojuego y de lo ocurrido en el transcurso de la iteración para actualizar el plan de proyecto respecto a la situación actual. Con esta forma de trabajo se puede evaluar el avance del proyecto, lo cual permite realizar cambios a tiempo y tomar decisiones para cumplir con los plazos planificados. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar la productividad.

**Beta:** La fase tiene como objetivos evaluar y ajustar distintos aspectos del videojuego como por ejemplo gameplay, diversión, curva de aprendizaje y curva de dificultad, además de eliminar la mayor cantidad de errores detectados. Se trabaja en forma iterativa liberando distintas versiones del videojuego para verificar. Para ello primero se distribuye la versión beta del videojuego a verificar y se determinan los aspectos a evaluar. Mientras esta se verifica, se envían reportes con los errores o evaluaciones realizadas. Estos reportes son analizados para ver la necesidad de realizar ajustes al videojuego. Se puede optar por liberar una nueva versión del videojuego para verificar una vez que se realizan los ajustes. El ciclo termina cuando se alcanza el criterio de finalización establecido en el plan del proyecto.

**Cierre:** Esta fase tiene como objetivos entregar la versión final del videojuego al cliente según las formas establecidas y evaluar el desarrollo del proyecto. Para la evaluación se estudian los problemas ocurridos, los éxitos conseguidos, las soluciones halladas, el cumplimiento de objetivos y la certeza de las estimaciones. Con las conclusiones extraídas se registran las lecciones aprendidas y se plantean mejoras a la metodología.

**Gestión de riesgos:** Esta fase se realiza durante todo el proyecto con el objetivo de minimizar la ocurrencia y el impacto de problemas. Esto se debe a que distintos riesgos pueden ocurrir en cualquiera de las fases, por lo cual siempre debe existir un seguimiento de los mismos. Para cada uno de los riesgos que se identifican se debe establecer la probabilidad y el impacto de ocurrencia, mecanismos de monitoreo, estrategia de mitigación y plan de contingencia.

## Estudio de Factibilidad

El hardware que se propone para el desarrollo de Nuestro Videojuego es válido tanto para su desarrollo como para su implementación.

### Factibilidad Técnica

En esta alternativa proponemos, en cuanto al hardware, específicamente una Tablet donde se debe instalar, este debe cumplir con los siguientes requerimientos.

CANTIDAD	DESCRIPCION
1	Tablet Marca: Mittraveler Modelo: M758A Versión de Android: 4.0.3 o superior Tamaño: 7pulgadas Ram: 512MB DDR3 Memoria interna: 2G a mas Resolución pantalla:800px *480px Redimiento:2D Procesador: Allwinner AL31.0GHz

Tabla #1 Factibilidad Técnica

**Factibilidad Económica**

A Continuación se detalla el resultado del análisis de la cotización realizada para esta alternativa en Cuanto a software, Hardware y recursos humanos.

**Hardware**


<b>MULTICOMERCIAL. S.A.</b> Avenida Principal de Altamira # 558 (Edificio CECA de Tres Pisos) Managua, Nicaragua		 <b>FACTURACION</b>	PBX: (505) - 2770537 FAX: (505) - 2773989 Apartado Postal C-106 e-mail: ceca@ceca.com.ni												
<b>CLIENTE: MISCELANEO</b> ATENCION A: COU: REC./ENVIO: DIRECCION: COR		RUC #J0310000018731  <b>No.: COU-0044777</b> Fecha: 28/08/2013 1:43:08PM Vendedor: HENRY G. ORTIZ													
<table border="1"> <thead> <tr> <th>Cant.</th> <th>Código</th> <th>Descripción</th> <th>SubTotal</th> <th>Porc. Descto</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td align="center">1.00</td> <td>COM-AA000298 TABLET</td> <td>Mitraveler 7P MTACTIL ANDROID 4.1</td> <td align="right">C\$ 3,521.00</td> <td align="center">0.00%</td> <td align="right">C\$ 3,521.00</td> </tr> </tbody> </table>	Cant.	Código	Descripción	SubTotal	Porc. Descto	Total	1.00	COM-AA000298 TABLET	Mitraveler 7P MTACTIL ANDROID 4.1	C\$ 3,521.00	0.00%	C\$ 3,521.00			
Cant.	Código	Descripción	SubTotal	Porc. Descto	Total										
1.00	COM-AA000298 TABLET	Mitraveler 7P MTACTIL ANDROID 4.1	C\$ 3,521.00	0.00%	C\$ 3,521.00										
NOTAS															
Observaciones:	* Precios en Córdoba (COR) * Cotización válida por 15 días * Productos sujetos a disponibilidad														
			Subtotal: C\$ 3,521.00 Descuento: C\$ - SubTotal - IVA: C\$ 528.15 <b>Total General: C\$ 4,049.15</b>												

Figura #19 Cotización Tablet

**Software**

Sistema operativo Android 4.0.3 licencia Gratuita y de libre distribución.

**Recursos Humanos**

Cantidad	Personal	meses	Salario mensual	totales
2	Programadores	6	\$350	\$4200
1	Analista	6	\$500	\$3000
1	Encargado de Pruebas	2	\$150	\$300
			total	\$ 7500

Tabla #2 Factibilidad Recursos Humanos

- Son 2 Programadores lo que implica un costo total por programador de \$350 por mes
- El encargado de prueba verificara el correcto funcionamiento del Videjuego una vez que ha sido concluido. Por lo que su salario es de 2 meses

**Factibilidad operacional**

Para la realización de este Videjuego, consideramos necesario el siguiente personal:

- 2 Programadores
- 1 Analista
- Un encargado de pruebas

Para operar el Videjuego, se requiere solamente de una persona, el usuario final.

**Cronograma de Actividades del Proyecto Tours Nica.**

Task name	Start	Finish	Resource Names
1 Desarrollo de temas, Objetivos Generales	Sat 09/03/13	Tue 12/03/13	Proceso, Desarrollo
2 Desarrollo del Guion del Videojuego	Thu 14/03/13	Sat 16/03/13	Proceso, Desarrollo
3 Documentación del Lenguaje a utilizar	Sat 23/03/13	Sat 30/03/13	Proceso, Desarrollo
4 Justificación, Agradecimiento, planeamiento	Sat 30/03/13	Fri 05/04/13	Proceso, Desarrollo
5 Marco conceptual	Sat 06/04/13	Fri 19/04/13	Desarrollo/concepto
6 Diseño de imágenes del videojuego	Sat 20/04/13	Fri 26/04/13	Planificación
7 Recopilación de información de clase	Fri 26/04/13	Fri 17/05/13	Planificación
8 Programación videojuego fase 1	Fri 26/04/13	Fri 10/05/13	Planificación
9 Mundo o escenario del videojuego	Sat 11/05/13	Sat 18/05/13	Planificación
10 Características de los personajes relacionados	Mon 20/05/13	Fri 24/05/13	Planificación
11 Ejecución del videojuego	Mon 24/06/13	Mon 24/06/13	Elaboración
12 Evaluación de las características funcionales	Thu 27/06/13	Fri 12/07/13	Elaboración
13 Objetivo cumplir del videojuego	Sat 15/06/13	Sat 22/06/13	Elaboración
14 Actualizar el plan del proyecto embasado	Tue 25/06/13	Fri 12/07/13	Elaboración
15 versión 1.1 videojuego Tours Nica	Mon 05/08/13	Fri 09/08/13	Beta
16 versión 1.2 videojuego Tours Nica	Tue 13/08/13	Fri 23/08/13	Beta
17 versión 1.3 videojuego Tours Nica	Mon 26/08/13	Sat 07/09/13	Beta
18 Marco conceptual	Wed 11/09/13	Sun 15/09/13	Gestion de Riesgo
19 Diseño de imágenes del videojuego	Tue 17/09/13	Fri 20/09/13	Gestion de Riesgo
20 Actualizar el plan del proyecto embasado	Wed 25/09/13	Fri 27/09/13	Gestion de Riesgo
21 version final del video Juego	Fri 27/09/13	Fri 11/10/13	Gestion de Riesgo

Tabla #3 Cronograma de actividad #1

Task Name	Start	Finish	Resource Names
6			
1	Sat 09/03/13	Tue 12/03/13	Proceso, Desarrollo
2	Thu 14/03/13	Sat 16/03/13	Proceso, Desarrollo
3	Sat 23/03/13	Sat 30/03/13	Proceso, Desarrollo
4	Sat 30/03/13	Fri 05/04/13	Proceso, Desarrollo
5	Sat 06/04/13	Fri 19/04/13	Desarrollo/concepto
6	Sat 20/04/13	Fri 26/04/13	Planificacion
7	Fri 26/04/13	Fri 17/05/13	Planificacion
8	Fri 26/04/13	Fri 10/05/13	Planificacion
9	Sat 11/05/13	Sat 18/05/13	Planificacion
10	Mon 20/05/13	Fri 24/05/13	Planificacion
11	Mon 24/06/13	Mon 24/06/13	Elaboracion
12	Thu 27/06/13	Fri 12/07/13	Elaboracion
13	Sat 15/06/13	Sat 22/06/13	Elaboracion
14	Tue 25/06/13	Fri 12/07/13	Elaboracion
15	Mon 05/08/13	Fri 09/08/13	Beta
16	Tue 13/08/13	Fri 23/08/13	Beta
17	Mon 26/08/13	Sat 07/09/13	Beta
18	Wed 11/09/13	Sun 15/09/13	Gestion de Riesgo
19	Tue 17/09/13	Fri 20/09/13	Gestion de Riesgo
20	Wed 25/09/13	Fri 27/09/13	Gestion de Riesgo
21	Fri 27/09/13	Fri 11/10/13	Gestion de Riesgo

Tabla #4 Cronograma de actividad 2

Tabla #5 Cronograma de actividad 3

Task Name	Start	Finish	Resource Names
1 ✓ Desarrollo de temas, Objetivos Generales	Sat 09/03/13	Tue 12/03/13	Proceso,Desarrollo
2 ✓ Desarrollo del Guion del Videojuego	Thu 14/03/13	Sat 16/03/13	Proceso,Desarrollo
3 ✓ Documentacion del Lenguaje a utilizar	Sat 23/03/13	Sat 30/03/13	Proceso,Desarrollo
4 ✓ Justificacion,Agradecimiento,planeamiento	Sat 30/03/13	Fri 05/04/13	Proceso,Desarrollo
5 Marco conceptual	Sat 06/04/13	Fri 19/04/13	Desarrollo/concept
6 Diseño de imagenes del videojuego	Sat 20/04/13	Fri 26/04/13	Planificacion
7 ✓ Recopilacion de informacion de clases	Fri 26/04/13	Fri 17/05/13	Planificacion
8 Programacion videojuego fase1	Fri 26/04/13	Fri 10/05/13	Planificacion
9 ✓ Mundo o escenario del videojuego	Sat 11/05/13	Sat 18/05/13	Planificacion
10 ✓ Caracteristicas de los personajes relacionados	Mon 20/05/13	Fri 24/05/13	Planificacion
11 ✓ Ejecucion del videojuego	Mon 24/06/13	Mon 24/06/13	Elaboracion
12 ✓ Evaluacion de las caracteristicas funcionales	Thu 27/06/13	Fri 12/07/13	Elaboracion
13 ✓ Objetivo cumplir del videojuego	Sat 15/06/13	Sat 22/06/13	Elaboracion
14 Actualizar el plan del proyecto embasado	Tue 25/06/13	Fri 12/07/13	Elaboracion
15 version 1.1 videojuego Tours Nica	Mon 05/08/13	Fri 09/08/13	Beta
16 version 1.2 videojuego Tours Nica	Tue 13/08/13	Fri 23/08/13	Beta
17 version 1.3 videojuego Tours Nica	Mon 26/08/13	Sat 07/09/13	Beta
18 Marco conceptual	Wed 11/09/13	Sun 15/09/13	Gestion de Riesgo
19 ✓ Diseño de imagenes del videojuego	Tue 17/09/13	Fri 20/09/13	Gestion de Riesgo
20 ✓ Actualizar el plan del proyecto embasado	Wed 25/09/13	Fri 27/09/13	Gestion de Riesgo
21 ✓ version final del video Juego	Fri 27/09/13	Fri 11/10/13	Gestion de Riesgo

Task Name	Start	Finish	Resource Names
1 ✓ Desarrollo de temas, Objetivos Generales	Sat 09/03/13	Tue 12/03/13	Proceso, Desarrollo
2 ✓ Desarrollo del Guion del Videojuego	Thu 14/03/13	Sat 16/03/13	Proceso, Desarrollo
3 ✓ Documentación del Lenguaje a utilizar	Sat 23/03/13	Sat 30/03/13	Proceso, Desarrollo
4 ✓ Justificación, Agradecimiento, planeamiento	Sat 30/03/13	Fri 05/04/13	Proceso, Desarrollo
5 Marco conceptual	Sat 06/04/13	Fri 19/04/13	Desarrollo/concept
6 Diseño de imágenes del videojuego	Sat 20/04/13	Fri 26/04/13	Planificación
7 ✓ Recopilación de información de clase:	Fri 26/04/13	Fri 17/05/13	Planificación
8 Programación videojuego fase 1	Fri 26/04/13	Fri 10/05/13	Planificación
9 ✓ Mundo o escenario del videojuego	Sat 11/05/13	Sat 18/05/13	Planificación
10 ✓ Características de los personajes relacionados	Mon 20/05/13	Fri 24/05/13	Planificación
11 ✓ Ejecución del videojuego	Mon 24/06/13	Mon 24/06/13	Elaboración
12 ✓ Evaluación de las características funcionales	Thu 27/06/13	Fri 12/07/13	Elaboración
13 ✓ Objetivo cumplir del videojuego	Sat 15/06/13	Sat 22/06/13	Elaboración
14 Actualizar el plan del proyecto embajada	Tue 25/06/13	Fri 12/07/13	Elaboración
15 versión 1.1 videojuego Tours Nica	Mon 05/08/13	Fri 09/08/13	Beta
16 versión 1.2 videojuego Tours Nica	Tue 13/08/13	Fri 23/08/13	Beta
17 versión 1.3 videojuego Tours Nica	Mon 26/08/13	Sat 07/09/13	Beta
18 Marco conceptual	Wed 11/09/13	Sun 15/09/13	Gestion de Riesgo
19 Diseño de imágenes del videojuego	Tue 17/09/13	Fri 20/09/13	Gestion de Riesgo
20 Actualizar el plan del proyecto embajada	Wed 25/09/13	Fri 27/09/13	Gestion de Riesgo
21 version final del video Juego	Fri 27/09/13	Fri 11/10/13	Gestion de Riesgo

Tabla #6 Cronograma de actividad 4



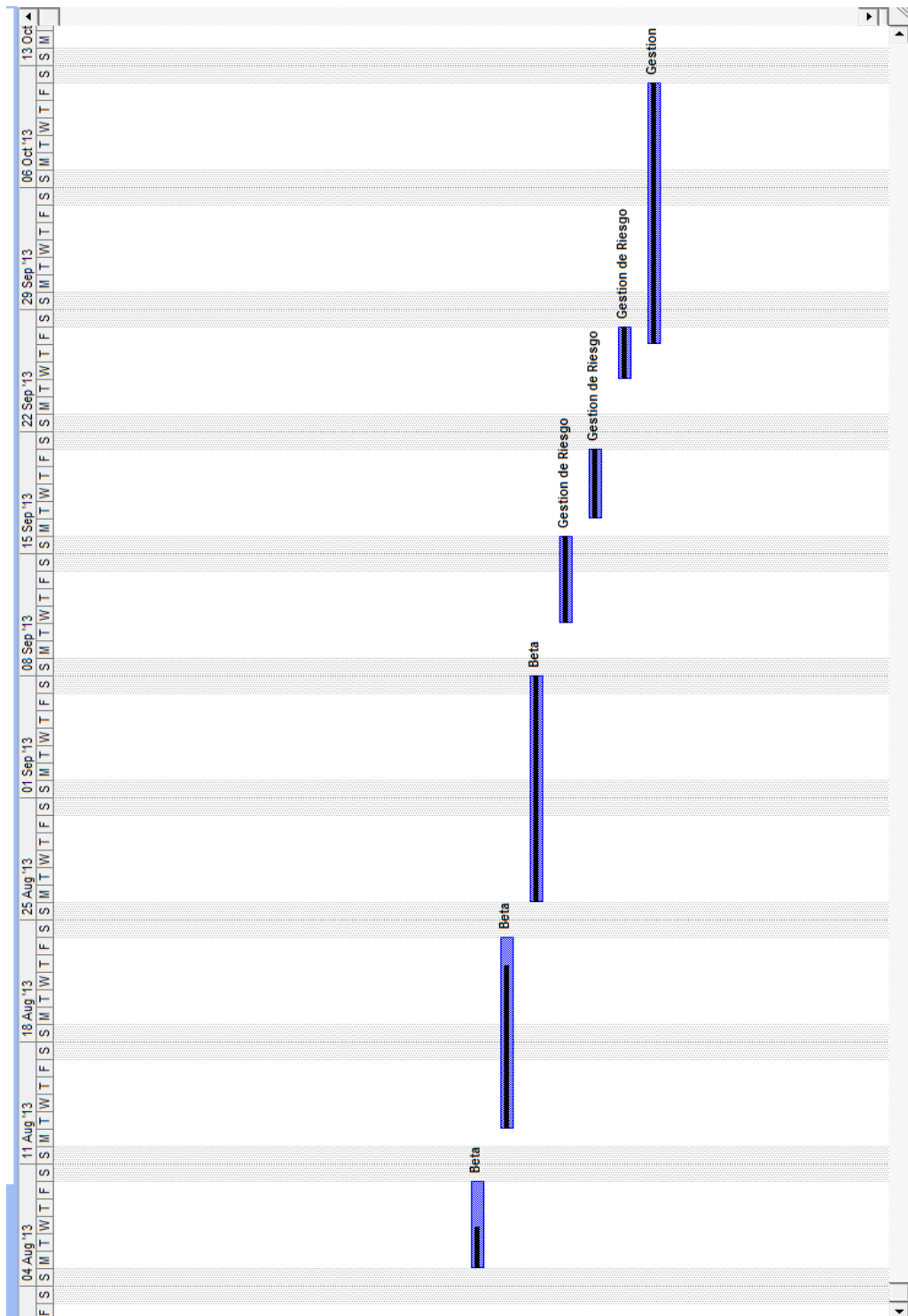


Tabla #7 Cronograma de actividad 5

## Entorno de trabajo haciendo uso de eclipse y el plugin oficial de android el ADT

En esta imagen podemos apreciar tres proyectos diferente generado con el asistente de Libgdx (framework utilizado para el desarrollo del video juego) el primer proyecto SuperNicatourv2 , es el proyecto común tanto para el proyecto android como el proyecto desktop, es en este proyecto donde desarrollamos todo nuestro código fuente.

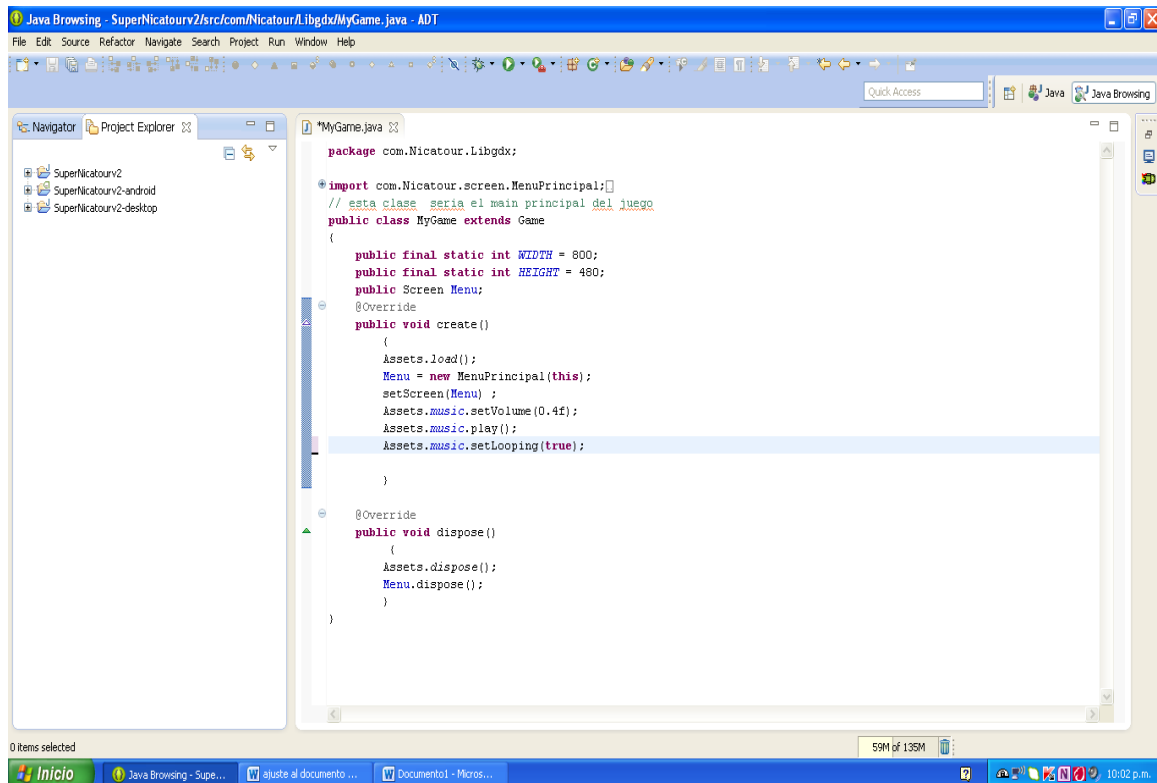


Figura #20 Ambiente de Programación Eclipse 1

En esta imagen podemos apreciar en el lateral izquierdo las clases diseñada y utilizada en nuestro proyecto

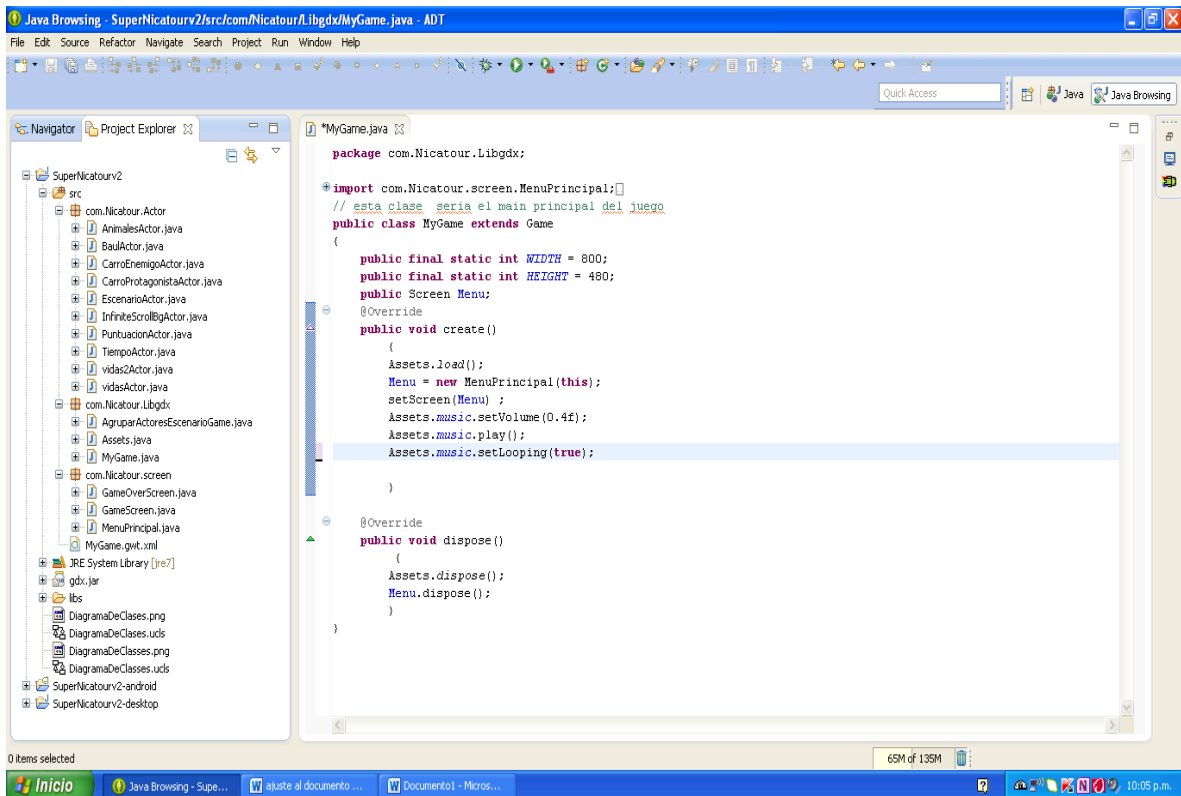


Figura #21 Ambiente de Programación Eclipse 2

En esta imagen podemos apreciar la estructura de nuestro proyecto Android entre lo que se destacan la clase principal y los recurso de nuestra carpeta assets(imágenes ,sonido y música)

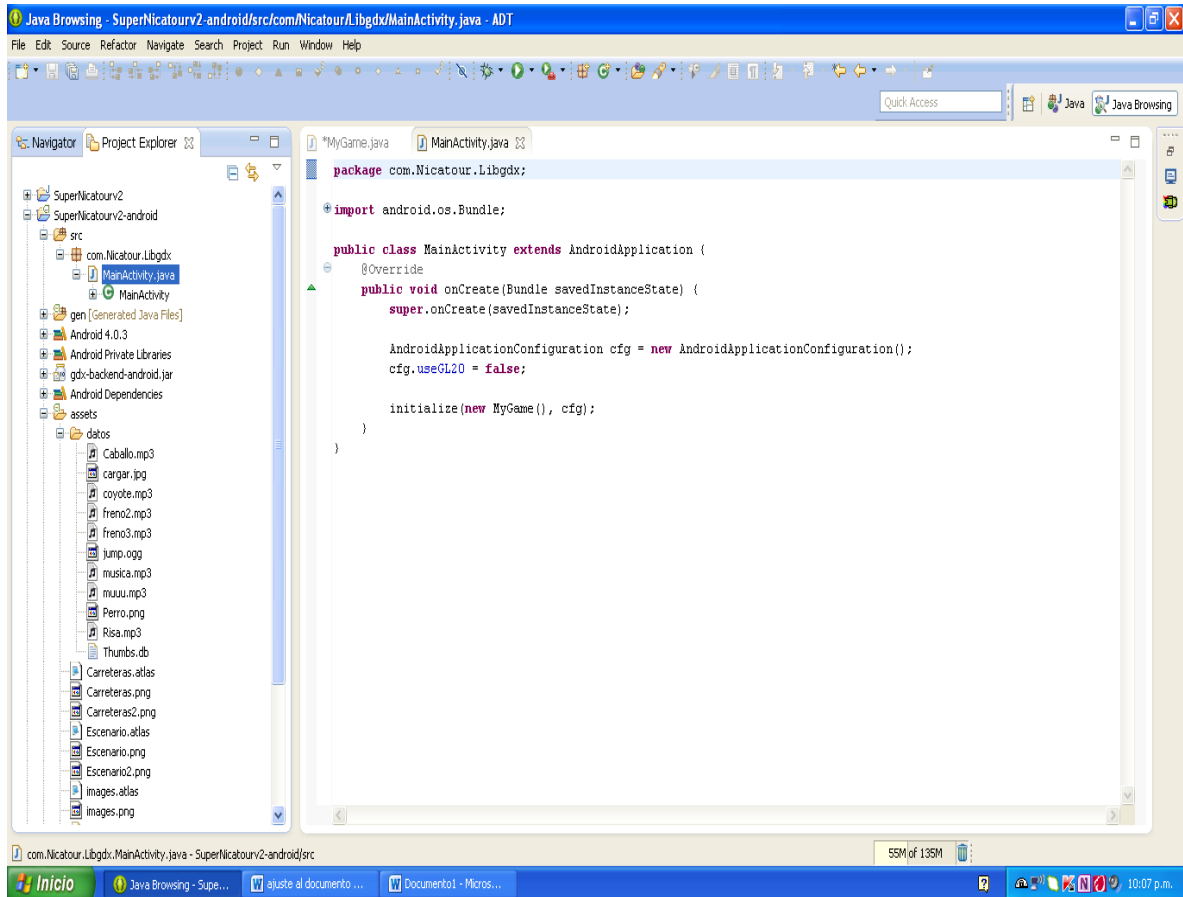


Figura #22 Ambiente de Programación Eclipse 3

## DESARROLLO

### Elaborar el guión del videojuego “Tours Nica”.

#### Resumen general de la historia

Andrés, es un turista español, de ojos claros y test morena decide emprender un viaje a Nicaragua, con el objetivo de vacacionar e interesado en conocer las bellezas culturales y naturales que el país ofrece.

El destino que llama la atención es la encantadora ciudad colonial de (Granada) quiere disfrutar de sus imponentes isletas, playa, secretos e historia.

Ansioso por emprender su viaje y aprovechando el descuento que le ofrecen, renta un auto estilo clásico, con las condiciones necesarias para realizar el recorrido.

El auto que ha elegido está encantado con el fantasma de caramanzai, el último turista que lo utilizó y murió en un accidente tratando de esquivar un lote de ganados. Con el fin de robarle su alma para si el poder descansar en paz.

Cuando Andrés aborda el auto, sin darse cuenta despierta al espíritu de caramanzai, es allí donde comienza el interesante viaje de nuestro amigo, cargado de suspenso, ambición y aventura. Ya en este momento las vacaciones planeadas por Andrés no era la que el destino le tenía preparada.

En el transcurso de su viaje hacia su destino, Andrés nota algo muy extraño que lo mantiene pensativo y es que el observa que solo el transita por la carretera lo que despierta inquietud, temor y pone atento al retrovisor viendo si encuentra compañía en esa carretera tan solitaria, de repente en un abrir y cerrar de ojos le parece haber visto un auto del mismo modelo que lo viene siguiendo, cuando intenta verificar nuevamente ya el auto ha desaparecido, este acontecimiento hace que Andrés se ponga más nervioso y acelera, por lo que él quiere llegar lo antes posible a Granada.

Pasado el tiempo Andrés logra ver nuevamente el auto, pero lo extraño es que de repente se vuelve transparente o invisible Y cada vez se le acerca más, ahora es notable que Andrés está siendo perseguido, y es cuando empiezan a aparecer obstáculos en la carretera que al esquivarlos le roban tiempo y, le impiden acelerar.

Andrés sigue conduciendo cuando de repente escucha una voz que le dice “Andrés no voltees hacia atrás y no pierdas tiempo tu alma está en peligro el auto que te persigue es producto de las fuerza del mal y quiere robarte el alma para reincarnarse, evita colisionar con los animales que te restará tiempo y solo te queda una hora para llegar”

Nuestro amigo tendrá que llegar lo más rápido posible a Granada, en el transcurso de la carretera se le aparecerán baúles brillantes enviados por las fuerzas del bien para ayudarlo a enfrentar las fuerzas del mal.

## Documento de Diseño

### 1. Imágenes utilizadas

A continuación se presenta las imágenes empaquetada en un TextureAtlas



Figura #23 Texture Atlas 1

### 3. Descripción de los personajes

**Actor carro protagonista:** es el actor que manipulará el jugador para ganar la partida haciendo uso de la pantalla táctil, deslizando el dedo de arriba hacia abajo según convenga para esquivar los obstáculos.



Figura #24 Carro protagonista.

1. **Actor carro enemigo:** este actuara en base a las acciones del carro protagonista es decir si el carro protagonista colisiona con los obstáculos esto provocará que los puntos ganados se le resten y que el carro enemigo avance, si este logra colisionarle se le robará una vida.



Figura #25 Carro Enemigo

**Actor obstáculo.** Está compuesto por tres tipos de animales domésticos (vacas, caballos y perros) que aparecerán de forma aleatoria y en diferentes colores tanto en el carril 1 como en el carril 2 de la carretera, el jugador tendrá que estar atento para evitar colisionarlo.



Figura #26 Obstáculos

**Actor baúl:** es el encargado de sumarle vida al jugador cada vez que este colisione con el carro protagonista, aparecerá cada cierto tiempo de forma aleatoria en el carril 1 como en el carril2, razón por la cual hay que estar atento para poder ganar vidas que nos permita ganar la partida del juego.



Figura #27 Baúl de vidas

## Pantallas, Mundo o escenario

El juego “ToursNica” cuenta con cuatro pantalla con diferentes funcionalidades”

## Menú Principal



Figura #28 Pantalla Principal

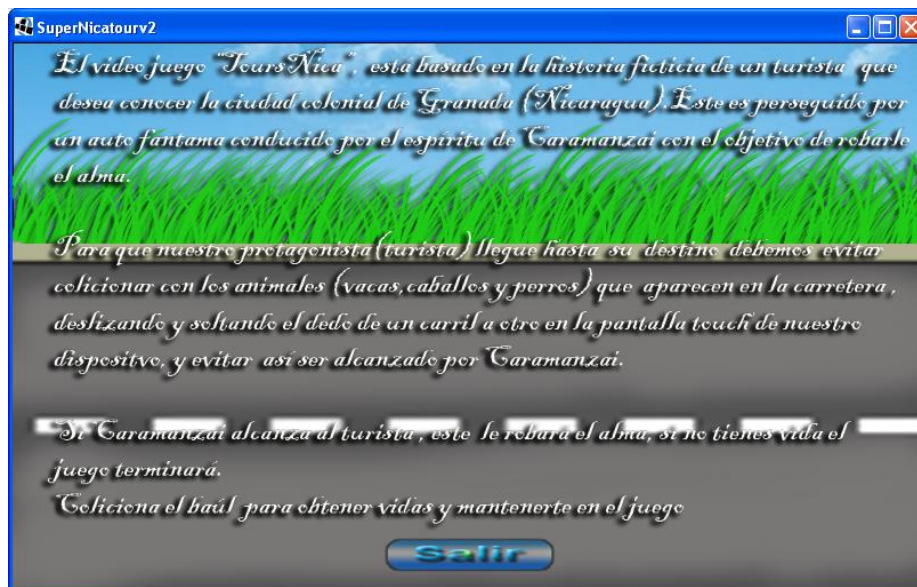


Figura #29 Pantalla ayuda

Pantalla de ayuda donde el jugador comprenderá más sobre la historia y uso del videojuego.



Esta es la pantalla inicial del juego, consta de dos botones táctiles, que al manipular con el dedo sobre el botón jugar nos permite entrar al escenario del juego e iniciar la partida

Y al manipular con el dedo sobre el botón salir, nos permite salir de la aplicación.

### Mundo o escenario

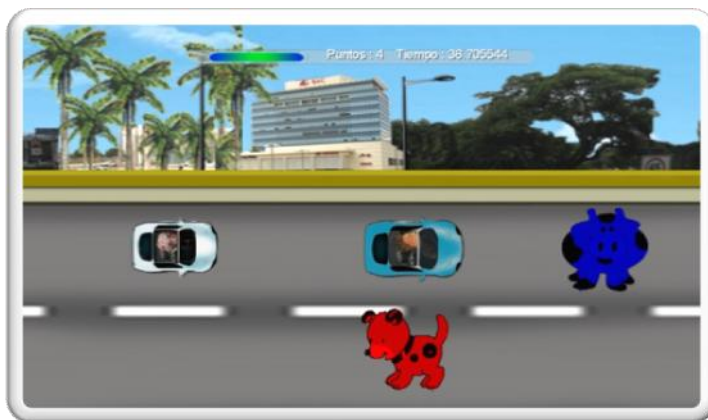


Figura #30 Escenario del videojuego

Esta pantalla es el mundo o escenario del juego donde interactúan todos los actores del videojuego.

### GameOver



Figura #31 Pantalla Game over

Esta pantalla aparecerá cada vez que el jugador haya perdido la partida, al pulsar con el dedo sobre Game Over nos trasladará a la pantalla del menú principal por si desea continuar jugando.

Pantalla fin de partida



Figura #32 Pantalla fin de Partida

Esta pantalla aparecerá cuando el jugador haya finalizado con éxito la partida.

### **Objetivos y misiones**

- 1- Hacer que el protagonista llegue a su destino y ganar la partida
- 2- Evitar ser alcanzado por el carro enemigo
- 3- Acumular las vidas necesarias para evitar perder la partida
- 4- Evitar colisionar con los obstáculos para sumar el mayor puntaje.

### **Mecánica del videojuego**

1. Interacción del usuario con el dedo al mover el carro protagonista del carril 1 al carril 2,
2. Avance del carro enemigo sobre el eje de las x y movimientos de carril según el carro protagonista.
3. Efectos de sonidos y rotación de los obstáculos al colisionar con el carro protagonista.
4. Incrementos y decrementos de la barra que controla las vidas del video juego.
5. Incrementos y decrementos de los puntajes del jugador
6. Visualización del tiempo transcurrido en el escenario.
7. Visualización de la pantalla Game Over cuando se ha perdido la partida
8. Visualización de la pantalla de felicitaciones cuando se ha ganado la partida.

## **Detalles técnico**

Para desarrollar este proyecto (Videojuego) nos hemos valido, de herramientas de programación, diseño e edición , las cuales detallamos a bajo:

### **Entorno de programación:**

Instalación del ADT (Android Development Tools )de android , es un plugin para el IDE Eclipse que está diseñado para darle un ambiente potente e integrado, actualizado hasta la versión 22.0.5

Actualización del SDK de android en eclipse que nos proporciona las bibliotecas API y las herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android, actualizado hasta la versión 22.0.5

Configuración del AVD(Android Virtual Machine) interfaz gráfica de usuario en la que se pueden crear y administrar dispositivos Android Virtual (AVD), que son necesarios por el [emulador de Android](#) .

Instalación del JDK (se Development Kit)de java 7 update25, necesario para compilar y ejecutar el proyecto java.

Utilización del lenguaje de programación java para el desarrollo del proyecto

### **Herramienta de diseño**

Para diseñar cada una de nuestras imágenes se hizo uso de fotoshop CS5.

### **Herramientas de edición**

Para editar los sonidos y la música se hizo uso de pennacle estudio 12.

**Caso de uso, diagrama de clase**

A continuación en la siguiente figura #33 se encuentra nuestro diagrama de caso de uso.

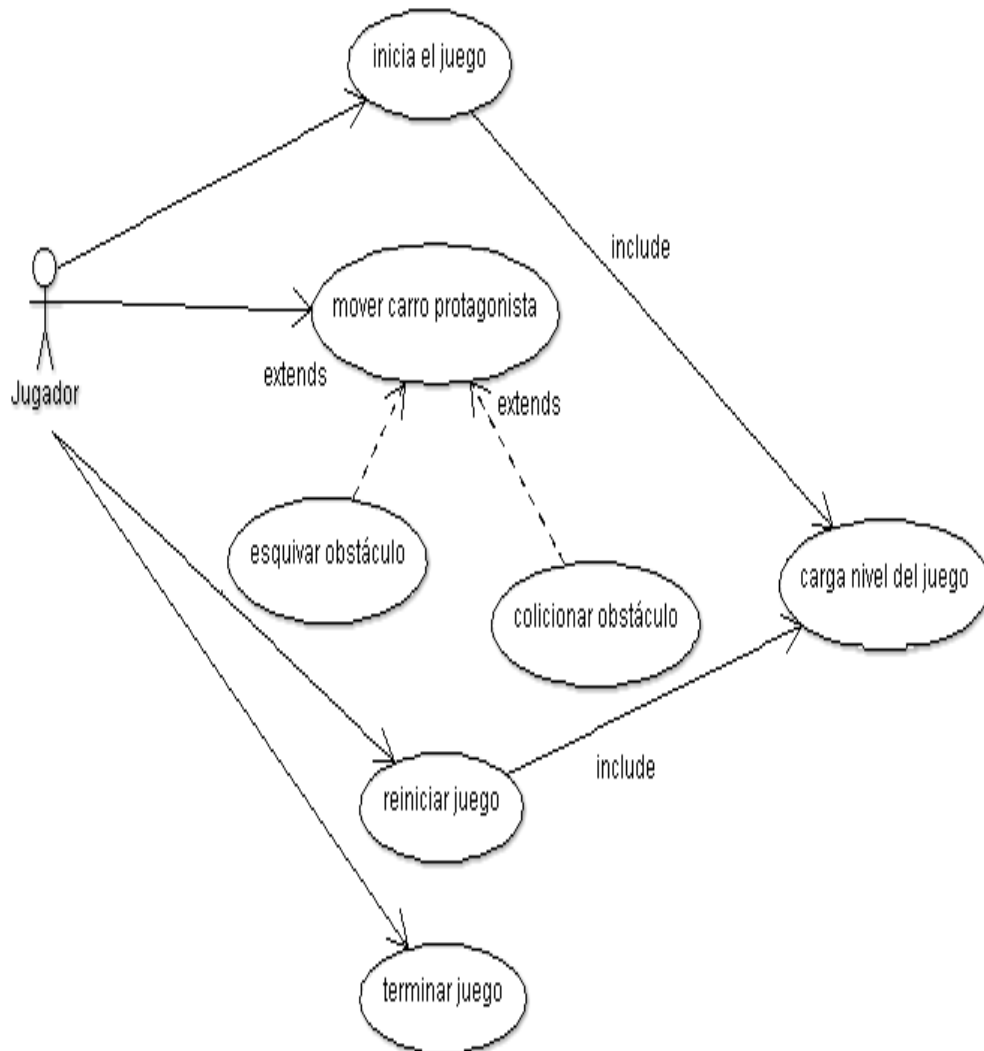


Figura #33 caso de uso

**Proceso:** inicia el juego

**Breve descripción:** El usuario selecciona la opción del menú *Jugar*. **Actor principal:** Usuario

1. El usuario selecciona la opción *Jugar* en el menú principal.
2. La aplicación carga el primer nivel del videojuego.
3. La aplicación muestra el fondo de pantalla, el indicador de nivel , y prepara los buffer para renderizar las escenas.

**Proceso:** mover Carro Protagonista

**Breve descripción:** El usuario selecciona la opción del carro Actor **Actor principal:** Usuario

1. El usuario desliza el dedo (touch ) Arriba hacia abajo.
2. El carro protagonista esquivo los obstáculos.
- 3- El carro protagonista colisiona los obstáculos.

**Proceso:** reiniciar juego

**Breve descripción:** El usuario selecciona la opción del menú. **Actor principal:** Usuario

- 1- Cargar de menú
1. El usuario selecciona la opción *Jugar* en el menú principal.
2. La aplicación carga el primer nivel del videojuego.
3. La aplicación muestra el fondo de pantalla, el indicador de nivel , y prepara los buffer para renderizar las escenas.

**Proceso:** Terminar Juego

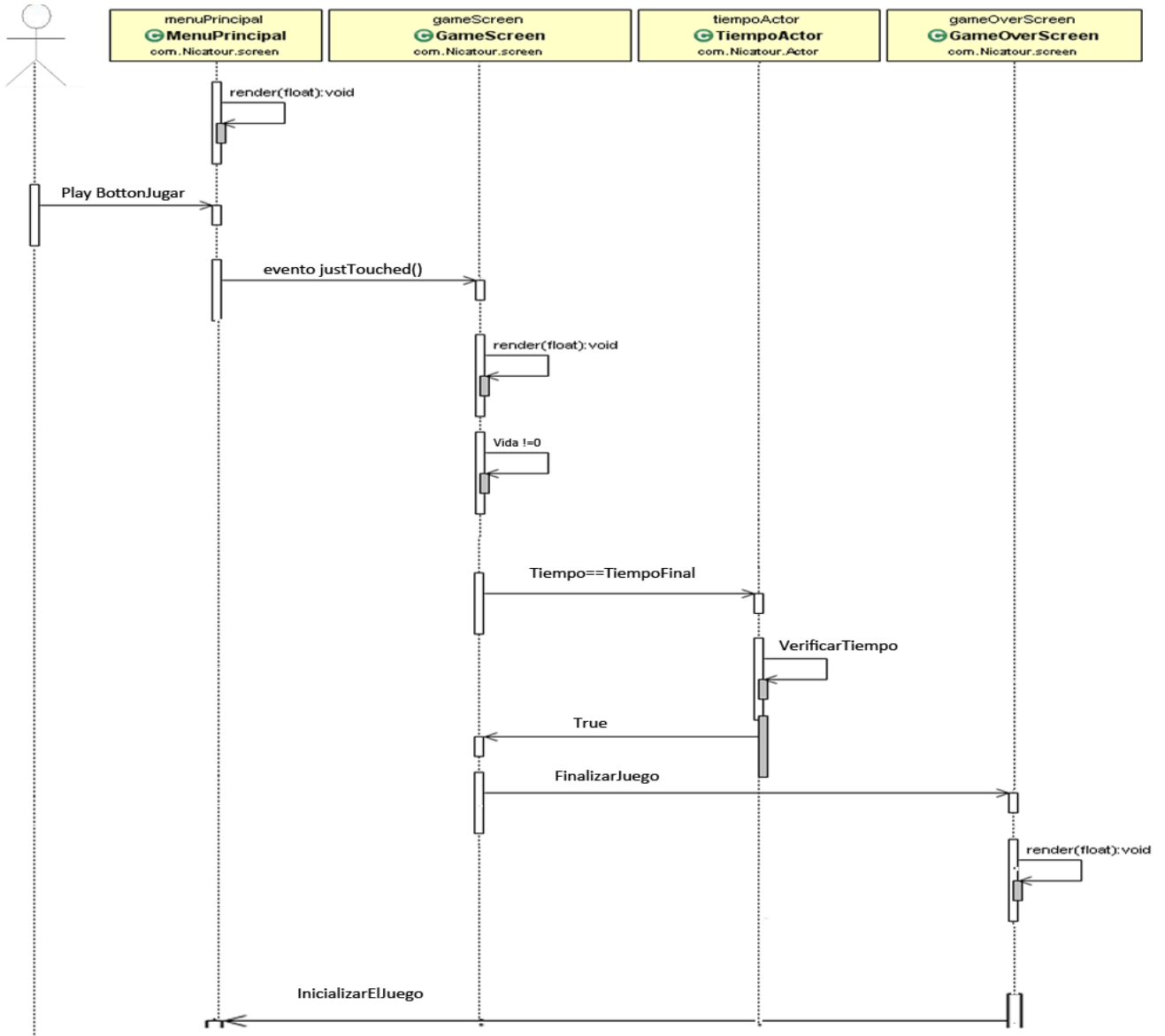
**Breve descripción:** El usuario selecciona la opción del menú. **Actor principal:** Usuario

- 1- Cargar de menú
1. El usuario selecciona la opción *salir* en el menú principal.
2. La aplicación se cierra.

.

**Diagrama de Secuencia de Iniciar Juego**

A continuación (*Figura #34*) se muestra la secuencia de ejecución al pulsar el botón de iniciar juego. El ejemplo mostrado es el paso del menú principal al menú jugar mediante el botón *Jugar*. *Figura #34* Diagrama de Secuencia



### Diagrama de clase

En la *Figura #35* podemos ver el diagrama de clases general de la aplicación. Es una versión simplificada para entender la estructura básica

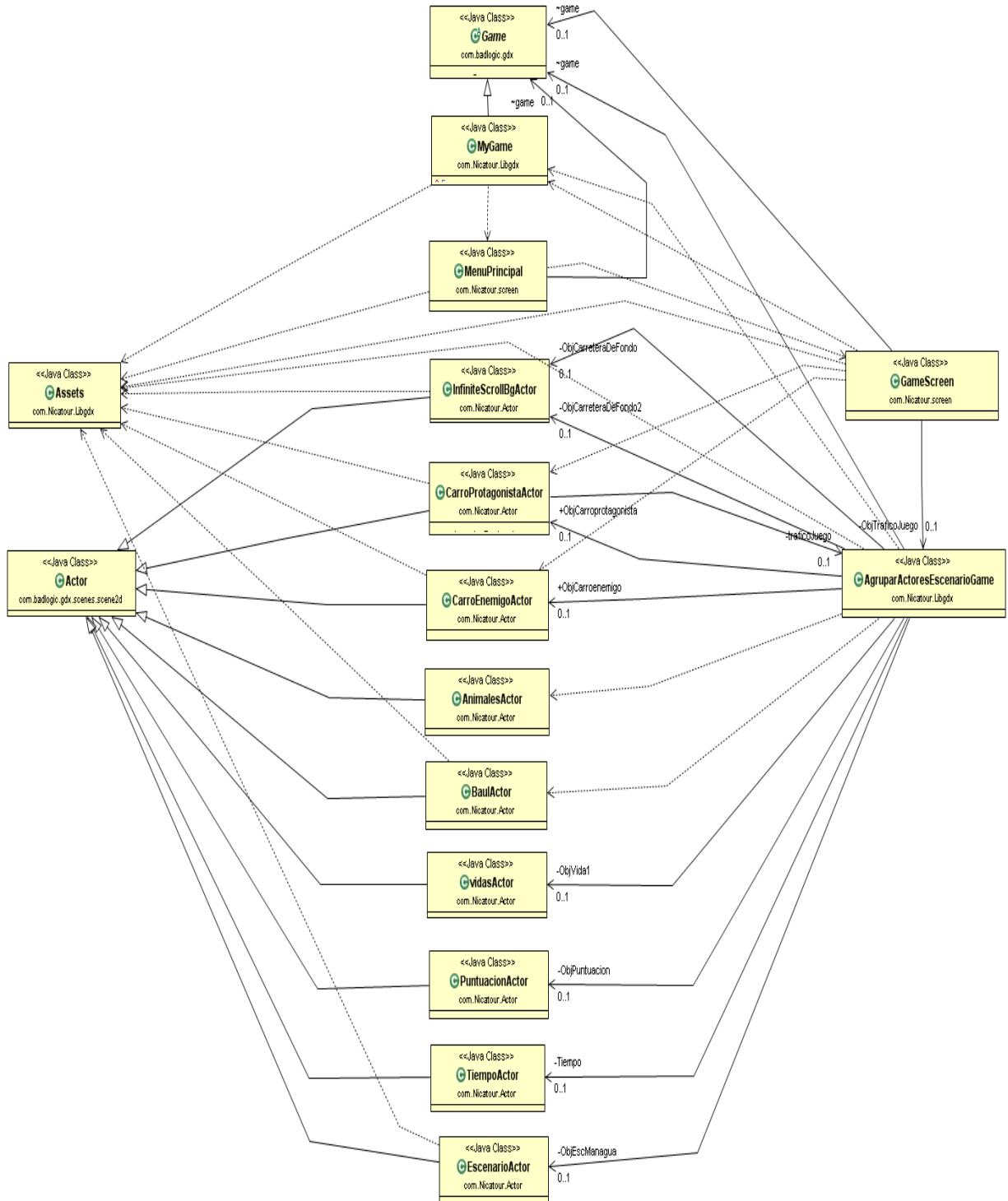


Figura #35 Diagrama de clase

A continuación damos una breve explicación de las clases más importante de nuestro video juego.

### Clase Assets()

Es una clase estática encargada de cargar todo los recursos que utilizaremos en memoria, tales como las imágenes, sonido y la música.

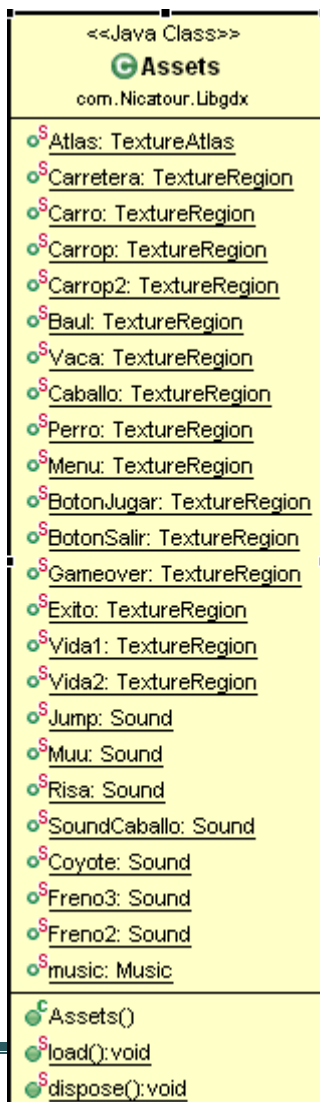
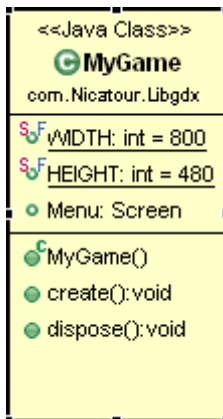




Figura #36 Clase Assets

Clase MyGame()



Clase que inicializa el videojuego con su debida resolución, carga los recursos que serán utilizados a través del método load() de la clase statica Assets() y manda a inicializar el menú principal.

Figura #37 Clase Mygame

Clase MenuPrincipal()

Implementa la clase abstracta Screen y es la encargada de renderizar la pantalla principal del videojuego .

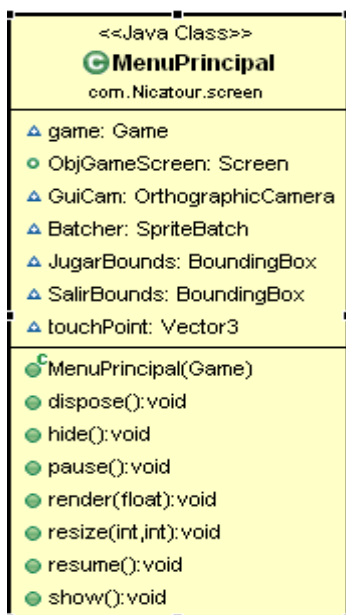
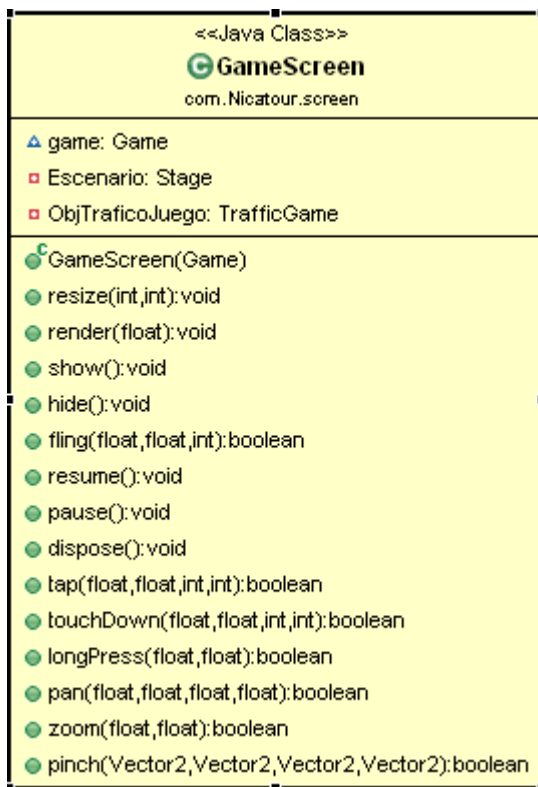


Figura #38 Clase menuprincipal

**Clase GameScreen()**



Clase encargada de inicializar en un objeto stage() el objeto AgruparActoresEscenarioGame(), que a su vez inicializa cada uno de los actores del video juego.

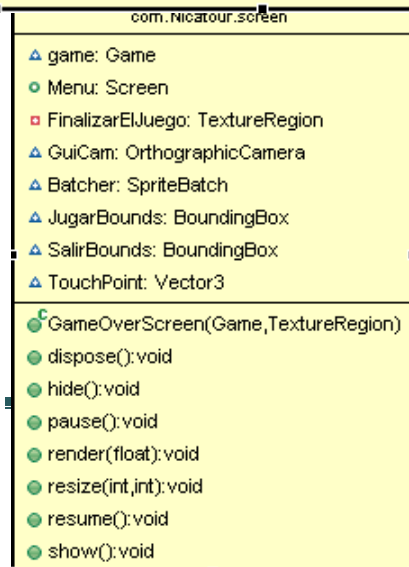


Figura #39 clase Gamescreen

**Clase GameOverScreen()**

Implementa la clase abstracta Screen encargada de rendirizar la pantalla del Game Over cuando se ha

perdido la partida o la pantalla de éxito cuando se ha ganado la partida.

Figura #40 Clase Gameoverscreen

### Clase InfiniteScrollBgActor()

Clase encargada de implementar el efecto Scroll de forma horizontal en el escenario simulando la velocidad en la carretera, haciendo uso de los métodos statico addAction(),forever(),sequence() y moveTo de la clase statica Action librería de escene2d de libGDX.

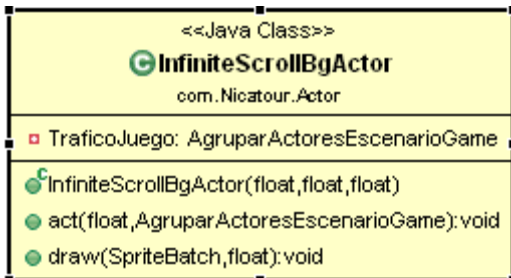


Figura #41 Clase InfiniteScrollBgActor

### Clase CarroProtagonista()

Clase encargada de definir las propiedades de nuestro carro protagonista, tales como pintar, mover y actualizar en el escenario, los métodos utilizados para realizar el movimiento de desplazamiento en el eje de las y son :tryMoveUP() y tryMoveDown.

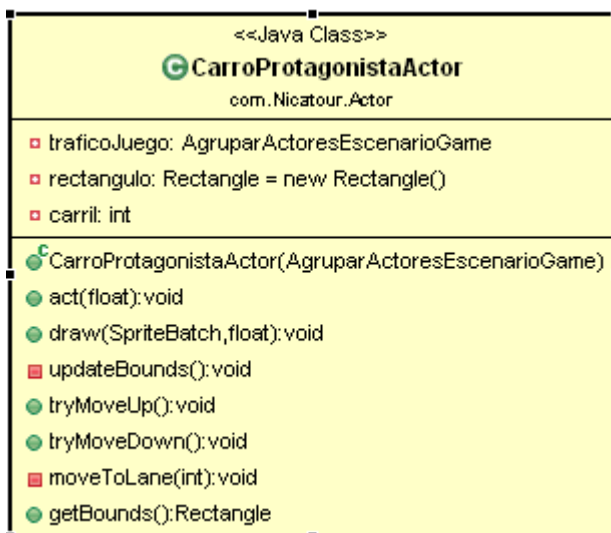


Figura #42 Clase carroProtagonista

### Clase AnimalesActor()

Clase encargada de generar los obstáculos de forma aleatoria, tanto en el carril1 como en el carril2 de forma indefinida.

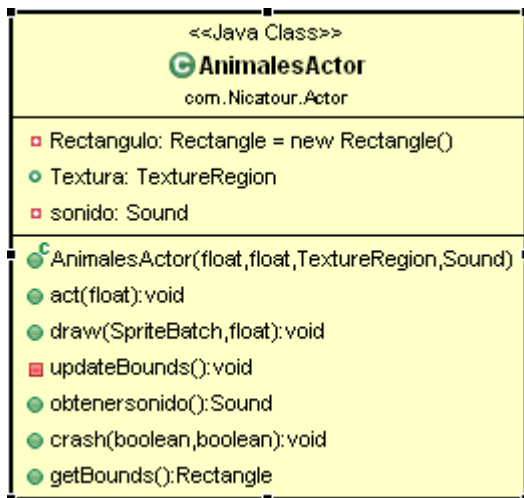


Figura #43 clase AnimalesActor

### Clase VidasActor()

Clase encargada de controlar, pintar y actualizar la barra de vida en la parte superior del escenario.

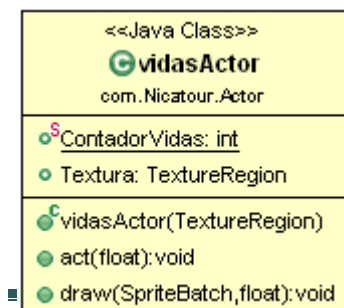


Figura #44 Clase VidasActor

**Clase PuntuacionesActor()**

Esta clase hace uso de un objeto BitmapFont() para mostrar en el escenario el sistema de puntos del juego.

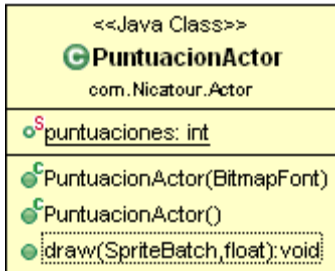


Figura #45 PuntuacionesActor

**Clase EscenarioActor()**

Clase encargada de simular que el video juego Tours Nica sucede en el trayecto de Metrocentro a Granada mostrando vistas panorámica del recorrido.

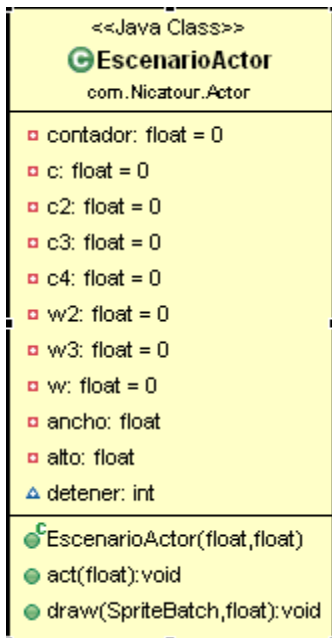


Figura #46 Clase EscenarioActor

## Clase AgruparActoresEscenarioGame()

Como su nombre lo indica esta clase es la responsable de inicializar cada uno de los objetos actores del videojuego.

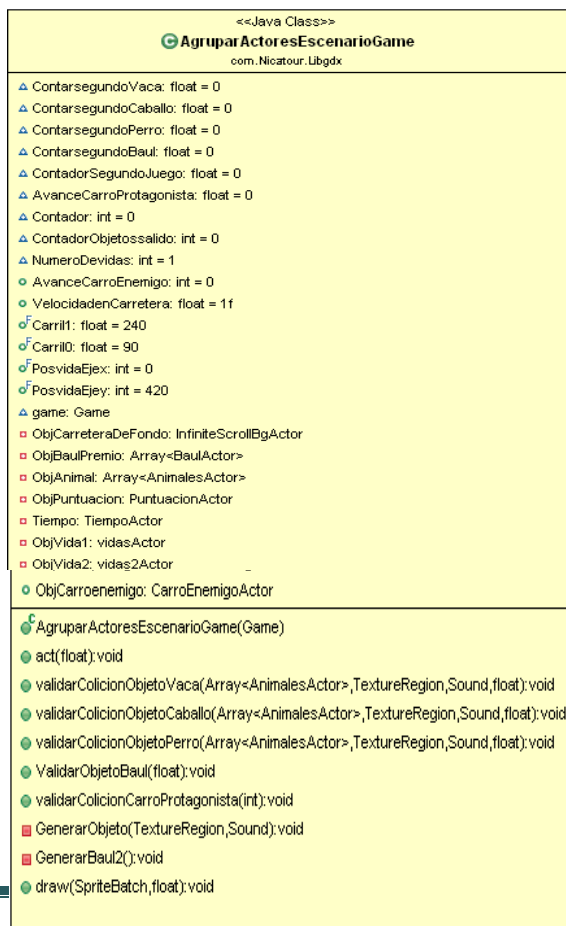


Figura #47 Clase AgruparActoresEscenarioGame

### **CONCLUSIONES**

- Una vez realizado nuestro proyecto, hemos concluido, que todo video juego debe tener asociado una historia que sirva de guión para su desarrollo y éxito del mismo.
- El desarrollo de un buen documento de diseño, es esencial para la comprensión de la mecánica del Video.
- El paradigma orientada a objeto (POO) , nos da ventajas , en el desarrollo del videojuego, en cuanto a Comprensión, reusabilidad, mantenimiento rendimiento, tiempo en el desarrollo y una mayor interpretación de lo que pasa de manera sistemática e interna en la programación y ejecución del proyecto

### **RECOMENDACIONES**

- ✓ El guión del videojuego Tours Nica puede ampliarse en niveles y mundo para crear una nueva versión.
- ✓ Nuestro documento de diseño puede estar sujeto a cambios con respecto a la modificación del guión.
- ✓ El video juego Tours Nica puede expandirse a diferentes niveles o escenarios apoyándose de las clases previas desarrolladas.



### **BLOGRAFÍA**

- [1] García,A(2008), Programación de videojuegos con SDL para Windows y Linux, <http://www.etnassoft.com/biblioteca/programacion-de-videojuegos-con-sdl-para-windows-y-linux/>.
- [2] Gerardo Abraham Morales Urrutia, Claudia Esther Nava López, Luis Felipe Fernández Martínez, y Mirsha. (2007), Procesos para el Desarrollo de Videojuegos Instituto de Ingeniería y Tecnología. Universidad Autónoma de Ciudad Juárez <http://www2.uacj.mx/IIT/CULCYT/enero-abril2010/7%20Art%204.pdf>
- [3] Programación para dispositivos móviles. <http://wwwdi.ujaen.es/asignaturas/progav/progav-tema6.pdf>
- [4] Caffa, A (2008) Sistemas operativos para dispositivos móviles. <http://miriammeza.files.wordpress.com/2010/09/presentacion.pdf>
- [5] Escobedo, j (2008),México, Comunicación Móvil. <http://www.monografias.com/trabajos63/comunicacion-movil/comunicacion-movil.shtml>
- [6 ] Garcia,D (2012),Desarrollo de videojuegos sobre la plataforma. <http://upcommons.upc.edu/pfc/bitstream/2099.1/14016/1/76880.pdf>
- [7] Aranaz D, (2009), UNIVERSIDAD CARLOS III DE MADRID ESCUELA POLITÉCNICA SUPERIOR, Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma android de Google.
- [8] Ernesto Antonia Tejero; Jose María Rodríguez,(2011), Sevilla Video Juego para Android
- [9]Gómez, S,(2011),Manual de programación Android v2 pdf <http://www.slideshare.net/dfartigle/manual-programacin-android-sgolivernet-v20>
- [10]González J. Seoane J Robles, G (s.f), Introducción al software libre. <http://curso-sobre.berlios.de/introsobre/1.0/libre.pdf>

[11] Bemoob,(s.f),curso de programacion en android para principiante.

<http://es.scribd.com/doc/139439180/Curso-de-programacion-basico-de-Android-FAQSAndroid>

[12]<http://jugabilidad.wikispaces.com/Producci%C3%B3n+y+Desarrollo+de+Videojuegos>  
Os

[13] <http://es.wikipedia.org/wiki/Android>

[15] <http://libgdx.badlogicgames.com/>

[16] Origen y evolucion de las Tablet

<http://tabletsnuevageneracion.blogspot.com/>