

**DESARROLLO DE UN JUEGO DE LABERINTO EN JAVA J2ME PARA EL
SISTEMA OPERATIVO ANDROID 2.2**

**ANA MARÍA MONTOYA BURITICÁ
WILSON HOLBEIN GUTIÉRREZ FRANCO**



**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
INGENIERÍA EN ELECTRÓNICA
SANTIAGO DE CALI
2011**

**DESARROLLO DE UN JUEGO DE LABERINTO EN JAVA J2ME PARA EL
SISTEMA OPERATIVO ANDROID 2.2**

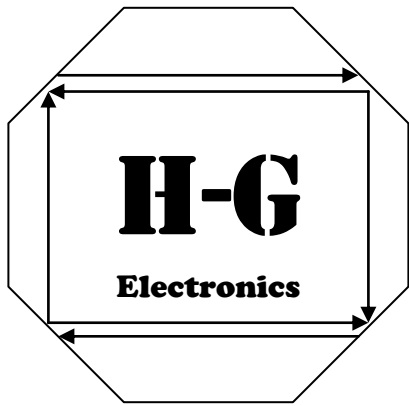
**ANA MARÍA MONTOYA BURITICÁ
WILSON HOLBEIN GUTIÉRREZ FRANCO**

Curso de Profundización en Computación Móvil

**Director: Ing. Gabriel Mauricio Ramírez
Tutor: UNAD**



**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
INGENIERÍA EN ELECTRÓNICA
SANTIAGO DE CALI
2011**



**HOLBEIN-GUTIERREZ
SOLUCIONES PROFESIONALES**

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Santiago de Cali, Mayo 23 de 2011

DEDICATORIA

A mi madre, **AMALIA FRANCO** y a mi padre, **LEONEL GUTIERREZ** por haberme formado con grandes valores y darme su apoyo incondicional, los cuales hacen que me sienta orgulloso de lo que hoy soy. A pesar de los tropiezos que tuve nunca me rendí, para llegar así a las metas que siempre me prometí.

A todos mis compañeros, en especial a los que compartieron conmigo algo más que un salón de clases.

Y a DIOS ya que sin su ayuda no hubiera podido alcanzar las metas que me he propuesto.

Wilson Holbein Gutiérrez Franco.

AGRADECIMIENTOS

Parecía que ya había acabado el trabajo duro, pero todavía quedaba algo muy complicado. No se trata de detenerse unos instantes para pensar los nombres de las personas que han hecho posible que ahora me encuentre escribiendo estas palabras, eso sería difícil, lo fácil es intentar no olvidar a nadie, por que la ayuda de todos ha sido fundamental para alcanzar el objetivo deseado.

Agradezco a los tutores de la Universidad Nacional Abierta y A Distancia por los conocimientos compartidos y enseñados para mi desarrollo profesional, en especial a la Dra. Rocío Pabon por haber confiado en mi persona, por la paciencia y por el apoyo incondicional en el inicio como estudiante de la Unad Cead Acacias, A la Ingeniera Andrea Barrero por apoyarme y acogerme en el Cead de Yopal como uno más de sus aprendices.

Gracias también a mis queridos compañeros, que me apoyaron y permitieron entrar en su vida durante estos tres años de convivir dentro y fuera del salón de clases, en especial a la compañera Marly Bernal por los consejos, el apoyo y el ánimo que me brindo, al compañero Edgar Pedroza, Nelson Garzón, Ana María Montoya, a mis compañeros de trabajo y universidad ya que entre todos nos ayudábamos, Jhonny Giraldo, Martin Rodríguez, Raúl Romero, Andrés Vargas y en especial al compañero Álvaro Gómez que Dios lo tenga en su gloria.

Gracias a todos.

GLOSARIO

COMPUTACIÓN MÓVIL: La computación móvil es un paradigma computacional que tiene como objetivo proporcionar a los usuarios acceso permanente a una red fija o móvil independientemente de su posición física. Es la capacidad de acceder a la información, aplicaciones y servicios desde cualquier lugar y en cualquier momento.

JAVA: Es un lenguaje de programación orientado a objetos, lo cual significa que toda la funcionalidad que se desee desarrollar debe estar enmarcada en clases y objetos. Java fue creado en la década de los 90 por la empresa Sun Microsystems y su característica principal es que es multiplataforma, es decir, Java es un lenguaje de programación que permite ejecutar aplicaciones que se desarrollan en varias plataformas, sistemas operativos y equipos sin necesidad de compilar o realizar cambios en el código fuente.

J2ME: J2ME (Java 2 Micro Edition) es una versión de Java diseñada para programación en equipos celulares. J2ME posee ciertas limitaciones en cuanto a clases disponibles, tipos de datos y funciones que se pueden utilizar. Estas limitaciones se deben a que los equipos celulares poseen poco poder de procesamiento y de memoria. A pesar de esto, las características de J2ME son las mismas que Java: Programación Orientada a Objetos, Herencia, Encapsulamiento, Portabilidad y Compatibilidad entre plataformas. J2ME permite que los teléfonos corran aplicaciones pequeñas e instalables por el usuario.

MEMORIA: La memoria del equipo es el componente que permite almacenar información, desde datos simples como los contactos telefónicos y mensajes de texto hasta aplicaciones, juegos, bases de datos, imágenes y videos. Generalmente estos equipos utilizan memoria del tipo ROM o Flash ROM que tienen la capacidad de mantener la información almacenada, aunque el equipo se encuentre apagado o sin batería. Con el tiempo y con el uso de aplicaciones móviles más avanzadas, los usuarios comenzaron a solicitar mayor capacidad y por ello es frecuente ver equipos con varias decenas de MB disponibles y con posibilidad de incorporar memorias y tarjetas externas.

MEMORIA DE APLICACIONES: El teléfono celular generalmente tiene destinado cierto porcentaje de su memoria total al almacenamiento de las aplicaciones y juegos que, o bien vienen preinstalados en el equipo, o bien son instalados por el

usuario. Esta memoria es variable de acuerdo con el equipo, pero para equipos modernos puede llegar a 5 MB o más. Además de la memoria disponible para los ejecutables de una aplicación, se encuentra la memoria que el programa utilizará durante su ejecución, por lo que hay que tener en cuenta no sólo el tamaño requerido para la instalación del programa, sino también el tamaño que el software requiere libre para poder ejecutarse. Además del uso que haga la aplicación en cuestión, se debe tener en cuenta que siempre existen librerías del lenguaje y otras utilidades que se deben ejecutar en esta memoria para el correcto funcionamiento del software. Esta memoria es usualmente llamada *Heap Memory* y a veces, se comparte con la memoria de aplicaciones y a veces está virtualmente separada. Otro uso de la memoria por parte de las aplicaciones es un sector destinado a almacenar información para futuras ejecuciones. Esta memoria se denomina *Storage Memory*.

TABLA DE CONTENIDO

	pág.
INTRODUCCION	12
1. PROBLEMA DE INVESTIGACION Y JUSTIFICACION	13
2. OBJETIVOS	14
2.1 OBJETIVO GENERAL	14
2.2 OBJETIVOS ESPECÍFICOS	14
3. REVISIÓN DE LA LITERATURA	15
3.1 MARCO CONTEXTUAL	15
3.2 MARCO CONCEPTUAL	15
3.3 MARCO HISTÓRICO	16
3.4 MARCO LEGAL	16
4. METODOLOGÍA	20
4.1 METODOLOGÍA DE LA INVESTIGACIÓN	20
4.1.1 Tipo de estudio	20
4.1.2 Población	20
4.1.3 Muestra	20
4.1.4 Fuentes de obtención de datos	20
4.1.5 Pasos metodológicos	21
4.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE	21
5. ANALISIS, DISEÑO, IMPLEMENTACION Y PRUEBAS	23
5.1. OBJETIVO DE LA APLICACIÓN	23
5.1.1 Características del juego	23
5.1.2 Funcionamiento	24
5.1.2.1 Factores técnicos	24
5.1.2.2 Factores de ergonomía	24
5.2 REQUERIMIENTOS DE LA APLICACIÓN	26

	pág.	
5.2.1	Requerimientos técnicos	26
5.2.2	Requerimientos Funcionales.	26
6.	DISEÑO	28
6.1	CICLO DE VIDA DE UN MIDLET	28
6.2	INTERFACES DE USUARIO	28
6.3	LISTADO DE CASOS DE USO, ESPECIFICACIÓN Y DIAGRAMA	31
6.4	DIAGRAMA DE ESTADOS DEL JUEGO "DANGEROUS ROAD"	32
6.5	DIAGRAMA DE CLASES	33
7.	IMPLEMENTACIÓN	34
7.1	DEFINICIÓN DE LA TECNOLOGÍA DE LA APLICACIÓN	34
8.	RESULTADOS Y PROYECCIONES	35
8.1	RESULTADOS	35
8.2	PROYECCIONES	35
9.	CONCLUSIONES	36
	WEBGRAFIA	37

LISTA DE FIGURAS

	pág.
Figura 1. Diagrama de casos de uso	31
Figura 2. Diagrama de estados	32
Figura 3. Diagrama de clases	33

INTRODUCCIÓN

La programación y la creación o elaboración de aplicaciones para dispositivos móviles actualmente es muy importante, ya que cada día crece el número de usuarios de estos aparatos y con ello el número de exigencias y requerimientos, por esta razón se hace indispensable crear aplicaciones que hagan que el usuario se sienta cómodo o que lo entretengan para ser usadas en su dispositivo móvil, por esta razón hay una rama de la computación que se llama computación móvil la cual se puede definir como la serie de artefactos y equipos portátiles, hardware, que hacen uso de la computación para lograr su funcionamiento, así, se tiene a las computadoras portátiles, los teléfonos celulares, los cuadernos de notas computarizados, las calculadoras de bolsillo, etc.

La computación móvil ha dejado de ser un instrumento de presunción, para convertirse en un medio de trabajo. Pero eso no es todo: empieza transformarse en el medio de trabajo. Ya es parte del uniforme de los profesionales, del vendedor y hasta de los repartidores de refrescos que con él controlan el inventario de su camión.¹

El presente trabajo consta de un trabajo escrito en donde se muestra como se elaboro una aplicación donde se incluyen los requerimientos técnicos, el análisis y el diseño, la aplicación es un laberinto en J2ME desarrollado en eclipse para el sistema operativo Android 2.2.

¹ http://www.wikilearning.com/curso_gratis/computacion_movil_notebooks-computacion_movil/29102-2

1. PROBLEMA DE INVESTIGACION Y JUSTIFICACION

Los video juegos son aplicaciones de gran uso por las personas, ya que tienen la función de entretener y divertir al usuario, además de ayudar a mejorar el proceso de la creatividad, generación de estrategias e ideas, también ayuda a aumentar la capacidad de concentración, por ello se pensó en el desarrollo de un juego para un dispositivo móvil que soporte Java y tenga instalado el sistema operativo Android 2.2, el juego es un laberinto con diversas pantallas y varios niveles de dificultad, a los cuales se les adicionara un tiempo límite para encontrar la salida a cada laberinto, con el fin de lograr un juego más emocionante; un juego de este tipo ayuda a el usuario a pasar el tiempo y a entretenerse, fortaleciendo habilidades mentales en sus ratos de ocio o espera.

Muchos estudios han demostrado que los video-juegos pueden desarrollar la habilidad mental ya que el cerebro está activo y alerta cuando se juega con una aplicación de estas. Herman Moraldo un desarrollador de videojuegos e investigador sobre este tema desde una perspectiva teórica. Actualmente es uno de los directores y miembros fundadores de ADVA (Asociación de Desarrolladores de Videojuegos Argentina), y dueño de un emprendimiento dedicado al desarrollo de videojuegos para eventos y comunicación en general, el promueve el desarrollo de los videojuegos y la importancia de éstos como productos industriales, culturales, y objetos teóricos de importancia: “los juegos son sistemas cuyo funcionamiento está ligado íntimamente al de nuestra cognición. No podríamos entender nunca el efecto de los juegos si lo disociáramos de nuestro sistema cognitivo².

El desarrollo de este video juego el cual se llamara “DANGEROUS ROAD” permitirá a los usuarios divertirse a la vez que desarrolla y fortalece en el usuario destreza, habilidad y mayor concentración, el juego se va a desarrollar de tal manera que sea fácil de usar y entender, para que los niños también lo puedan usar, sobre todo si presentan alguna dificultad, como bajo nivel de concentración, memoria entre otras habilidades cognitivas. De lo anterior se puede deducir la pregunta, ¿como desarrollar un juego que a parte de que sea entretenido ayude a mejorar habilidades cognitivas y mentales sin dejar a un lado factores tan importantes como la diversión y la jugabilidad?

² <http://portal.educ.ar/noticias/entrevistas/hernan-moraldo-los-videojuegos-1.php>

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un juego el cual es un laberinto en java J2ME para un dispositivo móvil con el sistema operativo Android 2.2, el cual permita diversión, entretenimiento, jugabilidad y que ayude a mejorar , fortalecer y desarrollar habilidades cognitivas y mentales en los usuarios.

2.2 OBJETIVOS ESPECÍFICOS

- Elaborar un juego que divierta, sea fácil de usar, atractivo al usuario y que desarrolle habilidades, como la concentración, estrategia y la creatividad, entre otras.
- Desarrollar un juego en donde haya una buena relación entre la interfaz grafica y la jugabilidad.
- Aprender a utilizar el programa java J2ME y el IDE eclipse para la elaboración de un juego para un dispositivo móvil.
- Conocer como se fabrica un video juego para el sistema operativo android, las características que soporta este sistema operativo y su funcionamiento.
- Comprender las características de Java J2ME, sus librerías y sus lenguajes lógicos.
- Comprender que los dispositivos móviles en la actualidad son dispositivos de uso masivo y con gran aceptación en el público.

3. REVISIÓN DE LA LITERATURA

3.1 MARCO CONTEXTUAL

La programación en java cada vez crece su auge en Colombia, al saber que con menos esfuerzos se hacen programas hermosos, bueno es que java ha evolucionado, con la nueva plataforma JEE, la ayuda al programador es enorme. Realmente los programadores deben preocuparse por las instrucciones que hacen único al programa, pues digamos que ya todo viene listo para usarse, solo se debe “configurar”, bueno es que a ese ambiente de componentes es al que queremos llegar. Componentes listos para usar.³

Todo esto gracias a los señores de JCP (Java Community Process) donde participan empresas tan importantes como IBM, Oracle, Sun Microsystems, Novell Entre otros... todos menos Microsoft, Todo gratis, ya que es código libre.⁴

Seria genial que desde el colegio, los niños crezcan construyendo sus propios sitios web y haciendo pequeñas tareas de software. Ejercicios de Software o de lógica en el sitio web o con aplicaciones de escritorio o en móviles ó manejo de componentes electrónicos externos o robots, siempre buscando su motivación ante todo.⁵

3.2 MARCO CONCEPTUAL

Java es toda una tecnología orientada a la programación de software con el cual se puede 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

³ <http://www.facebook.com/topic.php?uid=4768357861&topic=2907>

⁴ <http://www.fayerwayer.com/2009/04/queda-poco-para-un-historico-eclipse-solar/>

⁵ <http://www.facebook.com/topic.php?uid=4768357861&topic=2907>

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.⁶

Java brinda una solución para la programación en todo tipo de plataformas; en especial la J2ME (Java orientado a Móviles, PDAs, entre otros).

3.3 MARCO HISTÓRICO

En los teléfonos móviles al principio se podían hacer una serie de cosas muy limitadas, que ya venían instaladas de fábrica. Aunque las funciones eran más o menos comunes, cada móvil era distinto, ya que se manejaba con el software propio de cada fabricante, que podía ser muy distinto al de otro. Más tarde los móviles nos permitieron instalar pequeñas aplicaciones, sobre todo juegos en Java. Ahí se comenzó a diferenciar entre software propio del móvil y lo que se podía instalar. Aunque como el sistema operativo del móvil aún dependía del fabricante, se debía de asegurar de que lo que se pretendía instalar iba a funcionar en el terminal.

Con la evolución de los terminales móviles y PDAs, se hizo necesario el poder instalar y desinstalar aplicaciones más complejas acorde a las necesidades de los usuarios. Y no tenía sentido tener que crear la aplicación para el sistema de cada fabricante. Por lo que aparecieron los verdaderos sistemas operativos, que funcionaban en móviles de distintos fabricantes.

Los primeros sistemas operativos para móviles fueron Symbian (de Panasonic, Siemens AG, Nokia, Sony-Ericsson entre otras), Palm (sobre todo para PDAs), BlackBerry y Windows Mobile (de Microsoft). Apple revoluciono el mundo de la telefonía móvil con el lanzamiento de su familia iPhone, con el sistema operativo iPhone OS, y su pantalla multitáctil.⁷

3.4 MARCO LEGAL

- El Proyecto De Ley Sobre Software Libre En Colombia

⁶ <http://todojava.awardspace.com/>

⁷ <http://www.buenastareas.com/ensayos/Ensayo-Sobre-Android/1569351.html>

Por una iniciativa parlamentaria, presentada a principios del año 2002, Colombia cuenta y discute en el seno del Congreso de la Republica, la posibilidad de adoptar a nivel positivo una legislación que regule y establezca de manera imperativa las políticas de uso y empleo del Software Libre en sus sistemas de información, al interior de las entidades e instituciones del Estado y las empresas donde el estado posea mayoría accionaría.

El proyecto consta de veintiuno artículos, a través de los cuales se busca regular y obtener un control sobre los sistemas de información evitando depender de proveedores únicos, promover la igualdad en el acceso a la información pública por parte de los ciudadanos, garantizar la transparencia de las tecnologías que utiliza el Estado para su funcionamiento salvo casos de seguridad nacional, evitar el acceso a la información por parte de terceros no autorizados según la constitución y la ley garantizando la seguridad nacional y la privacidad de los ciudadanos, y lograr la promoción y el desarrollo de una industria de software nacional.⁸

Aparte de incentivar el desarrollo tecnológico nacional e incrementar la seguridad nacional, existen importantes razones para este proyecto de ley, inspiradas en los principios y valores por los que debe luchar el Estado.

El artículo 15 de nuestra Constitución Política afirma el derecho de todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos y en archivos de entidades públicas y privadas.

El artículo 74 garantiza que todas las personas tienen derecho a acceder a los documentos públicos salvo los casos que establezca la ley.

El Estado debe garantizar el buen manejo de la información de los ciudadanos. Aquellos datos cuya confidencialidad es establecida por la ley deben ser tratados de manera tal que el acceso a ellos se dé únicamente a las personas e instituciones autorizadas.

En aras de la transparencia, el Estado debe permitirles a los ciudadanos comprender los sistemas de información que utiliza, en tanto que juegan un papel gran importancia en su funcionamiento. Sería muy saludable que a la ciudadanía

⁸ http://www.cecolda.org.co/index.php?option=com_content&task=view&id=17&Itemid=40

se le permitiera analizar y conocer los programas que se utilizan para manipular su información, calcular las tarifas de servicios públicos o los impuestos.

El software utilizado por el Estado debe tratarse de la misma manera que la demás información pública, en tanto que de él dependen procedimientos llevados a cabo en su interior. Por esto, debería permitirse su estudio y análisis por parte de los ciudadanos (lo cual no es posible al utilizar software propietario).⁹

El Software Libre es un tipo particular de software que le permite al usuario el ejercicio de cuatro libertades básicas:

Explicación de las libertades básicas del Software Libre

- **Libertad Cero:** "usar el programa con cualquier propósito". Es decir, el ejercicio de esta libertad implica que lo podemos utilizar con cualquier fin, ya sea educativo, cultural, comercial, político, social, etc. Esta libertad deriva de que hay ciertas licencias que restringen el uso del software a un determinado propósito, o que prohíben su uso para determinadas actividades.
- **Libertad Uno:** "Estudiar como funciona el programa, y adaptarlo a sus necesidades". Significa que podemos estudiar su funcionamiento (al tener acceso al código fuente) lo que nos va a permitir, entre otras cosas: descubrir funciones ocultas, averiguar como realiza determinada tarea, descubrir que otras posibilidades tiene, que es lo que le falta para hacer algo, etc. El adaptar el programa a mis necesidades implica que puedo suprimirle partes que no me interesan, agregarle partes que considero importantes, copiarle una parte que realiza una tarea y adicionarla a otro programa, etc.
- **Libertad Dos:** "Distribuir copias". Quiere decir que soy libre de redistribuir el programa, ya sea gratis o con algún costo, ya sea por email, FTP o en CD, ya sea a una persona o a varias, ya sea a un vecino o a una persona que vive en otro país, etc.
- **Libertad Tres:** "Mejorar el programa, y liberar las mejoras al público". Tengo la libertad de hacer mejor el programa, o sea que puedo: hacer menores los requerimientos de hardware para funcionar, que tenga mayores prestaciones,

⁹ <http://sheilafierro.blogspot.com/2007/09/ley-de-software-libre-en-colombia.html>

que ocupe menos espacio, que tenga menos errores, etc. El poder liberar las mejoras al público quiere decir que si yo le realizo una mejora que permita un requerimiento menor de hardware, o que haga que ocupe menos espacio, soy libre de poder redistribuir ese programa mejorado, o simplemente proponer la mejora en un lugar público (un foro de noticias, una lista de correo, un sitio Web, un FTP, un canal de Chat).¹⁰

Con la única restricción del copyleft (o sea, cualquiera que redistribuya el software, con o sin cambios, debe dar las mismas libertades que antes), y con el requisito de permitir el acceso al código fuente (imprescindible para ejercer las libertades 1 y 3).

Es importante resaltar que el software libre no atenta de ninguna manera contra los derechos de autor y de propiedad intelectual: no tiene nada que ver con la piratería, en tanto que los autores autorizan explícitamente a los demás a hacer uso de sus creaciones ofreciéndoles las libertades anteriores.

¹⁰ <http://www.monografias.com/trabajos51/software-libre/software-libre2.shtml>

4. METODOLOGÍA

4.1 METODOLOGÍA DE LA INVESTIGACIÓN

4.1.1 Tipo de estudio

Los estudios descriptivos buscan desarrollar una imagen o fiel representación (descripción) del fenómeno estudiado a partir de sus características. Describir en este caso es sinónimo de medir. Miden variables o conceptos con el fin de especificar las propiedades importantes de comunidades, personas, grupos o fenómeno bajo análisis.¹¹

Teniendo en cuenta la finalidad del estudio del desarrollo de software aplicativo a la telefonía móvil, se puede identificar dicho proyecto en un estudio de investigación de tipo descriptivo.

4.1.2 Población

La presente investigación tiene por objeto analizar a la comunidad, para realizar una distribución comercial de juegos para dispositivos móviles de gran consumo que operan en todo el mundo.

4.1.3 Muestra

Se utilizó una muestra no probabilística y de conveniencia, debido a que no se utilizan procedimientos de selección por casualidad, en la presente investigación se seleccionaron individuos ya conocidos, enfocados en la tecnología de dispositivos móviles.

4.1.4 Fuentes de obtención de datos.

Los datos necesarios para llevar a cabo esta investigación, se han obtenido con fuentes de información secundarias: como son consultas en internet sobre la popularidad de los juegos laberinto, documentos para crear aplicaciones J2ME, libros y videos de programación en Android y desarrollo de software.

¹¹ <http://tgrajales.net/investipos.pdf>

4.1.5 Pasos metodológicos

La monografía y la aplicación se realizarán siguiendo los siguientes pasos metodológicos:

- **Investigación.** Se hará una búsqueda exhaustiva sobre los conceptos, el uso y la aplicación de la plataforma Java y los IDE como Eclipse. Se revisarán trabajos anteriores sobre el tema de desarrollo de aplicaciones y juegos en Java J2ME con el fin de tomarlos como antecedentes a la monografía.
- **Elaboración de la monografía.** Se desarrollará la monografía con base a lo investigado y lo realizado en la aplicación, esto de acuerdo con lo exigido en la guía de presentación.
 - Etapa 1. Consta de 4 pasos, selección del tema, definición del problema, marcos de referencia y definición de la metodología.
 - Etapa 2 consta de 3 pasos, análisis, diseño e implementación
 - Etapa 3 consta de 3 pasos, resultado y proyecciones, conclusiones y bibliografía.
- **Terminación y Sustentación de la monografía.** Se presentará la monografía total tal cual se trabajó durante el semestre junto con la aplicación que se desarrolló conjuntamente a esta. Se sustentará ante el director del curso el producto final

4.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE

La elaboración del diseño de software laberinto en J2ME y la implementación del mismo está basada en:

- Investigación del sistema operativo Android y sus diversas plataformas.
- Investigación y metodología de programación en J2ME.
- Investigación en el desarrollo del software laberinto sobre plataforma Android.
- Implementar el software en la plataforma Android.
- Verificar los resultados.
- Realizar correcciones.
- Implementar mejoras.

Para lograr la elaboración del juego se harán las siguientes actividades.

- Investigación sobre java J2ME sus características
- Investigación sobre el sistema operativo Android 2.2
- Realización del código para la aplicación
- Pruebas de funcionamiento de la aplicación
- Entrega de la monografía
- Muestra de la aplicación
- Sustentación del aplicativo.

El desarrollo del software laberinto y la implementación en la plataforma Android, está basado en los conocimientos metodológicos, teóricos y prácticos adquiridos en el transcurso de nuestra carrera, complementado con el curso de computación móvil.

Previamente el desarrollo del software laberinto se está elaborando bajo la asesoría del Ingeniero Gabriel Mauricio Ramírez Villegas docente de la universidad Nacional Abierta y a Distancia Cead (Palmira) y donde será comprobado su funcionamiento en los laboratorios de Informática para su respectiva sustentación.

5. ANALISIS, DISEÑO, IMPLEMENTACION Y PRUEBAS

5.1 OBJETIVO DE LA APLICACIÓN

El objetivo de la aplicación es lograr un juego divertido, que sea fácil de jugar y entender y que funcione en un móvil con sistema operativo Android 2.2 que soporte java y además que ayude a mejorar la concentración y la creatividad.

5.1.1 Características del juego

- Una interfaz grafica que sea atractiva al usuario además de funcional.
- Un juego que no ocupe mucho espacio en la memoria del dispositivo móvil donde se instale.
- El juego debe lograr divertir al usuario, a la vez que ayude a mejorar algunos procesos cognitivos.

5.1.2 Funcionamiento

Para el funcionamiento de la aplicación en este caso un juego de laberinto, es necesario que el móvil soporte java J2ME.

Las teclas del celular servirán para desplazar el objeto por los diferentes posibles caminos del laberinto, así como para pausar y comenzar una nueva partida.

- **Espectacularidad:** el videojuego ha de tener un buen diseño gráfico y una música adecuada al tipo de juego, que sepa acompañar correctamente la acción de este.

No basta sólo con esto, sino que además el videojuego deberá tener cierta calidad técnica y aprovechar las características que el hardware del dispositivo pone a disposición del programador; pero con cierta moderación, ya que se corre el peligro de sacrificar dosis de jugabilidad por querer centrarse demasiado en los alardes técnicos.¹²

¹² personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

- **Jugabilidad:** sin duda alguna, el objetivo de mayor prioridad. Nunca se debe disminuir la jugabilidad en pro de un juego más espectacular técnicamente. No hay una fórmula para obtener una buena jugabilidad, pero sí se considerarán algunos factores necesarios, aunque no imprescindibles: facilidad de manejo, dificultad ajustada y creciente, no repetitividad.

Estos dos objetivos no siempre son fáciles de conseguir, y menos en dispositivos móviles, debido a dos factores: características técnicas y ergonomía. A continuación se explicarán ambos y se enumerarán posibles criterios a seguir para diseñar un buen videojuego.

5.1.2.1 Factores técnicos. A la hora de diseñar un videojuego para un dispositivo móvil, no se debe olvidar que éstos tienen unas características técnicas limitadas y, por tanto, no es posible realizar videojuegos de gran espectacularidad técnica. Y no sólo eso, sino que las aplicaciones tampoco deben tener una gran carga de proceso, es decir, gran número de elementos a controlar a la vez (por ejemplo, muchos enemigos en pantalla), una inteligencia artificial compleja y costosa computacionalmente, o cálculos matemáticos intensos, ya que todas estas características consumen demasiados recursos de procesador y puede hacer que el juego sea demasiado lento.

Afortunadamente, estos factores tienen cada vez menos importancia, ya que la tecnología móvil avanza a pasos de gigante, y hoy en día empezamos a disfrutar de dispositivos móviles con varios megas de RAM, aceleración 3D, microprocesadores potentes, y otras características que hacen que se puedan realizar videojuegos que pocos años atrás estaban reservados a videoconsolas y ordenadores personales

5.1.2.2. Factores de ergonomía. Con los dispositivos móviles actuales, este es el factor a tener más en cuenta a la hora de diseñar un videojuego.¹³

- **Controles.** A medida que la tecnología avanza, no sólo se consiguen dispositivos más potentes, sino que cada vez son más pequeños. Aunque gracias a las nuevas características multimedia, las pantallas van adquiriendo tamaños y definiciones aceptables, esto se compensa con teclados pequeños, lo cual puede suponer un serio problema a la hora de controlar la acción del

¹³ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

juego, si ya de por sí es poco cómodo utilizar un teclado numérico para jugar, si éste además es de reducido tamaño puede dar como resultado una dificultad excesiva para manejar los controles y, como consecuencia, que el juego, por muy jugable que sea *a priori*, adquiera una dificultad excesiva que haga perder toda la jugabilidad que debiera tener.

En la mayoría de teléfonos, los controles se tendrán que realizar mediante el teclado numérico. Esto quiere decir que los botones para desplazar el personaje, así como los botones de acción, están en un pequeño espacio de 3x4 teclas. Lo ideal es que todo el control se pueda realizar mediante una sola mano, ya que es bastante incómodo tener dos manos en un teclado numérico de tamaño pequeño. La recomendación para este caso es que se utilicen las teclas 2, 4, 6, y 8 para controlar la dirección (más 1, 3, 7, 9 en caso que se quieran usar diagonales) y, para simplificar los controles, un sólo botón de acción, que será el centro (5).

En el juego, es recomendable que no sean frecuentes los momentos en que haya que moverse y pulsar el botón de acción a la vez (ejemplo: un mata-marcianos), ya que la acción implica que se deban soltar los botones de dirección.

En la actualidad, cada vez son más los teléfonos que vienen con un sencillo *controlpad*. Esto proporciona mucha comodidad a la hora de jugar, por lo cual facilita la tarea del diseño del juego. Ahora se puede asir el teléfono con las dos manos, sin que un dedo estorbe a otro, por lo que, mientras con una mano se mueve al personaje, con la otra se puede realizar la acción.

También se podrán implementar varias acciones, cada una con su botón. Para no complicar el manejo, se recomienda que como máximo sean 2 acciones de uso frecuente como por ejemplo, un botón para disparar y otro para saltar.

Para los *controlpads* actuales, hay que tener en cuenta que son de sólo 4 direcciones, con lo cual, si queremos usarlos habrá que suprimir el movimiento en diagonal del personaje.¹⁴

- **Pantalla.** Otro problema que, como ya se ha comentado, va mejorando con el tiempo, es el tamaño de la pantalla. Los teléfonos con mayores prestaciones en la actualidad suelen tener definiciones de unos 150x200 píxeles, aproximadamente. Esto causa que, en muchas ocasiones, el diseñador tenga que escoger entre espectacularidad gráfica o jugabilidad. Para determinados juegos, si se crean unos *sprites* (elementos móviles) demasiado grandes,

¹⁴ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

ocuparán demasiado espacio en pantalla, reduciendo así el campo de acción y la jugabilidad. Si, en cambio, se crean unos *sprites* demasiado pequeños, puede ser que el jugador tenga que hacer un esfuerzo visual demasiado grande, y que la jugabilidad se vea reducida por que los elementos no se identifican bien, o por el esfuerzo visual necesario.

Además, y como problema secundario, con tamaños de *sprite* pequeños se reduce el detalle gráfico de los elementos y, por tanto, la espectacularidad del juego.

La recomendación que se hace en este proyecto, es que los elementos del juego sean suficientemente grandes como para ser identificados fácilmente, pero que el campo de visión de la escena del juego sea suficientemente grande como para poder prever con antelación las situaciones del juego (por ejemplo: que podamos divisar si se acerca un enemigo y reaccionar a tiempo).

Si el juego que se esta diseñando no puede cumplir las recomendaciones aquí expuestas, quizás es que no es un tipo de juego adecuado para realizarse sobre un dispositivo móvil.¹⁵

5.2 REQUERIMIENTOS DE LA APLICACIÓN

5.2.1. Requerimientos técnicos

- Correr sobre KVM (Maquina Virtual Kilobyte)
- 160 Kb a 512 Kb de memoria total disponible para el entorno de java.
- Procesador de 16 a 32 bits.
- Bajo consumo de energía. A menudo estos dispositivos utilizan baterías como fuente de energía.
- Soporta algún tipo de conectividad a una red. La mayoría de estos dispositivos utilizan una conexión de banda-ancha baja cerca de 9600 bps a menudo es conexión inalámbrica.¹⁶

5.2.2 Requerimientos Funcionales

La aplicación se desarrollo bajo el perfil MIDP 2.0 (Mobile information Device Profile) lo cual utiliza los siguientes requerimientos:

¹⁵ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

¹⁶ osgg.net/omarsite/resources/proceedings/J2ME.pdf

- Seguridad usando HTTPS (hypertext transfer protocolo ver secure socket layer)
- Inclusión formal de Over the air (OTA)
- Mejora las capacidades de interfaz de usuario
- Parser para XML
- APL para sonido.
- Pantalla de al menos 96 x 54 pixeles
- Al menos un tipo de entrada para usuario.
- 128 kb de memoria no volátil para correr componentes MID (Mobile Information Device)
- Al menos 8 Kb de memoria no volátil para almacenar datos persistentes de las aplicaciones.
- 32 Kb de memoria volátil para correr Java.
- Conectividad inalámbrica a redes.¹⁷

¹⁷ osgg.net/omarsite/resources/proceedings/J2ME.pdf

6. DISEÑO

Las aplicaciones de J2ME no son programas completos, sino partes que se desarrollan y ejecutan en un entorno limitado por la máquina virtual, el perfil y la configuración. Esto quiere decir que carecen de un punto de entrada como puede ser la función *main* de *Java 2, Standard Edition*.

6.1 CICLO DE VIDA DE UN MIDLET

En J2ME, será el *Java Application Management* (JAM) el que ceda en determinados momentos la ejecución a la aplicación: podrá decidir cuándo se crea o se destruye, además de poder sacar temporalmente la aplicación del estado de ejecución (pausarla). Esto lo hará llamando a los métodos *startApp()*, *pauseApp()*, y *destroyApp()* definidos en el *MIDlet*.¹⁸

6.2 INTERFACES DE USUARIO

Hay dos maneras de establecer la interacción entre la aplicación y el usuario:

- **Interfaz de alto nivel.** Conjunto de clases proporcionadas por J2ME para proporcionar funcionalidades de entrada y salida de datos independientes del dispositivo. El programador sólo necesitará preocuparse de la gestión de los datos, ya que J2ME se encargará de dibujar en pantalla y obtener los datos de la manera más adecuada para el dispositivo donde se esté ejecutando la aplicación.

Se crea a partir de las clases derivadas de *Screen*:

- **Aler t:** es una pantalla que muestra una información al usuario y espera un tiempo antes de mostrar el siguiente elemento *Displayable*.
- **Lis t:** una pantalla que muestra una lista de opciones para elegir. Puede ser de tipo EXCLUSIVE o MULTIPLE, para selecciones individuales o múltiples respectivamente, o del tipo IMPLICIT, con la cual podremos definir un conjunto de acciones a realizar para el elemento seleccionado, mediante una lista de comandos.

¹⁸ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

- **TextBox:** es una pantalla que permite al usuario editar texto.
- **Form:** pantalla compuesta por un conjunto de elementos derivados de la clase *Ítem*, tales como imágenes, cajas de texto, agrupaciones de opciones, etc... A continuación se describen sus funciones:
- **ChoiceGroup:** similar a la clase *List*. Implementan la misma interfaz: *Choice*. Su función es similar, *ChoiceGroup* puede introducirse en un formulario junto con otros elementos.
- **CustomItem:** no tiene ninguna funcionalidad en concreto. Está creado para que mediante subclases de ésta, el programador pueda crear nuevos elementos de formulario.
- **DateItem:** componente editable para presentar y manejar fechas y horas.
- **Gauge:** componente gráfico que representa una barra de progreso. Dado un valor entre 0 y el valor máximo definido, dibuja una barra de tamaño proporcional a estos valores.
- **ImageItem:** elemento que contiene una imagen.
- **Spacer:** utilizado para crear separaciones entre elementos del formulario. No permite la interacción con el usuario.
- **StringItem:** elemento que contiene una cadena de caracteres. Su finalidad es únicamente informativa, ya que no es editable por el usuario.
- **TextField:** componente de texto editable. Similar a la clase *TextBox*, con la diferencia de que *TextField* está creada para ser introducida en un formulario y *TextBox* para ser visualizada a pantalla completa.

Hasta ahora, se han explicado los diferentes elementos que se pueden visualizar en pantalla. Ahora es necesario saber cómo puede interactuar el usuario con ellos. A todos estos elementos se les pueden asociar comandos mediante la clase *Command*: una clase que guarda información semántica sobre una acción. Ésta puede ser una etiqueta, un tipo de comando (OK, CANCEL, BACK, etc...) o un índice de prioridad.

Mediante el método *addCommand ()*, de las clases *Ítem* y *Displayable* se pueden asociar comandos y, cuando se ejecute un comando, será necesario tener asociada una clase que se encargue de recogerlo y tratarlo.

Deberá asociarse una clase, creada por el programador, que implemente la interfaz *CommandListener*, y asociada al *Ítem* o *Displayable* mediante el método *setCommandListener()*. Ésta clase *CommandListener* tratará los comandos cuando se produzcan mediante el método *commandAction ()*.¹⁹

¹⁹ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

- **Interfaz de bajo nivel.** Conjunto de clases para acceder directamente a los recursos del dispositivo (pantalla, teclado, etc...). El programador se encargará del dibujado en pantalla de dicha interfaz y de la gestión de los eventos del teclado, por lo que puede ser una tarea bastante complicada, debido a la gran variedad de pantallas, teclados, y dispositivos de entrada que existen. Como consecuencia, una interfaz programada a bajo nivel puede no funcionar correctamente en algunos dispositivos extraños e inusuales.

Todas las clases destinadas a las interfaces de usuario, tanto de alto como de bajo nivel, son subclases de *Displayable*. Ésta clase tiene dos clases derivadas: *Canvas*, para las interfaces de bajo nivel, y *Screen*, para las de alto nivel.²⁰

El Interfaz de bajo nivel se realiza a partir de la clase *Canvas*, cuya función principal es el dibujado en pantalla mediante la clase *Graphics*. Por otro lado, también gestiona diversos eventos relacionados con la entrada y salida, gestionados mediante la sobrecarga de los siguientes métodos de *Canvas*:

- **ShowNotify:** método llamado inmediatamente antes de que el objeto *Canvas* se muestre en pantalla. Puede sobrecargarse para inicializar algunos datos relacionados con la presentación del *Canvas* antes de que éste sea mostrado.
- **HideNotify:** llamado inmediatamente después de que el objeto *Canvas* salga de pantalla.
- **KeyPressed:** llamado cuando el teclado registra la pulsación de una tecla.
- **KeyRepeated:** llamado cada cierto intervalo de tiempo definido, si la tecla se mantiene pulsada.
- **KeyReleased:** llamado al dejar de pulsar una tecla.
- **PointerPressed:** llamado cuando el dispositivo apuntador es pulsado.
- **PointerReleased:** llamado cuando el dispositivo apuntador deja de estar pulsado.
- **PointerDragged:** llamado cuando el dispositivo apuntador se mueve mientras se mantiene pulsado.
- **Paint:** llamado cada vez que es necesario refrescar los datos gráficos de pantalla. Este método se encarga de dibujar en el *Canvas*, mediante un objeto de clase *Graphics*.²¹

²⁰ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

²¹ personals.ac.upc.edu/mario/others/proyecto_ulpgc.pdf

6.3 LISTADO DE CASOS DE USO, ESPECIFICACIÓN Y DIAGRAMA

Actor: Usuario final

Listado de casos de uso

- Leer descripción (leerdesc)
- Leer y analizar características (leer cara)
- Comprar aplicación (compapli)
- Descargar la aplicación (descapli)
- Instalar aplicación del juego de laberinto (instapli)
- Abrir el juego y empezar a jugar (abrir juego)
- Buscar asistencia vía email

Actor desarrollador

- Desarrollar la aplicación. (desapli)
- Probar la aplicación (probapli)
- Subir el juego a un portal o sitio de internet para su descarga. (subjuego)
- Desarrollar pantallas para descarga. (desapant)
- Brindan soporte vía e-mail.

Diagrama de casos de uso

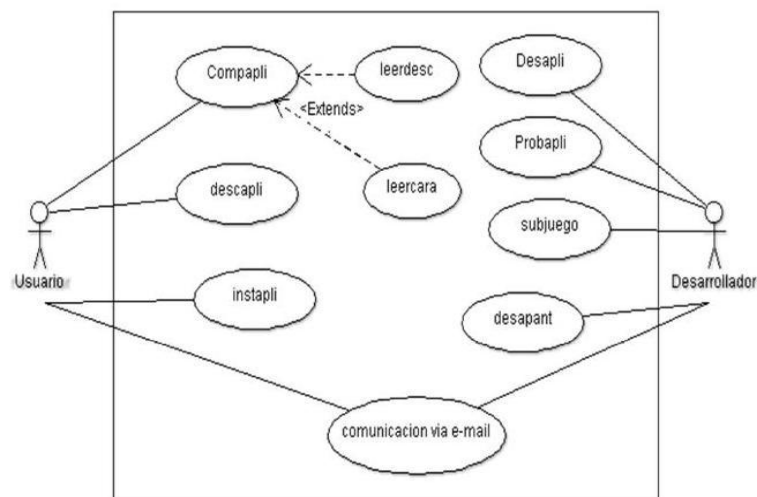


Figura 1. Diagrama de casos de uso

6.4 DIAGRAMA DE ESTADOS DEL JUEGO “DANGEROUS ROAD”

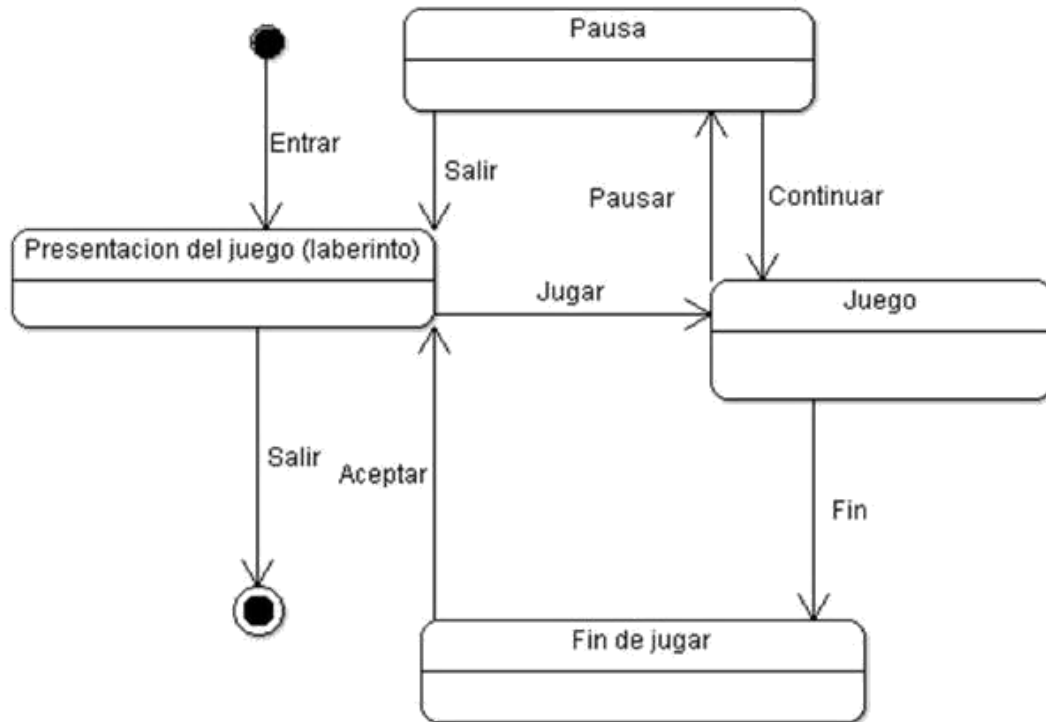


Figura 2. Diagrama de estados

Los estados son los siguientes:

1. La presentación, es la parte en donde se incluye el nombre del juego, si quiere jugar la partida o salir de la aplicación.
2. El juego, es la parte principal de la aplicación, en donde el usuario se divierte y se concentra.
3. Pausa. estado de pausa, donde se puede elegir entre continuar el juego o volver a la pantalla de presentación. A este estado se llega desde el juego, o automáticamente mediante el evento *pauseApp()* del Midlet, por si se recibe una llamada de teléfono para no perder la partida.
4. Fin. Aquí se muestra la finalización del juego y el número de puntos logrado.

6.5 DIAGRAMA DE CLASES

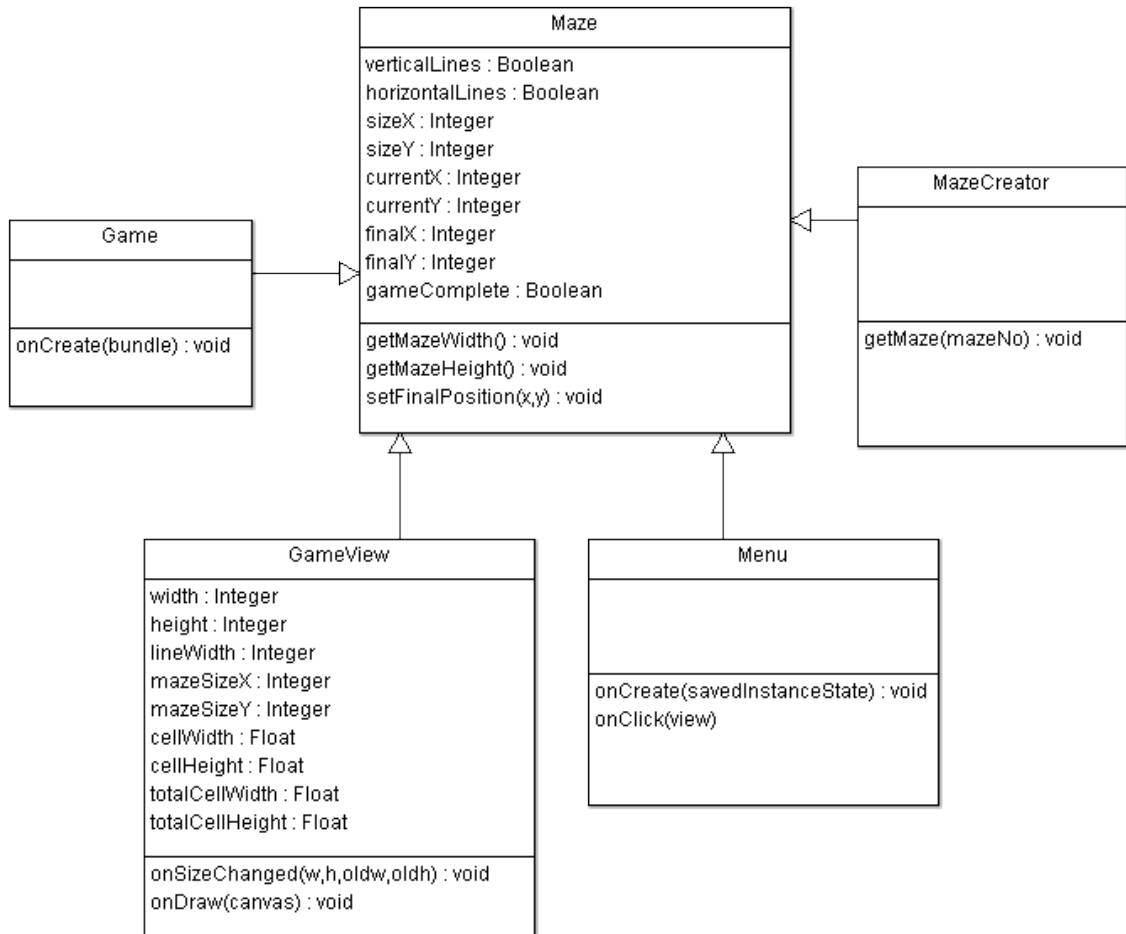


Figura 3. Diagrama de clases

7. IMPLEMENTACIÓN

7.1 DEFINICIÓN DE LA TECNOLOGÍA DE LA APLICACIÓN

La aplicación fue desarrollada en eclipse versión helios, utilizando la plataforma Java micro Edition (J2ME), para ser usada en cualquier dispositivo móvil que soporte java incluyendo el sistema operativo Android 2.2.

El móvil el cual usara la aplicación debe tener una buena memoria RAM y un microprocesador para que el juego funcione de una manera idónea.

El juego va a correr sobre KVM (Maquina Virtual Kilobyte) y debe haber 160 Kb a 512 Kb de memoria total disponible para el entorno de java y un Procesador de 16 a 32 bits.

La aplicación se implementará para el sistema operativo Android 2.2, esto quiere decir que para cualquier Smartphone que posea este sistema operativo para móviles Android 2.2 o superior lo puede instalar y utilizar sin ningún tipo de inconveniente o problema.

8. RESULTADOS Y PROYECCIONES

8.1 RESULTADOS

Se logró un juego divertido, fácil de jugar y entender y que funcione en un móvil con sistema operativo Android 2.2 que soporte java y además que ayuda a mejorar la concentración y la creatividad. Aunque el desarrollo de la aplicación fue satisfactorio, hubo ciertos inconvenientes en la configuración del J2ME con el eclipse al igual que con la instalación del plugin de Android.

El desarrollo del software uso los plugins de Android, los plugins de Eclipse (eclipse) estos plugins se configuraron en eclipse, el cual fue el programa donde se elaboro el software. También se descargo la plataforma SDK de Android 2.2 con el fin de lograr emular el juego como si estuviera corriendo en Android.

La elaboración del proyecto se elaboro en 4 pasos, investigación, elaboración de la monografía, termino y sustentación de la monografía, paralelamente a estos tres pasos se desarrollo la aplicación que en este caso era un juego de laberinto desarrollado en eclipse bajo la plataforma Java J2ME.

Se logro que el funcionamiento de la aplicación sea muy sencillo y que corra rápido en el sistema operativo Android 2.2.

8.2 PROYECCIONES

Se espera que el programa se pueda montar en la red, para su descarga, esto con el fin de que este al alcance de todos y que los beneficios que trae el jugar esta aplicación sea para todas las personas. Se podría poner en la página una donación voluntaria.

También se espera que se puedan elaborar mejoras por parte de todos los posibles usuarios del juego de laberinto. Como elaborar código para modificar ciertas características y hacerlo más divertido y entretenido, (código libre).

El uso de la aplicación será libre, es decir el usuario esta en la libertad de instalarlo donde quiera y de hacerle modificaciones si lo ve necesario.

Lograr que el juego sea de muy buena aceptación en los usuarios, ya que el juego no se desarrollo solo con el objetivo de entretener, también para desarrollar ciertas habilidades cognitivas, como la concentración, la creatividad entre otras.

9. CONCLUSIONES

- La elaboración del juego de laberinto “Dangerous Road” ayudo a comprender el funcionamiento de las aplicaciones para dispositivos móviles en este caso para Android 2.2
- Se logro un juego divertido, entretenido el cual permite fortalecer ciertas habilidades cognitivas y mentales en todos los usuarios que descarguen y jueguen el “Dangerous Road”; además posee buena jugabilidad y funciona de una manera eficiente y eficaz, lo cual da un valor agregado al juego.
- Se aprendió a utilizar el programa eclipse con la plataforma J2ME y SDK Android 2.2, así como a configurar el eclipse con los plugins de las dos plataformas nombradas anteriormente.
- El desarrollo de este juego permitió conocer las características técnicas y de funcionamientos que soporta el sistema operativo Android 2.2 con el fin de lograr una aplicación que sea soportada por dicho sistema operativo.
- Los dispositivos móviles en este caso los Smartphones en la actualidad se han convertido en equipos de uso cotidiano y por esta razón se elaboro el juego ya que en cualquier sitio o lugar se puede usar y recibir los beneficios que tienen los videojuegos.

WEBGRAFÍA

- Aspectos Legales Del Software Libre. 12 de marzo de 2011.
Disponible desde internet. [http://www.cecolda.org.co/index.php?option=com_content &task=view&id=17&Itemid=40](http://www.cecolda.org.co/index.php?option=com_content&task=view&id=17&Itemid=40)
- Critica sobre java. (s.f). Recuperado el 2 de marzo de 2011,
Disponible desde internet: <http://www.fayerwayer.com/2009/04/queda-poco-para-un-historico-eclipse-solar/>
- Concepto de la programación en java. (s.f). Recuperado el 1 de marzo de 2011.
Disponible desde internet: <http://todojava.awardspace.com/>
- El software libre. 1 de marzo de 2011. Disponible desde internet:
<http://www.monografias.com/trabajos12/elsoflib/elsoflib.shtml>.
- Explicación de las libertades básicas del software libre a nivel mundial. (s.f).
Recuperado el 1 de marzo de 2011.
Disponible desde internet: <http://www.monografias.com/trabajos51/software-libre/software-libre2.shtml>
- La historia de Android. 15 de marzo de 2011.
Disponible desde internet: <http://www.celularis.com/software/historia-android.php>
- Los videojuegos también sirven para pensar. 15 de febrero de 2011.
Disponible desde internet. <http://portal.educ.ar/noticias/entrevistas/hernan-moraldo-los-videojuegos-1.php>
- Ley De Software Libre En Colombia. 10 de marzo de 2011.
Disponible desde <http://sheilafierro.blogspot.com/2007/09/ley-de-software-libre-en-colombia.html>.
- Mario Macías Lloret. (2005). Diseño y creación de aplicaciones de entretenimiento sobre dispositivos móviles. Universidad de Las Palmas de Gran Canaria. descargado el 1 de abril de 2011,

Disponible desde internet: <http://personals.ac.upc.edu/mario/others/proyecto.ulpgc.pdf>

- Metodología de investigación científica. (s.f). Recuperado el 18 de abril de 2011.
Disponible desde internet: de <http://www.cienciaytecnologia.gob.bo/convocatorias/publicaciones/metodologia.pdf>
- Omar Salvador Gómez. (2005, septiembre 09). Desarrollo de aplicaciones para dispositivos móviles utilizando J2ME. Instituto tecnológico de San Juan del Rio Queretano. Descargado el 1 de abril de 2011.
Disponible desde internet: <http://osgg.net/omarsite/resources/proceedings/J2ME.pdf>
- Perspectiva de un programador java la programación puede sacar a Colombia del subdesarrollo. (s.f). Recuperado el 1 de marzo de 2011.
Disponible desde internet: <http://www.facebook.com/topic.php?uid=4768357861&topic=2907>
- Que es Android (s.f). Recuperado el 1 de marzo de 2011.
Disponible desde internet: <http://www.buenastareas.com/ensayos/Ensayo-Sobre-Android/1569351.html>