

UNIVERSIDAD MAYOR DE SAN ÁNDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**“USO DEL KINECT EN LOS VIDEOJUEGOS PARA LA
CONCIENTIZACIÓN SOBRE EL CUIDADO DEL AGUA”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS

POSTULANTE: UNIV. YOLGER PEREZ CONDE

TUTORA METODOLÓGICA: LIC. MENFY MORALES RÍOS

ASESORA: LIC. BRIGIDA ALEXANDRA CARVAJAL
BLANCO

LA PAZ – BOLIVIA

2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Agradecimiento

A Dios

Por la vida y salud que me dio y permitió llegar a este punto tan especial de mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valor mi vida.

A mi Familia

A mi padre Germán, por mostrarme la determinación de hacer las cosas y enseñarme que todo en la vida es alcanzable.

A mi madre Yolanda, quien me enseñó que la perseverancia y cario son la base de la felicidad y el éxito.

A mis Hermanas Patricia y Soledad, que me mostraron que las personas siempre tenemos algo bueno, que siempre debemos ponernos de pie en todas las pruebas que nos pone la vida.

A mis pequeñas Hermanas Noemi y Ericka y mi sobrino Nicolas, que le dieron alegría a mí vida y por sacarme una sonrisa en los malos momentos de mi vida.

¡Gracias por haber estado en mi vida, los quiero mucho!

A la Universidad

Un agradecimiento a la Universidad Mayor de San Andrés que durante años me ha acogido en sus aulas, las cuales han sido como un segundo hogar y en especialmente a la Carrera de Informática que me dio la oportunidad de formar parte de ella. ¡Gracias!

A mis Docentes

Gracias por su tiempo, por su apoyo así como por la sabiduría que me transmitieron para el desarrollo de mi formación profesional, en especial a mi tutora: Lic. Menfy Morales Rios por haberme apoyado en el desarrollo de este trabajo y del mismo modo a mi asesora: Lic. Brigida Alexandra Carvajal Blanco por su tiempo, paciencia, persistencia, comprensión y por impulsarme. ¡Gracias!

A mis Amigos

Quienes hicieron más grato mi estancia dentro de la universidad y por darme agradables recuerdos junto a ellos.

A mi Grupo de Estudios

Por permitirme ser parte de una gran experiencia en mi vida y de sus integrantes con los cuales vimos crecer este grupo, e especial un agradecimiento al Lic. Jorge Cupi, quien siempre confió en los integrantes del grupo e impulso a dar lo mejor de cada uno de ellos.

Dedicatoria

Dedicarte a ti por tomarte la molestia de haber leído esta tesis, espero que te ayude y que te sirva de apoyo, ya que esta tesis llego a ser un gran paso para mí y espero que sea uno para el tuyo.

Aprovechando, dedicarte las siguientes palabras, no tengas miedo de equivocarte, ten más miedo de no hacerlo, ya que gracias a nuestras equivocaciones podemos aprender más nunca renuncies siempre sigue adelante.

Espero que estas palabras te ayuden a seguir adelante e impulsarte a tomar nuevas y mejores metas.

¡Gracias por leer mi tesis!

Tabla de Contenido

Agradecimiento	1
Dedicatoria.....	3
Tabla de Contenido.....	4
Lista de Tablas.....	7
Lista de Figuras	9
Resumen	11
Abstract.....	12
Capítulo 1: Marco Referencial	13
1.1. Introducción	13
1.2. Antecedentes	13
1.2.1. Ámbito Medio Ambiente.....	13
1.2.2. Ámbito Informático	14
1.3. Planteamiento del Problema	17
1.4. Justificación	17
1.4.1. Justificación Educativa.....	17
1.4.2. Justificación Social	18
1.4.3. Justificación Técnica	18
1.5. Objetivos	18
1.5.1. Objetivos Generales.....	18
1.5.2. Objetivos Específicos	18
1.6. Límites y Alcances.....	18
1.6.1. Limites	18
1.6.2. Alcances.....	18
1.7. Metodología	18
1.7.1. Metodología de Investigación	18
1.7.2. Metodología Sistémica	19
1.7.3. Metodología de Desarrollo	19
1.8. Herramientas	19
1.8.1. Software.....	19
1.8.2. Hardware	20
Capítulo 2: Marco Teórico	21

2.1.	Videojuegos	21
2.1.1.	Tecnología	21
2.1.2.	Educación	22
2.2.	Estrategias de Concientización en el cuidado del Medio Ambiente.....	23
2.2.1.	Cuidado del Medio Ambiente.....	23
2.2.2.	Estrategias Didácticas.....	23
2.2.3.	Formulación de Estrategias	24
2.2.4.	Implementación de la Estrategia.....	24
2.3.	Tecnología Háptica	25
2.4.	Kinect XBOX 360.....	26
2.4.1.	Hardware: Partes Fundamentales	27
2.4.2.	Funcionamiento Básico	29
2.4.3.	Especificaciones del Sensor Kinect.....	31
2.4.4.	Rangos de Utilización del Sensor Kinect.....	31
2.4.5.	Recomendaciones de Uso.....	32
2.5.	Kinect for Windows SDK 1.8.....	33
2.6.	Unity 3D.....	34
2.6.1.	Multiplataforma	35
2.7.	Metodología SUM	36
2.7.1.	Fase 1: Concepto	37
2.7.2.	Fase 2: Planificación.....	37
2.7.3.	Fase 3: Elaboración	38
2.7.4.	Fase 4: Beta	38
2.7.5.	Fase 5: Cierre.....	38
2.7.6.	Fase 6: Gestión de Riesgo	39
Capítulo 3: Marco Aplicativo		40
3.1.	Introducción	40
3.2.	Metodología SUM	40
Fase 1: Concepto.....		40
Fase 2: Planificación.....		40
3.4.1.	Planificación Inicial.....	44
3.4.2.	Requisitos Funcionales.....	44

3.4.3. Requisitos No Funcionales	46
3.2.3. Diagrama de Actividades.....	48
Presupuesto.....	49
Fase 3: Elaboración.....	50
Diseño.....	50
Implementación.....	53
Fase 4: Beta.....	63
Fase 5: Cierre	63
Capítulo 4: Pruebas.....	64
Capítulo 5: Conclusiones y Recomendaciones.....	70
5.1. Conclusiones	70
5.2. Recomendaciones	70
Capítulo 6: Bibliografía	71
Capítulo 7: Anexos	74

Lista de Tablas

Tabla 1 Estrategia para elaboración del Videojuego	24
Tabla 2 Comparación con distintas Herramientas Hápticas	26
Tabla 3 Funcionamiento Basico de la Kinect.....	29
Tabla 4 Especificaciones del Sensor Kinect.....	31
Tabla 5 Recomendaciones de Uso.....	32
Tabla 6 Características SDK Kinect.....	33
Tabla 7 Requerimientos para el SDK Kinect for Windows v. 1.8	34
Tabla 8 Agrupación de Plataformas	35
Tabla 9 KV-01 Iniciar Aplicación	42
Tabla 10 KV-02 Ver Menú Principal	42
Tabla 11 KV-03 Reconocer Gestos	43
Tabla 12 KV-04 Configuración.....	43
Tabla 13 KV-05 Videojuegos.....	44
Tabla 14 Planificación Inicial.....	44
Tabla 15 RF-01 Iniciar Videojuego.....	45
Tabla 16 Ver Menú Principal	45
Tabla 17 RF-03 Abrir Menú Configuración.....	45
Tabla 18 RF-04 Definir gestos y posturas	45
Tabla 19 RF-05 Menú de Juegos	46
Tabla 20 RF-06 El agua es el hogar de los peces	46
Tabla 21 RF-08 Plugin Kinect ajustes con el videojuegos	46
Tabla 22 RNFR-01 Tiempo de Respuesta.....	47
Tabla 23 RNFI-01 Interfaz Gráfica	47
Tabla 24 RNFI-02 Contenido	47
Tabla 25 RNFO-01 Acceso a la aplicación	47
Tabla 26 RNFS-01 Plataformas.....	48
Tabla 27 Costos de Herramientas.....	49
Tabla 28 Costos de Software	50
Tabla 29 Costos Totales	50
Tabla 30 Modulo 0 Adaptación del Plugin de Kinect	51

Tabla 31 Modulo 1 Menú Principal.....	51
Tabla 32 Modulo 2 Configuración	52
Tabla 33 Modulo 3 El agua es el hogar de los peces.....	52
Tabla 34 Expectativas del Videjuego con los niños	65
Tabla 35 Aceptación del Videjuego por los niños.....	66
Tabla 36 Kinect un recurso Educativo	67
Tabla 37 Resultados de concientización con los niños.....	68

Lista de Figuras

Figura 1 Capturas WaterDate 2	14
Figura 2 Capturas de Fusion4D	15
Figura 3 Programa KODU en Ecuador.....	16
Figura 4 Fase de Proceso de la Metodología Sistémica	19
Figura 5 Arquitectura de una tecnología Háptica	25
Figura 6 Hardware Kinect	28
Figura 7 Rangos de Utilización del sensor Kinect	31
Figura 8 Plataformas disponibles en Untiy 3D.....	35
Figura 9 Fase de Proceso de SUM.....	37
Figura 10 Diagrama de Casos de Uso	41
Figura 11 Diagrama de Navegacional	49
Figura 12 Arquitectura General del Videojuego	51
Figura 13 Diagrama de clases del Plugin	53
Figura 14 Flujo de detección de Gestos.....	54
Figura 15 SwipeRight - State Detect	55
Figura 16 SwipeRight - State Completed.....	55
Figura 17 SwipeLeft - State Detect	56
Figura 18 SwipeLeft - State Completed.....	56
Figura 19 Stop - State Detect.....	56
Figura 20 Stop - State Completed.....	57
Figura 21 Clase LerpManager	57
Figura 22 Clase LerpObject.....	58
Figura 23 Menu Principal - Game	58
Figura 24 Menu Principal - Configuración.....	59
Figura 25 Configuración	59
Figura 26 Juego El agua es el hogar de los peces, en estado de detenido	60
Figura 27 Juego El agua es el hogar de los peces, en ejecución.....	60
Figura 28 Pagina principal de Sketchfab.....	61
Figura 29 Clase Generate Fish.....	61
Figura 30 Valores definidos en la clase GenerateFish	62

Figura 31 Escenario de Puntuación de la partida	63
Figura 32 Presentación con el curso	64
Figura 33 Participación del curso	64

Resumen

En esta tesis se propone la implementación de un videojuego cuyo objetivo tiene el de poder concientizar en el cuidado del agua, además de emplear el uso de nuevas herramientas tecnológicas como la Kinect for Windows v. 1.8., la cual nos permitirá usar como medio para el diseño de un interfaz Háptica. Para completar la elaboración de esta tesis se empleara el uso del lenguaje de programación C# usando como editor de código Visual Studio 2013 Community, y como herramienta de Desarrollo de la interfaz el motor de videojuegos en su versión libre Unity. Además, para el desarrollo del videojuego, se está proponiendo una metodología ágil de desarrollo orientada a videojuego SUM, la cual se adapta a distintos equipo de trabajo.

Abstract

In this thesis the implementation of a proposed game has aimed to raise awareness of the care of water, in addition to using new tools using technologies like Kinect for Windows v. 1.8., Which allow us to use as a means for design a haptic interface. To complete the development of this thesis using the programming language C # was employed as editor of code using Visual Studio 2013 Community, and as a development tool interface the game engine Unity in its free version.

In addition, video game development, is proposing an agile development methodology SUM oriented game, which adapts to different team.

Capítulo 1: Marco Referencial

1.1. Introducción

Con el transcurso del tiempo aparecieron nuevos recursos tecnológicos los cuales ayudan y apoyan las diferentes tareas en nuestra vida diaria, mucho de estos nuevos recursos también son usados para nuestro entretenimiento, si bien aparecieron nuevas tecnologías, también aparecieron nuevos métodos de apoyo en la educación y estos se aplicaron haciendo uso de estas nuevas tecnologías.

Si bien hace años un videojuego era solo considerado para el entretenimiento, hoy en día estos llegan a formar parte de nuestro aprendizaje, muchos de los nuevos recursos tecnológicos han apoyado el aprendizaje por medio de videojuegos o contenidos multimedia dinámicos.

Con el transcurso del tiempo, también se observó un descuido hacia el medio ambiente, ya que afecta la calidad de vida de las plantas, animales y seres humanos. Estos problemas tienen como principal causa la baja conciencia y cultura ambiental de las personas.

Esta tesis presentará un videojuego el cual apoyará a la concientización sobre el cuidado del agua por medio de un recurso tecnológico que es Kinect for Windows v. 1.8., para ello es importante hacer conocer al usuario cuán importante es el agua en nuestras vidas de una forma entretenida y dinámica.

1.2. Antecedentes

Para la realización de esta tesis se consideran distintas áreas que hacen énfasis al tema; en la actualidad existen una gran variedad de tecnologías que apoyan a concientizar el uso de los recursos de nuestro planeta, a continuación veremos casos de proyectos o propuestas.

1.2.1. Ámbito Medio Ambiente

- Instrumentos de Concientización en las Escuelas.

El objetivo de este proyecto fue el de concientizar a los niños de 3 a 6 años, a sus padres y docentes de tres escuelas, abarcando el tema del reciclaje. Este proyecto demostró que muchas personas tienen clara la idea del reciclaje pero no el cómo aplicarla en su vida. El proyecto obtuvo buenos resultados ya que tanto niños, padres e incluso docentes de las escuelas comprendieron de que tan necesaria es el tema del reciclaje en nuestras vidas. (Escobar, Quintero, & Serradas, 2006)

- Estrategia para la Educación Ambiental para Docentes

Este trabajo fue efectuado en la Escuela Superior de Formación de Maestros Bautista Saavedra de Santiago de Huata. El propósito del trabajo ha sido implementar estrategias de Educación Ambiental a futuros educadores, con el objeto de que sea el pilar fundamental de enseñanza para transmitir conocimientos, principios y valores para preservar el medio ambiente en que vivimos. (Ucedo, 2012)

1.2.2. **Ámbito Informático**

- WaterDate 2

Aplicación móvil disponible en Apple Store y Google Play, esta app tiene como objetivos principales: concientizar sobre la importancia del agua en la vida y cómo su escasez pone en peligro la supervivencia de millones de personas y por otra parte, interactuar en un proyecto solidario. Y es que WaterDate, además de colaborar en esta iniciativa de Fundación Aqualogy, también participa con UNICEF en programas de supervivencia infantil en el Sahel (Africa), una zona dónde la falta de agua potable y problemas de saneamiento afectan a millones de niños. El juego muestra, a través del personaje de Maji, una niña africana, la distancia que debe recorrer a diario para llegar hasta el agua potable. A lo largo del juego se van formulando preguntas y respuestas y se ofrecen cifras y comparaciones gráficas relacionadas con el acceso al agua en países en vías de desarrollo (Corresponsables/Servimedia, 2013)



Figura 1 Capturas WaterDate 2

Fuente: Obtenidas de Play Store

- Fusion4D

Es una innovadora aplicación presentada en el Imagine Cup 2011, cuya iniciativa a encontrar nuevos recursos dentro de la educación, es un innovador proyecto orientado al 3D que permite a los usuarios interactuar con objetos 3D, dando la sensación como si los objetos estén en sus manos, el usuario tiene permitido mover, rotar, escalar los objetos, o incluso observarlos por parte, también permite la navegación en el tiempo para ver lo que algunos objetos cambian en determinados periodos de tiempo.

FUSION4D es fácil de usar para el usuario solo tiene que realizar unos gestos con las manos e incluso permite la interacción por comandos de voz para manipular objetos o navegar dentro de la interfaz. Además de eso, el sistema utiliza dispositivos de bajo costo, tales como Kinect for Windows v1.8 y una pantalla que no requiere de especificaciones especiales.

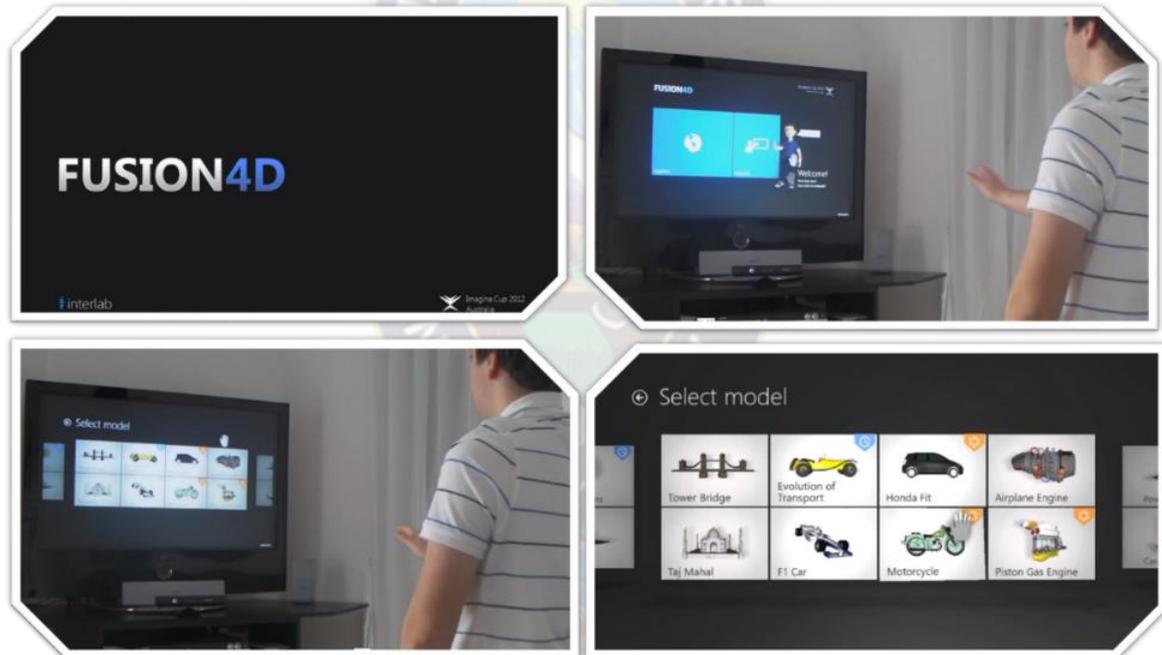


Figura 2 Capturas de Fusion4D

Fuente:(Duncan, 2011)

- Proyecto KODU en Ecuador

El centro Tecnológico Popular (CTP) fue seleccionado por la multinacional Microsoft como el centro piloto en Latinoamérica para la implementación del software KODU, herramienta tecnológica que motivó la realización del curso vacacional “Yo Programo”, a través del cual 168 menores aprendieron, durante el mes de marzo, a crear videojuegos.

Los expertos Jorge Cupi y Yamilet Contreras, integrantes del área de proyectos de innovación y Desarrollo de Microsoft, constataron desde el miércoles 25 de marzo el progreso de los pequeños en la administración de las funciones del software KODU.

Las creaciones de los chicos fueron revisadas, una a una por los profesionales de la multinacional, percatándose que “nuestros pequeños programadores“ a más de potenciar su ingenio y creatividad, plasmaron diferentes escenarios en juegos de acción, combate, aventura, con varios niveles de complejidad, lo que fue recibido con mucho agrado por parte de los representantes de la firma de software.

“El departamento de Desarrollo e Innovación de Microsoft está dedicado al apoyo a los estudiantes de colegio y universidad, pequeñas empresas de desarrollo hasta llegar a aplicaciones de nivel empresarial... Hemos encontrado en el CTP un aliado extraordinario”, manifestó Jorge Cupi.



Figura 3 Programa KODU en Ecuador

Fuente: (Guayas, 2015)

1.3. Planteamiento del Problema

Son numerosos los estudios que destacan el potencial educativo de los videojuegos en el desarrollo de habilidades de los niños.

A pesar de ello, está claro que la aplicación de los videojuegos no es la pieza maestra que solucione todos los problemas de nuestro sistema educativo, pero sin duda es un buen punto de partida para seguir avanzando.

Desde que se instauraron las primeras escuelas públicas, los métodos educativos basados en la repetición y en la memorización han perdurado hasta nuestros días. Sin embargo, estos modelos han dejado de ser validos debido a los cambios que las nuevas tecnologías de la información y la comunicación han supuesto para la sociedad.

Actualmente, necesitamos de un sistema educativo que utilice métodos que estimulen la creatividad. Las personas creativas son capaces de adaptarse mejor a los nuevos desafíos aportando más respuestas y soluciones originales fruto de la motivación. Y para ello necesitamos descubrir qué nos motiva.

El uso de los juegos en el aula no es algo nuevo pero, tradicionalmente, su utilización ha estado limitada a la memorización de contenidos, desaprovechando su potencial para desarrollar otras destrezas en alumnos y alumnas. Los creadores de videojuegos, sin embargo, han sabido convertirlos en todo un fenómeno cultural y en el entretenimiento preferido de personas de todas las edades. Poseen la capacidad de estimular a la vez que divierten y pueden, por tanto, ser una potente estrategia educativa. Muchas son las voces coincidentes en la necesidad de cambiar la relación enseñanza-aprendizaje y de incorporar los videojuegos a las aulas. (educaLAB, 2015) Para ello nos planteamos el siguiente problema:

“¿Se puede utilizar la tecnología de los videojuegos para concientizar sobre el cuidado del agua?”

1.4. Justificación

1.4.1. Justificación Educativa

El desarrollo de un videojuego abre una nueva brecha a la posibilidad de apoyar la concientización del cuidado de medio ambiente en nuestra ciudad por medio del uso de las nuevas tecnologías para mostrar más interés por parte de las personas.

1.4.2. Justificación Social

Hoy en día los nuevos recursos tecnológicos son utilizados en distintas áreas, por lo cual surge la idea de realizar un videojuego como alternativa para concientiza el cuidado del medio ambiente.

1.4.3. Justificación Técnica

El videojuego permite el reconocimiento del usuario y su interacción por medio del Kinect, sin contar con el control tradicional de consola para una mejor experiencia con el usuario. El videojuego permitirá apoyar a la concientización sobre el cuidado del agua.

1.5. Objetivos

1.5.1. Objetivos Generales

Desarrollar un Videojuego bajo el sensor de reconocimiento Kinect for Windows v. 1.8 para concientizar sobre el cuidado del agua.

1.5.2. Objetivos Específicos

- Buscar antecedentes de proyectos incluyendo tesis que hayan formado parte del mismo tema.
- Elaborar el guion del juego, bajo el tema de educación ambiental.
- Determinar el software y hardware con el cual se desarrollan el videojuego.
- Implementar el videojuego y hacer pruebas bajo una metodología ágil.

1.6. Límites y Alcances

1.6.1. Límites

Así mismo el videojuego está limitado a algunas funcionalidades:

- Solo se enfocara a tratar temas sobre el cuidado, uso e importancia del agua.
- Estará limitada a un idioma.

1.6.2. Alcances

Para cumplir con el objetivo principal de esta tesis, el videojuego tiene las siguientes funcionalidades:

- Reconocimiento de gestos del usuario.
- Interacción dinámica con el interfaz Háptica del Videojuego.
- Permitir informar sobre el uso y cuidado del agua.

1.7. Metodología

1.7.1. Metodología de Investigación

Se ha utilizado el Método Sistemico Heurístico para el desarrollo de la investigación. Este método de investigación es uno de los instrumentos lógicos más contemporáneos en el ámbito de la metodología, orientado a la percepción holística de la realidad de donde se extraerá la propia problemática y las soluciones correspondientes.

1.7.2. Metodología Sistemica

La Metodología Sistemica es uno de los instrumentos lógicos más contemporáneos en el ámbito de la metodología, orientado a la percepción holística (total) de la realidad de donde se extraerá la propia problemática y las soluciones correspondientes (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2012).



Figura 4 Fase de Proceso de la Metodología Sistemica

Fuente: (Barchini, Sosa, & Herrera)

La metodología del Marco Lógico, que es considerada un instrumento de planificación, el cual permite estructurar los principales elementos de un proyecto, subrayando los lazos lógicos entre los insumos previstos, la planeación de actividades y los resultados esperados.

1.7.3. Metodología de Desarrollo

Para el desarrollo del videojuego, se ha empleado la metodología ágil, SUM que se adapta a distintos requerimientos, utilizada para el desarrollo orientado a videojuegos de corto plazo.

1.8. Herramientas

1.8.1. Software

Para la implementación del videojuego se emplearan distintas herramientas de desarrollo:

- Microsoft Windows 8.1 64 bits (Sistema Operativo del equipo de Desarrollo).
- Microsoft Visual Studio 2013 Community (Entorno de Desarrollo del Código)

- Unity 5 (Entorno de Desarrollo de la interfaz del videojuego)
- SDK Kinect for Windows v. 1.8. (Set de herramientas para uso del Kinect)
- Kinect Wrapper Package for Unity 3D (Plugin de integración de Kinect con Unity)
- Inskcape 0.48 (Herramienta de desarrollo de los recursos gráficos del videojuego)

1.8.2. Hardware

Para la interfaz Háptica del videojuego se hará uso del siguiente hardware:

- Kinect XBOX 360 y Conector USB Kinect para PC



Capítulo 2: Marco Teórico

2.1. Videojuegos

Los videojuegos representan en la actualidad una de las entradas más directas de los niños a la cultura informática y a la cultura de la simulación. Estos son muy utilizados por niños y adolescentes, son muy criticados por sus contenidos y muy poco empleados por los educadores (Eguía Gómez, Contreras Espinosa, & Solano Albajes, 2012).

2.1.1. Tecnología

Lejos han quedado los días desde el desarrollo de los primeros videojuegos, caracterizados principalmente por su simplicidad y por el hecho de estar desarrollados completamente sobre hardware. Debido a los distintos avances en el campo de la informática, no solo a nivel de desarrollo de software, sino por la capacidad de Hardware, también dentro de la aplicación de métodos, técnicas y algoritmos, la industria del videojuego ha evolucionado hasta llegar a cotas inimaginables, tanto a nivel de jugabilidad como de calidad gráfica, tan solo hace unos años.

La evolución de la industria de los videojuegos ha estado ligada a una serie de hitos, determinados particularmente por juegos que han marcado un antes y un después, o por fenómenos sociales que han afectado de manera directa a dicha industria. Juegos como Doom, Quake, Final Fantasy, Zelda, Tekken, Gran Turismo, Metal Gear, The Sims o World of Warcraft, entre otros, han marcado tendencia y han contribuido de manera significativa al desarrollo de videojuegos en distintos géneros. (Eguía Gómez, Contreras Espinosa, & Solano Albajes, 2012)

A pesar de la acelerada evolución de la industria del videojuego, hoy en día existe un gran número de retos que el desarrollador de videojuegos ha de afrontar a la hora de producir un videojuego. En realidad, existen retos que perdurarán eternamente y que no están ligados a la propia evolución del hardware que permite la ejecución de los videojuegos. El más evidente de ellos es la necesidad imperiosa de ofrecer una experiencia de entretenimiento al usuario basada en la diversión, ya sea a través de nuevas formas de interacción, como por ejemplo la realidad aumentada o la tecnología de visualización 3D, a través de una mejora evidente en la calidad de los títulos o mediante innovación en aspectos vinculados a la jugabilidad.

No obstante, actualmente la evolución de los videojuegos está estrechamente ligada a la evolución del hardware que permite la ejecución de los mismos. Esta evolución atiende, principalmente, a dos factores:

- 1) La potencia de dicho Hardware, una mayor potencia hardware implica que el desarrollador disfrute de mayores posibilidades a la hora de por ejemplo, mejorar la calidad gráfica de un título o de incrementar la IA¹ de los enemigos.
- 2) Capacidades interactivas del mismo, una mayor riqueza en términos de interactividad puede contribuir a que el usuario de videojuegos viva una experiencia más profunda, por ejemplo, mediante realidad aumentada o pantallas táctiles.

Así mismo podemos ver que la actualización tanto de hardware y software para los videojuegos es cada vez más grandes y específicos.

2.1.2. Educación

Dentro de la educación se adoptado actualmente el uso de los videojuegos como un medio dinámico e interactivo en las aulas.

Muchos videojuegos han llegado a favorecer el desarrollo de terminadas destrezas que ayudan al desarrollo intelectual de la persona. Es decir, que la inteligencia, contrariamente a lo que ha podido afirmar en alguna ocasión, no parece deteriorarse al utilizar este tipo de entretenimiento digital. Más bien al contrario, parece que favorecen el desarrollo de determinados aspectos de ella, en especial los de carácter espacial. Incluso hay quienes afirman que los jugadores suelen ser individuos más inteligentes, aunque no parece existir ninguna relación entre ambos aspectos. En todo caso, se apunta que quizá los individuos con mayor inteligencia se verán más atraídos por tipo de entretenimiento.

Desde el punto de vista educativo los videojuegos, permiten una enseñanza eficaz, en bastantes ocasiones mucho mejor que el actual sistema escolar. Algunos factores que fomentan la motivación son intrínsecos a los videojuegos. En investigaciones producidas en niños y niñas de 12 a 16 años que utilizaron videojuegos de aventura. Las principales conclusiones obtenidas fueron que:

- Aumentan las estrategias de lectura visual de imágenes y de lectura del espacio tridimensional.
- Ayudaban a trabajar por observación y la verificación de hipótesis.
- Facilitaban la comprensión de simulaciones científicas.
- Incrementar las estrategias para recibir y procesar información recibida de varias fuentes simultáneas (procesamiento en paralelo).

¹ IA: Inteligencia Artificial

Los videojuegos ayudan a adquirir y desarrollar ciertas estrategias fundamentales para el aprendizaje: la resolución de problemas, el aprendizaje de secuencias, el razonamiento deductivo y la memorización. También, simplifican la realización de trabajos en grupos de tipo cooperativo y el aprendizaje basado en la resolución de tareas (Mendoza Lopez, 2010).

2.2. Estrategias de Concientización en el cuidado del Medio Ambiente

2.2.1. Cuidado del Medio Ambiente

El cuidado del medio ambiente no es solamente informar, sino y más importante, cambiar actitudes y motivar a la acción para resolver los problemas ambientales que nos afligen y prevenir futuras complicaciones.

Este es un proceso dirigido a desarrollar una población mundialmente consciente y preocupada por el ambiente en su totalidad y los problemas asociados. Esta población tiene el conocimiento, actitudes, habilidades, motivaciones y compromiso para trabajar individualmente y colectivamente hacia la solución y prevención de problemas relacionados con el bienestar del medio ambiente (Luna & Luna, 2001).

2.2.2. Estrategias Didácticas

Se define a las estrategias de enseñanza como los procedimientos o recurso utilizados por el agente de enseñanza para promover aprendizajes significativos.

Las investigaciones de estrategias de enseñanza han abordado aspectos como los siguientes: diseño y empleo de objetivos e intenciones de enseñanza, preguntas insertadas, ilustraciones, modos de respuestas, organizadores anticipados, redes semánticas, mapas conceptuales y esquemas de estructuración de textos, entre otros.

Estas investigaciones puede llegar a apoyar a la concientización de las personas en el cuidado del Medio Ambiente, estas pueden ser adaptadas a las nuevas tecnologías, gracias a que estas pueden ser usadas como transmisores para el aprendizaje de nuevos conocimientos.

Las estrategias didácticas, utilizadas hoy en día son los juegos, ya que estos despiertan un gran interés en la población joven.

Para encontrar la mejor estrategia dentro de la educación ambiental debemos tomar en cuenta que desde cualquier perspectiva tiene la responsabilidad de contribuir a través de acciones concretas, a la promoción de una nueva ética centrada en la protección del medio ambiente que permita el aprovechamiento y los cambios necesarios para la construcción de un modelo educativo transformador y participativo que se integre a las condiciones culturales económicas

y sociales, para ello nuestra estrategia deberá cumplir las siguientes tareas: (Escobar, Quintero, & Serradas, 2006)

- Crear conciencia.
- Formular, informar y transmitir conocimiento.
- Desarrollar destrezas y aptitudes.
- Promover valores.
- Definir criterios y normas de actuación.
- Orientar los procesos de toma de decisiones.

2.2.3. Formulación de Estrategias

Los objetivos muestran a donde se quiere llegar y la estrategia es el cómo llegar. Las estrategias se formularon tratando que las fortalezas sean aprovechadas y las debilidades sean disminuidas, para aprovechar las oportunidades y vencer las amenazas.

Tabla 1 Estrategia para elaboración del Videojuego

Metas	Proporcionar un videojuego para el cuidado del agua, recurso que facilita la enseñanza, la concientización.		
Actividades	1° La elaboración del Videojuego del cuidado del agua.	2° Evaluar en su totalidad el videojuego de cuidado del agua.	3° Implementar un nuevo recurso para la concientización del cuidado del agua.
Metodología	Hacer un Análisis sobre el videojuego.	Se elaborará un videojuego con las siguientes características Fácil comprensión y dinámico.	Se efectuara las pruebas de corrección del contenido del Videojuego.
Resultados Esperados	El videojuego apoyara a la concientización de cuidado del agua	El videojuego es utilizado para la concientización del cuidado del agua.	Se trabajara con niños de escuelas, para motivar el uso del nuevo recurso.

2.2.4. Implementación de la Estrategia

Luego de formular la estrategia, se realizó su implementación, es así que a esta última etapa es denominada la de acción, donde se expone y practica las estrategias formuladas.

2.3. Tecnología Háptica

El término “Háptica” carece de definición formal para la Real Academia de la Lengua Española; proviene del griego Hápto que significa “relativo al tacto”. Los dispositivos hápticos permiten al usuario interactuar con objetos del entorno a través del sentido del tacto.

La tecnología Háptica implica:

- La exploración del entorno por medio de sensores, los cuales adquieren información característica del medio.
- Procesamiento en el dispositivo de la información censada.
- Emisión de una respuesta (feedback) en forma de sensación Háptica.

A continuación se presenta una propuesta de nivel de sistema de la tecnología Háptica (Cortés, 2014).

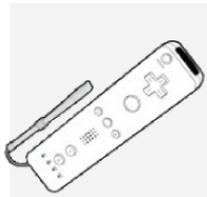
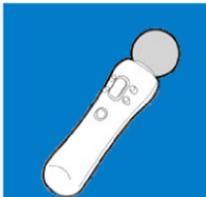
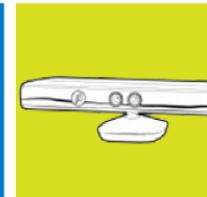


Figura 5 Arquitectura de una tecnología Háptica

Fuente: (Cortés, 2014)

Durante el transcurso de la tesis se evaluó las siguientes herramientas para determinar su valor pragmático en cuanto a las necesidades del videojuego de las cuales se consideraron las siguientes: Wiimote, PS Move y Kinect. A continuación se presentan los resultados obtenidos:

Tabla 2 Comparación con distintas Herramientas Hápticas

	  Wii Mote	  PS Move	  Kinect
Mando	Si	Si	Si
Cámara	No	Si	Si
Resolución	-	640x480 a 60 fps 320x240 a 120 fps	640x480 a 30 fps
Detecta Movimientos	Si	Si	Si
Reconocimiento Cuerpo	No	No	Si
Reconocimiento de Voz	No	Si	Si
Soporte para PC	Si	Si	Si

Aunque los tres dispositivos podrían ser totalmente aptos para poder realizar un control preciso de una aplicación, Kinect nos ofrece muchas posibilidades. En primer lugar, Kinect reconoce el cuerpo entero mientras que los demás dispositivos están sujetos al movimiento del brazo con el que se sujeta el mando, o ambos brazos (para lo que sería necesario disponer de varios mandos). De este modo, con Kinect se puede realizar un mayor número de movimientos, ya que reconoce hasta 20 partes del cuerpo diferentes (joints). Otro de los puntos fuertes de Kinect, es que se puede olvidar por completo de tener un mando en la mano, de modo que simplemente con un gesto de nuestro cuerpo se puedan realizar acciones en nuestros equipos de PC.

El uso de Kinect en ordenador es muy extendido que en el resto de los casos analizados, pudiendo encontrar muchos más ejemplos y ámbitos de uso que nos facilitara las cosas a la hora de desarrollar una aplicación.

2.4. Kinect XBOX 360

Kinect es un dispositivo de control de movimiento que fue inicialmente creado para jugar con la videoconsola XBOX 360 sin la necesidad de ningún tipo de mando o controlador. De este modo, el jugador hace uso de su propio esqueleto para interactuar con el juego, creando una

experiencia de usuario más realista y dando la opción al jugador de ser el protagonista de la historia.

Aunque el sensor Kinect ha sido desarrollado oficialmente por Microsoft, el diseño y la tecnología fue creada por la empresa israelí PrimeSense. Kinect fue anunciado por primera vez el 1 de Junio de 2009 en la Electronic Entertainment Expo 2009 bajo el nombre de “Project Natal” y posteriormente, en noviembre de 2010 sale al mercado como accesorio de la XBOX 360.

Con el lanzamiento de Kinect surgió una gran oportunidad para científicos, aficionados e inventores que comenzaron a utilizar y “piratear” Kinect para construir nuevas aplicaciones y conseguir aprovechar el potencial del que dispone este periférico. Tanto es así, que al poco de su lanzamiento, la empresa Adafruit, ofreció una recompensa a la primera persona que consiguiera hackear Kinect (Adafruit, 2010), siendo el ganador el español Héctor Mertin. Poco más tarde, la empresa PrimeSense lanzó el primer SDK no oficial para Kinect.

La primera reacción de Microsoft fue combatir a los piratas, aunque poco después se dio cuenta de que la liberación del software supondría una gran cantidad de nuevas herramientas para los usuarios. Así, el 16 de Junio del 2011 se publicó la primera beta de su SDK oficial, compatible con Windows 7, con licencia no comercial. Esta beta fue actualizada en noviembre de ese mismo año y, finalmente, en febrero de 2012 salía a la luz la versión 1.0 del SDK oficial de Microsoft. Con la libertad del software y la investigación realizada en entornos Windows con Kinect, se han podido realizar seros avances en multitud de disciplinas como marketing, medicina, negocios, ciencias de la computación, el entretenimiento y la robótica.

Las ventas del sensor Kinect alcanzaron los 10 millones de unidades vendidas en todo el mundo el 2011 (Acebo, 2012). Además, el libro Guinness World Records ha indicado oficialmente que es el producto de consumo electrónico de más rápida venta de la historia (Records, 2011). En dos meses, se vendieron una media de 133.333 unidades por día, haciendo así un total de 8 millones de unidades en sus primeros 60 días.

2.4.1. Hardware: Partes Fundamentales

El sensor Kinect dispone de cuatro partes fundamentales que se puede ver en la siguiente imagen y que se explica a continuación:

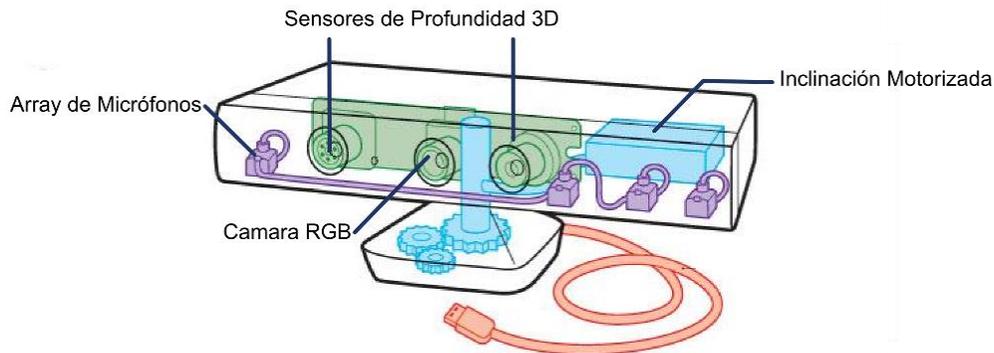


Figura 6 Hardware Kinect

Fuente: (Windows, 2013)

- a) Sensores de profundidad 3D. Los sensores tridimensionales hacen un seguimiento del cuerpo dentro del área del juego. Dispone de un proyector de profundidad (retícula izquierda) y un sensor de profundidad (retícula derecha), que calculan la distancia en función del tiempo que tarda en reflejar la luz.
- b) Cámara RGB. Una cámara RGB (Red, Green, Blue) que ayuda identificar y captar imágenes y videos con una resolución máxima de 640x480 a 30 fps².
- c) Array de micrófonos. Se usa un conjunto de cuatro micrófonos en el borde frontal inferior del sensor Kinect para el reconocimiento de voz.
- d) Inclinación motorizada. Se trata de un impulso mecánico en la base del sensor Kinect que inclina de manera automática el sensor hacia arriba o abajo según sea necesario hasta un máximo de 27°.

Además de estas partes el propio sensor Kinect dispone de lo siguiente:

- Memoria RAM de 512 MB.
- Acelerómetro, para estabilizar la imagen cuando se mueve.
- Ventilador, que no se encuentra encendido continuamente para no interferir con los micrófonos.

² Fps: Frames por segundo

2.4.2. Funcionamiento Básico

El funcionamiento básico de Kinect consta de tres partes diferenciada. Por una parte, el reconocimiento de imágenes, por otro el reconocimiento de voz y, finalmente, el motor de inclinación. A continuación se explicará brevemente en que consiste cada uno de ellos.

Tabla 3 Funcionamiento Basico de la Kinect

Funcionamiento Básico	
Reconocimiento de imágenes	<p>La configuración óptica de Kinect permite el reconocimiento de imágenes en tiempo real. La tecnología usada está disponible desde hace más de 15 años, por lo que no es altamente compleja. Microsoft, por su parte, ha conseguido algunos efectos y acciones que antes únicamente estaban disponibles a precios muy elevados.</p> <p>El sensor está compuesto de dos partes principales: un proyector y una cámara de infrarrojos VGA. La profundidad de los objetos es captada por la cámara gracias al rebote de los haces laser por el campo de juego, creando así un “campo de profundidad” que permite al sensor diferenciar entre los objetos estáticos de la sala y las personas utilizándolo. Este “campo de profundidad” consiste, básicamente, en que Kinect recibe este haz de luz como infrarrojos que varían en mayor o menor grado de color dependiendo de la distancia a la que se encuentran del sistema. De este modo, los cuerpos aparecen como rojo, verde, etc.; y los objetos más alejados aparecen en gris.</p> <p>Con los datos obtenidos en esta imagen, el software aplica una serie de filtros para que Kinect pueda saber qué es una persona y qué no, basándose en una serie de directrices como “una persona tiene dos brazos y dos piernas”.</p> <p>Una vez obtenida la información ordenada, se identifican las partes del cuerpo y crea un esqueleto en movimiento. Kinect tiene unas 200 posturas precargadas. De manera que se pueden llenar los espacios en blanco en caso de que se realicen movimientos que obstruyan la visión de su esqueleto. El principal inconveniente que podemos encontrar, es que los dedos no se asignan de forma individual en el esqueleto, impidiendo con</p>

	<p>ello una serie de movimientos. Todo esto es realizado por el sistema continuamente a 30 fps.</p>
<p>Reconocimiento de voz</p>	<p>El principal problema del reconocimiento de voz en Kinect, es que tiene que ser sensible a las voces hasta 5 metro de distancia, además de que debe ser capaz de ignorar los ruidos ambientales y cualquier otro sonido de la propia voz. Para poder realizar esto, el equipo de Microsoft realizo una serie de grabaciones de diferentes configuraciones, de forma que pudieran conseguir el mejor posicionamiento de los micrófonos en el sensor.</p> <p>El resultado de estas pruebas proporcionó la configuración actual de los micrófonos en Kinect. Se trata de un array de cuatro micrófonos posicionados boca abajo, uno a la izquierda y tres en la parte derecha.</p> <p>Este conjunto de micrófonos recoge las voces de la mejor manera posible. Aun así. Necesita ayuda del software de la cámara. La unidad de procesamiento cancela el ruido, mientras que un software usa la cámara para calcular de donde proviene el sonido, de modo que crear una burbuja alrededor del usuario y consigue separa la voz del usuario, omitiendo la del resto de personas que se encuentren alrededor.</p>
<p>Motor de inclinación</p>	<p>Tras las investigaciones de Microsoft en diferentes entornos y distintas configuraciones de los espacios y dadas las diferentes posibilidades que pueden encontrarse, Microsoft decidió que lo óptimo sería que la cámara pudiera moverse arriba y debajo de forma que pudiera adaptarse a cualquier entorno y cualquier campo de visión, calibrándose al espacio concreto y no teniendo que adaptar el espacio al a cámara.</p> <p>El motor se encuentra en la base de la cámara, pudiendo moverse arriba y abajo un total de 27°. La altura recomendada de utilización de la cámara para que el espacio de visión sea adecuado es de entre 1 y 2 metros. Además, existe un ventilador que únicamente entra en acción si es necesario, de forma que no interfiera con los micrófonos.</p>

2.4.3. Especificaciones del Sensor Kinect

A continuación se mostrarán las especificaciones técnicas del sensor Kinect, de modo que podamos conocer más sobre sus características y sobre cómo utilizarlo.

Tabla 4 Especificaciones del Sensor Kinect

Especificaciones del Sensor Kinect	
Ángulo de visión	43° de campo de visión vertical 57° de campo de visión horizontal
Ángulo de movimiento del motor	±27°
Velocidad de frames	30 fps
Resolución (Profundidad)	VGA (640x480)
Resolución (cámara RGB)	VGA (640x480)
Formato de audio	16 KHz, 16 bit PCM
Características de entrada de audio	Array de cuatro micrófonos con un convertidor analógico al digital de 24 bits (ADC) y procesamiento de señales de Kinect con cancelación de eco y eliminación de ruido de fondo.

Fuente: (Catuhe, 2013)

2.4.4. Rangos de Utilización del Sensor Kinect

El sensor Kinect está diseñado para videojuegos y escenarios similares. Originalmente estaba diseñado para su uso en XBOX 360, aunque con el SDK de Windows y el nuevo sensor Kinect para Windows, puede ser utilizado en otros contextos. A pesar de este cambio, tanto el diseño físico como las capacidades del sensor no se ven modificadas.

En la siguiente imagen se puede ver los rangos aplicados tanto el sensor Kinect de XBOX 360 (default Range), así como el sensor Kinect para Windows (Default Range y Near Range).

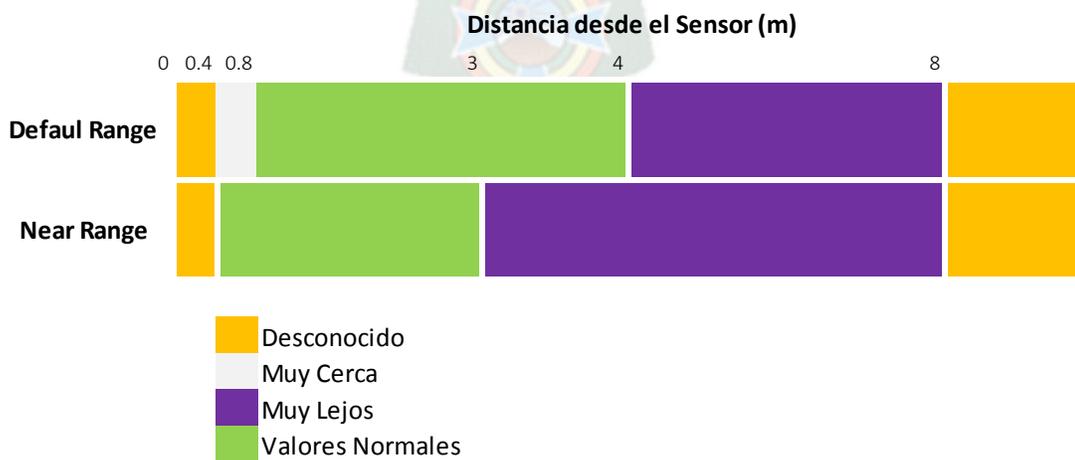


Figura 7 Rangos de Utilización del sensor Kinect

Este esquema se aplica a los valores devueltos por la API administrada, o por la API nativa con el flag `NUI_IMAGE_STREAM_FLAG_DISTINCT_OVERFLOW` encendido. Con este flag apagado, todos los rangos excepto para el caso de “valores normales” devuelve cero.

Con esta se puede observar como la distancia óptima del sensor, para poder utilizarlo sin ningún tipo de problema, se encuentra entre 1 y 4 metro. Esta distancia es ligeramente inferior (entre 0.5 y 3 metros) en el caso de Near Mode del sensor Kinect para Windows.

Cabe destacar, que el sensor Kinect solo detecta esqueletos que se encuentra de pie no reconoce figuras que se encuentren sentadas.

2.4.5. Recomendaciones de Uso

A continuación se mencionan una serie de recomendaciones para un uso óptimo del sensor Kinect, así como para evitar desperfecto en el aparato.

Tabla 5 Recomendaciones de Uso

Recomendaciones
Coloque el sensor sobre una superficie estable, en un lugar en el que no pueda caerse o ser golpeado durante su uso.
Ajuste la ubicación del sensor Kinect sólo moviéndose la base.
No ajuste el ángulo de visión del sensor con la mano, moviendo el motor de inclinación manualmente. Una vez instalado, los motores del sensor ajustaran el ángulo de visión para evitar el riesgo de dañar el sensor de Kinect.
No coloque el sensor Kinect fuera del alcance de la luz solar directa.
No utilice el sensor Kinect cerca de fuentes de calor.
Utilice el sensor de Kinect dentro de su rango de temperatura operativa (entre 5 y 35 grados centígrados).
Si el sensor está expuesto a una temperatura superior rango determinado, apáguelo y deje que la temperatura estabilice antes de usarlo de nuevo.
El sensor Kinect está protegido contra el sobrecalentamiento mediante un ventilador. Este ventilador está controlado por el firmware del sensor y apagará la cámara si la temperatura llega a los 90 grados centígrados. No existe una interfaz de software para controlar el ventilador del dispositivo.
Es necesario conectar el dispositivo a una fuente de alimentación externa, ya que, en caso contrario, su funcionalidad será limitada.

Si dispone de varios sensores Kinect, asegúrese de que están conectados en diferentes hub USB.

Asegúrese de que el dispositivo no está conectado en un hub utilizado por otros dispositivos.

Los controladores de Kinect no se instalaran correctamente, y no funcionarán correctamente si existe otro controlador para el sensor instalado en el PC.

No requiere herramientas para la calibración de audio y video.

Fuente: (Catuhe, 2013)

2.5. Kinect for Windows SDK 1.8.

Para desarrollar aplicaciones usando el sensor Kinect, deberá instalar el SDK Kinect for Windows v. 1.8., disponible en la página web oficial³.

Es una herramienta básica para los desarrolladores de aplicaciones. Este Kit permite crear aplicaciones que funcionen en los sistemas operativos Windows 7 y 8, utilizando la tecnología de sensores de Kinect.

Algunas de las características que incluyen el SDK, son:

Tabla 6 Características SDK Kinect

Características del SDK
Drivers para utilizar los sensores de Kinect en una computadora con Sistemas Operativos Windows 7 o 8.
API's ⁴ e interfaces de dispositivos, junto con documentación y técnicas para desarrolladores.
Ejemplos de Códigos.
Rastreo de esqueleto para la imagen de una o dos personas en movimiento, dentro del campo de visión de Kinect.
Cámara con profundidad en (X, Y, Z) para indicar la distancia del objeto con el sensor Kinect.
Procesamiento de Audio para un arreglo de 4 elementos de micrófonos, con un filtro de ruido acústico y eco, más una formación para identificar el origen del sonido, y su integración con la API de reconocimiento de voz "Microsoft Speech".

Fuente: (Catuhe, 2013)

³ <http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx>

⁴ API: Interfaces de Programación de Aplicaciones

A partir del SDK 1.5, Microsoft introdujo el Kinect for Windows Developer Toolkit donde contenía códigos de ejemplo y recursos para simplificar el desarrollo en aplicaciones usando el SDK del Kinect.

Para el buen funcionamiento del SDK deben ejecutarse programas que han sido desarrollados utilizando el SDK, dentro de un ambiente nativo de Windows. No se pueden ejecutar dentro de una máquina virtual, debido a que los drivers de Kinect y el SDK, deben ser instalados en la computadora que correrá la aplicación.

A continuación se muestran los requerimientos para su funcionamiento:

Tabla 7 Requerimientos para el SDK Kinect for Windows v. 1.8

Requerimientos de Hardware	Dual-Core, 2.66-GHZ como mínimo USB 2.0 Bus dedicado para el Kinect 2 GB de RAM Tarjeta Gráfica que soporte DirectX 9.0c Kinect for Windows Sensor
Requerimientos de Software	Microsoft Visual 2010 Express o versiones superiores. .Net Framework 4 (Instalado con Visual Studio) XNA Game Studio 4

Fuente: (Jana, 2012)

El SDK puede utilizarse para intereses personales, también está enfocado para entusiastas que buscan explorar de forma creativa interfaces NUI (Natural User Interface), con Kinect y su tecnología (Jana, 2012).

2.6. Unity 3D

Unity 3D es una de las plataformas para desarrollar videojuegos más completas que existen. Permite la creación de juegos para múltiples plataformas a partir de un único desarrollo, incluyendo juegos para consola (PlayStation, Xbox y Wii), escritorio (Linux, PC y Mac), navegador, móviles y tabletas (iOS, Android, Windows Phone y BlackBerry).

Es posiblemente la tecnología de mayor crecimiento en estos momentos, en especial con la reciente actualización que facilita el desarrollo de juegos 2D. Su principal limitación es el precio de su licencia completa, que puede alcanzar y superar los 4.500 \$ por una licencia completa para una sola persona. Analicemos los motivos de este hecho, las características de este entorno, sus limitaciones y su estado actual.

La primera versión de Unity se lanzó en la conferencia Mundial de Desarrolladores de Apple en 2005. Fue construido solamente para funcionar y generar proyectos en el equipo de la plataforma Mac y obtuvo el éxito suficiente como para continuar con el desarrollo del motor y

herramientas. Unity 3 fue lanzado en septiembre de 2010 y se centró en empezar a introducir más herramientas que los estudios de alta gama por lo general tienen a su disposición, con el fin de captar el interés de los desarrolladores más grandes, mientras que proporciona herramientas para equipos independientes y más pequeñas que normalmente serían difíciles de conseguir en un paquete asequible. La última versión de Unity, Unity 5, lanzada a principios de 2015, se anunció en Game Developers e incluye añadidos como Mecanim animation, soporte para DirectX 11 y soporte para juegos en Linux y arreglo de bugs y texturas. Desarrollado por creadores de juegos para mayor expectativa. (Wikipedia, 2015)

2.6.1. Multiplataforma

Una de las mayores ventajas de Unity 3D es que permite desarrollar juegos para muchas plataformas con realmente muy poco trabajo extra. A diferencia de otras tecnologías, con Unity 3D prácticamente no hay que hacer casi ajustes para adaptar compilar un juego para iOS teniéndolo ya para Android, salvo que queramos usar funcionalidades específicas de alguna de estas plataformas.

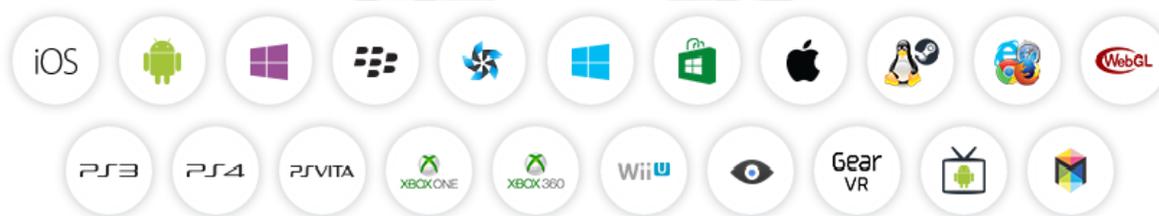


Figura 8 Plataformas disponibles en Unity 3D

Las plataformas permitidas son prácticamente todas las existentes actualmente y se pueden dividir en 4 grupos.

Tabla 8 Agrupación de Plataformas

<p>Móviles y Tablet</p>	<p>Unity 3D permite desarrollar para iOS, Android, Windows Phone 8 y BlackBerry 10. Las licencias completas para Windows Phone y BlackBerry están incluidas en la licencia básica de Unity 3D, principalmente porque de no ser así muy pocos desarrolladores harían esta inversión para estos dispositivos. iOS y Android, sin embargo, requieren licencias adicionales para hacer desarrollos completos.</p> <p>Para poder compilar para cada dispositivo es necesario algún trabajo extra mínimo. Para Android, por ejemplo, es necesario descargar e integrar el apk de Google para Android. iOS es más complicado: por</p>
--------------------------------	--

	<p>supuesto requiere primero un pago a Apple en concepto de licencia de desarrollador con un coste actual de 99\$ al año. Además necesitaras un Mac para poder realizar tu compilación final.</p> <p>Tanto si has desarrollado en Max como en PC (con Unity) en ambos casos Unity 3D no genera el fichero final sino un proyecto XCode que podrás abrir y compilar con tu entorno XCode en Mac.</p>
Escritorio	<p>También puedes compilar para PC, Mac y Linux. En este caso el trabajo extra a realizar es prácticamente nulo, salvo en el caso de Mac, que requiere de unos pasos adicionales.</p>
Consola	<p>Recientemente Unity 3D ha anunciado que es posible desarrollar sin coste adicional juegos para PS3/PS4, Wii U y Xbox 360. A pesar de esto, todavía no es trivial publicar juegos para estas consolas, al menos para empresas o equipos pequeños. Si bien Unity 3D no te exige licencias adicionales, si que necesitas los equipos de desarrollo de las diferentes plataformas, que deben ser proporcionados por Sony, Nintendo y Microsoft, respectivamente.</p> <p>Actualmente los desarrollos para consola realizados con Unity son aquellos de menor presupuesto, enfocados a descargas desde las tiendas online de las 3 consolas. Los primeros juegos ofrecidos por Sony para usuarios Plus de PS4 estaban hechos con Unity 3D. Los juegos más potentes de estas consolas usan otros motores, como Unreal.</p>

Fuente: www.unity.com

2.7. Metodología SUM

Las metodologías establecidas para el desarrollo de software no se adaptan a este proceso con garantías de calidad suficientes y no existe en este ámbito un claro planteamiento de cómo afrontar el trabajo. No obstante, se presenta en este proyecto la metodología SUM para videojuegos propuesta en Uruguay, la cual tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. SUM fue

concebida para que se adapte a equipos multidisciplinares pequeños, y para proyectos cortos (menores a un año de duración) (Acerenza, y otros, 2010).

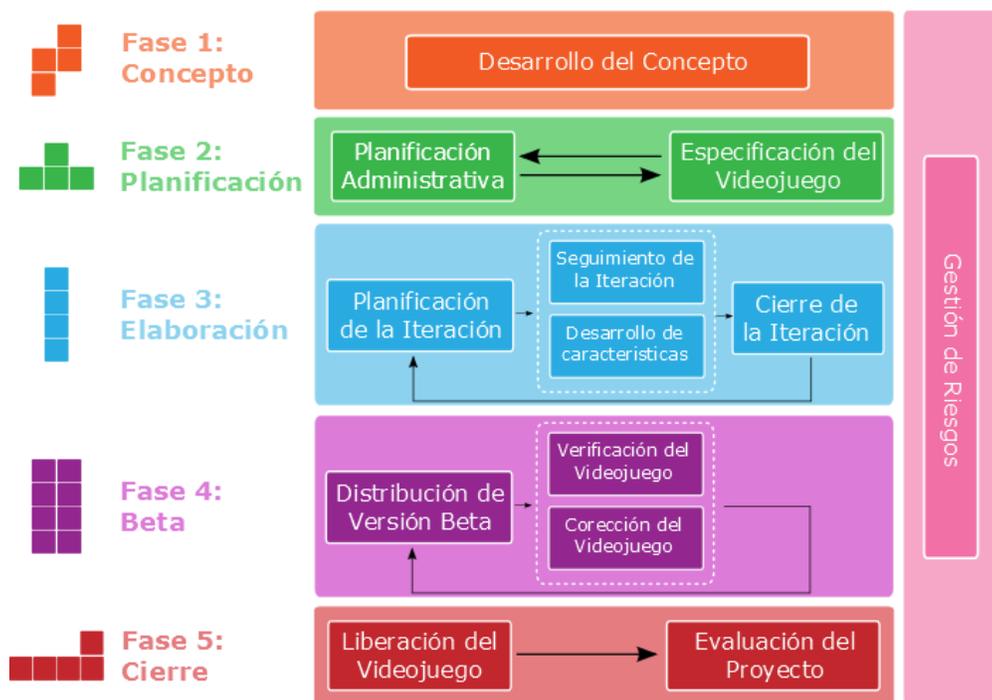


Figura 9 Fase de Proceso de SUM

2.7.1. Fase 1: Concepto

Tiene como objetivo principal definir el concepto del videojuego lo que implica definir 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). El concepto del videojuego se construye a partir de ideas y propuestas de cada rol involucrado sobre los aspectos a definir. Las propuestas se refinan a través de reuniones y se analiza su factibilidad con pruebas de concepto. Esta fase finaliza cuando se tiene el concepto valido entre todas las parte involucradas.

2.7.2. Fase 2: Planificación

La fase tiene como objetivo principal planificar las restantes fases del proyecto. Para ello es necesario definir las tareas, asignar las tareas para la elaboración de acuerdo a las necesidades técnicas del proyecto, determinar y tercerizar las tareas que el equipo no pueda cumplir, definir el presupuesto (si el proyecto cuento con uno) 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.

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.

2.7.3. Fase 3: 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 finalizar cada iteración. Éstas 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 planeados. 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 las tareas. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar la productividad.

2.7.4. Fase 4: Beta

La fase tiene como objetivo 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 a 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. Y el ciclo termina cuando se alcanza el criterio de finalización establecido en el plan del proyecto.

2.7.5. Fase 5: 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.

2.7.6. Fase 6: Gestión de Riesgo

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, mecanismo de monitoreo, estrategia de mitigación y plan de contingencia.



Capítulo 3: Marco Aplicativo

3.1. Introducción

El presente capítulo tiene como objetivo: desarrollar, construir e implementar un videojuego para cumplir los objetivos definidos anteriormente, con la finalidad de usar las nuevas tecnologías como medio de concientización del cuidado del medio ambiental.

Para el desarrollo del videojuego, se pone en práctica toda la fundamentación teórica expuesta en el capítulo anterior Marco Teórico, considerando como pilar fundamental el uso de las tecnologías de la información y comunicación.

En este documento se presenta en forma completa la metodología SUM para desarrollo de videojuegos. Esta se adapta a la realidad de proyectos de corta duración y equipos multidisciplinarios de pequeños grupos de trabajo.

3.2. Metodología SUM

En el capítulo anterior Marco Teórico, se hace referencia a la metodología empleada en esta tesis para la implementación del videojuego, para así completar los objetivos del mismo.

Fase 1: Concepto

Para la elaboración del videojuego se realizaron las investigaciones necesarias de videojuegos y proyectos que aplican el mismo tema de la tesis. De lo cual se obtuvo en el Marco Referencial toda la información para obtener un mejor resultado.

Fase 2: Planificación

En esta sección consta de una sola tarea, en la cual se desarrollan las características planificadas para la iteración a través de la ejecución de las tareas que la componen.

Se desarrollan y verifican las características planificadas para la iteración a través de la ejecución de las tareas que la componen. Para desarrollar una característica se deben completar todas las tareas definidas. Una vez completadas todas las tareas planificadas de la característica, estas se verifican de acuerdo a los criterios de evaluación establecidos. En caso de que no cumpla con alguno de los criterios se debe corregir hasta completar la tarea dentro de la característica.

Dentro de esta sección haremos uso de algunas herramientas para tener una planificación más ordenada.

Los casos de uso nos permitirán describir que hace el sistema desde el punto de vista del usuario, es decir, nos detallarán el uso del Sistema y cómo éste interactúa con los usuarios. De este modo

se podrá observar las interacciones típicas en un usuario (actor) y el sistema, describiendo que se hace sin entrar en el cómo los hace.

El primer paso para escribir un caso de uso de forma eficiente es definir el conjunto de actores que podrán llevar las acciones a cabo. Un actor, es un elemento que se comunica con el sistema y que es externo al sistema en sí mismo, es decir la persona o sistema que utiliza el producto. En nuestro caso, debido a que el controlador Kinect será una aplicación utilizada por distintos usuarios para jugar, nuestro actor será los jugadores.

A continuación mostraremos un diagrama de casos de uso, que nos será útil para ver de forma general y muy visual cuáles serán las acciones que el actor podrá realizar en el videojuego.

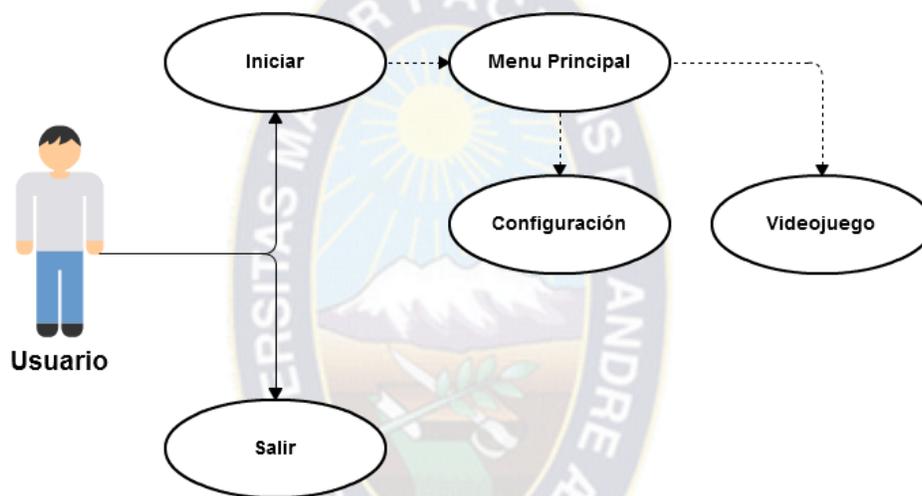


Figura 10 Diagrama de Casos de Uso

El siguiente paso será describir de forma detallada cada uno de estos casos de uso, de manera que se pueda ver cuál es la interacción entre el actor y el sistema, que condiciones deben darse para que el caso de uso pueda ser llevado a cabo o cuál es el flujo de interacción que da para realizar la acción.

En las siguientes tablas podremos ver las descripciones de todos los casos de uso mostrados en el diagrama de casos de uno mostrado anteriormente.

Tabla 9 KV-01 Iniciar Aplicación

Identificador:	KV-01	Título:	Iniciar Aplicación
Actores:	Usuario		
Descripción:	El usuario inicia el Videojuego		
Precondiciones:	Dispones del ejecutable en el ordenador y del sensor Kinect		
Flujo Normal:	<ol style="list-style-type: none"> 1. Conectar el sensor Kinect al equipo 2. Establece un espacio para el uso del sensor 3. Ejecutar el ejecutable de la aplicación 4. Esperar a que cargue por completo la aplicación 		
Flujo Alternativo:	No Aplica		
Postcondiciones:	Se cargará la aplicación y se desplegara el menú principal.		

Tabla 10 KV-02 Ver Menú Principal

Identificador:	KV-02	Título:	Ver Menu Principal
Actores:	Usuario		
Descripción:	Permite al usuario ver el menú principal		
Precondiciones:	El videojuego debe estar iniciado		
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario inicia el videojuego 2. La aplicación muestra las opciones dentro del menú 3. Realizare el gesto de deslizzamiento para movernos dentro del menú 4. Para seleccionar una opción detener el cursor en la opción por 2 seg. 		
Flujo Alternativo:	1. El usuario selecciona una opción del menu principal		
Postcondiciones:	Se muestra la selección del menú principal		

Tabla 11 KV-03 Reconocer Gestos

Identificador:	KV-03	Título:	Reconocer Gestos
Actores:	Usuario		
Descripción:	El usuario podrá realizar distintos gestos o posturas para la ejecución de distintas acciones definidas en el videojuego		
Precondiciones:	El videojuego debe estar iniciado		
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario realizará un gesto determinado 2. El videojuego ejecutará una acción dentro de cada escena específica 3. Algunos gestos no estarán habilitados dentro de algunas escenas. 		
Flujo Alternativo:	No aplica		
Postcondiciones:	Se ejecuta una acción dentro del videojuego		

Tabla 12 KV-04 Configuración

Identificador:	KV-04	Título:	Configuración
Actores:	Usuario		
Descripción:	El usuario podrá configurar la inclinación del sensor para un mejor uso.		
Precondiciones:	El videojuego debe estar iniciado		
Flujo Normal:	<ol style="list-style-type: none"> 1. Realizar cambios al ángulo de inclinación del sensor. 2. Guardar o ignorar los cambios efectuados. 		
Flujo Alternativo:	Volver al menú principal		
Postcondiciones:	Se ejecuta una acción dentro del videojuego		

Tabla 13 KV-05 Videojuegos

Identificador:	KV-05	Título:	Videojuegos
Actores:	Usuario		
Descripción:	El usuario podrá jugar dentro de los juegos implementados.		
Precondiciones:	El videojuego debe estar iniciado y seleccionar la opción videojuegos		
Flujo Normal:	<ol style="list-style-type: none"> 1. Seleccionar un juego dentro del menu de videojuegos 2. Comenzar Videojuego 		
Flujo Alternativo:	Volver al menu principal		
Postcondiciones:	Se ejecuta una acción dentro del videojuego		

3.4.1. Planificación Inicial

A continuación, se muestra una tabla con las estimaciones de tiempo que se dedicarían a cada una de las fases:

Tabla 14 Planificación Inicial

Nombre	Fecha Inicio	Fecha Fin	Duración
1. Planificación	04/08/2014	05/09/2014	25
2. Estudio de Viabilidad	08/09/2014	03/10/2014	20
2.1. Estudio de la tecnología	08/09/2014	19/09/2014	10
2.2. Estudio de librerías	22/09/2014	03/10/2014	10
3. Formación Inicial	06/10/2014	28/11/2014	40
3.1. Formación inicial Kinect	06/10/2014	17/10/2014	10
3.2. Formación inicial SDK Windows	20/10/2014	31/10/2014	10
3.3. Formación inicial Lenguaje C#	03/11/2014	28/11/2014	20
4. Análisis de Requisitos	05/01/2015	27/02/2015	40
4.1. Casos de uso	05/01/2015	16/01/2015	10
4.2. Analisis de Requisitos	19/01/2015	27/02/2015	30
5. Diseño	02/03/2015	03/04/2015	25
6. Implementación	06/04/2015	05/06/2015	45
7. Pruebas	08/06/2015	12/06/2015	5
8. Memoria y Documentación	15/06/2015	19/06/2015	5
Total	04/08/2014	19/06/2015	205

3.4.2. Requisitos Funcionales

A continuación se describirán los requisitos funcionales mediante los datos definidos en el punto anterior.

Tabla 15 RF-01 Iniciar Videojuego

Identificador:	RF-01	Título:	Iniciar Videojuego
Descripción:	EL usuario deberá poder iniciar el videojuego mediante un ejecutable o desde el propio entorno de desarrollo		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 16 Ver Menú Principal

Identificador:	RF-02	Título:	Ver Menú Principal
Descripción:	El usuario deberá poder ver el menú principal al iniciar la aplicación		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 17 RF-03 Abrir Menú Configuración

Identificador:	RF-03	Título:	Abrir Menú Configuración
Descripción:	El usuario deberá poder acceder al menú de configuración cuando este activo el menú principal de la aplicación a través de la selección de opciones dentro del menú principal		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 18 RF-04 Definir gestos y posturas

Identificador:	RF-04	Título:	Definir gestos y posturas
Descripción:	El usuario por medio de determinados gestos y posturas podrá realizar distintas acciones dependiendo del escenario que se encuentra.		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 19 RF-05 Menú de Juegos

Identificador:	RF-05	Título:	Menú de Juegos
Descripción:	El usuario podrá entrar al menú de videojuego al seleccionar la opción juegos dentro del menú principal		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 20 RF-06 El agua es el hogar de los peces

Identificador:	RF-06	Título:	El agua es el hogar de los peces
Descripción:	El usuario podrá ver los distintos animales que viven dentro del agua y además podrá recolectarlos para coleccionarlos		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 21 RF-08 Plugin Kinect ajustes con el videojuegos

Identificador:	RF-08	Título:	Plugin Kinect ajustes con el videojuego
Descripción:	Para la conexión entre el kinect y el engine se realizo algunos ajustes al plugin para que estos puedan cumplir correctamente todas las funciones		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

3.4.3. Requisitos No Funcionales

A continuación se describirán los requisitos no funcionales contemplados para la realización del proyecto. Esta definición se realizara de forma similar al caso de los requisitos funcionales, utilizando los mismos campos para su especificación.

Tabla 22 RNFR-01 Tiempo de Respuesta

Identificador:	RNFR-01	Título:	Tiempo de Respuesta
Descripción:	La aplicación deberá ejecutarse en tiempo real con un retardo máximo de 1 mini-segundo de forma que no influya en la ejecución del mismo		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 23 RNFI-01 Interfaz Gráfica

Identificador:	RNFI-01	Título:	Interfaz Gráfica
Descripción:	El sistema dispondrá de una interfaz gráfica intuitiva e amigable		
Versión:	2		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 24 RNFI-02 Contenido

Identificador:	RNFI-02	Título:	Contenido
Descripción:	El contenido del menú de la aplicación debe tener las opciones claras y ser conciso, de forma que no pueda llevar a confusión al jugador.		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 25 RNFO-01 Acceso a la aplicación

Identificador:	RNFO-01	Título:	Acceso a la aplicación
Descripción:	El acceso a la aplicación podrá ser llevado a cabo a través de un ejecutable o iniciando el proceso desde el propio entorno de desarrollo		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

Tabla 26 RNFS-01 Plataformas

Identificador:	RNFS-01	Título:	Plataformas
Descripción:	La aplicación funcionara unicamente en plataformas con sistema operativos Windows.		
Versión:	1		
Prioridad:	Alta		
Estabilidad:	Alta		
Necesidad:	Esencial		

3.2.3. Diagrama de Actividades

Mediante el siguiente diagrama se expondrá de forma muy visual el funcionamiento general de la aplicación, desde el punto inicial hasta el punto final. En él se podrán ver las diferentes decisiones u opciones durante la ejecución del videojuego.

Al Arrancar el videojuego, se carga los recursos y se muestra el menú principal. En este punto el jugador tiene dos opciones, de comenzar el juego o el de configuración.

Si el jugador selecciona la configuración este tendrá las opciones de habilitar o deshabilitar el audio del videojuego, además de configurar el ángulo de la vista del Kinect. Tendrá la opción de guardar la configuración o de ignorar los cambios.

Por último, si el jugador selecciona la opción de comenzar juego, este podrá acceder a una nueva partida.

Destacar que en cualquier momento de la ejecución del videojuego se podrá acceder al menú principal y se podrá abandonar el videojuego.

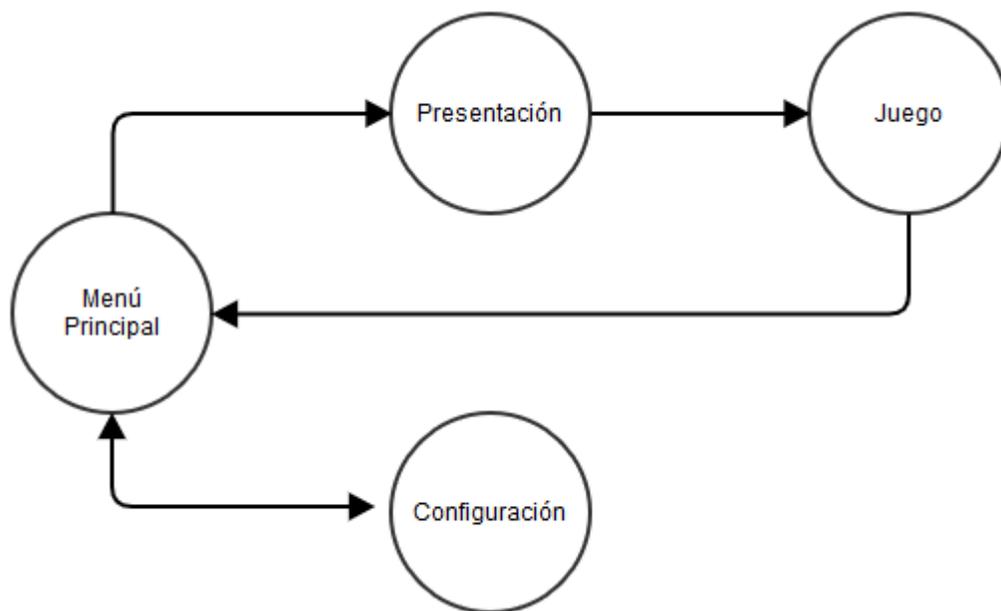


Figura 11 Diagrama de Navegacional

Presupuesto

Una vez establecida la planificación y la duración, se procederá a detallar los gastos relacionados por la realización del proyecto. En este punto, se incluirán tanto los costes directos (materiales, etc.), en esta planificación no se cuenta con costes indirectos.

Tal y como se ha visto en la planificación, el tiempo invertido en la realización del proyecto ha sido el comprendido entre principios de agosto y finales de junio, teniendo en total de 11 meses de duración, Con este dato, se obtendrán los diferentes costes asociados. A continuación se detallarán estos gastos.

Costos Directos

Equipos

En cuanto a costes de materiales, ha sido necesario adquirir un ordenador portátil, así como un sensor Kinect, tal y como se muestra en la siguiente tabla.

Tabla 27 Costos de Herramientas

Descripción	Coste (Bolivianos)	% uso dedicado	Dedicación (meses)
Ordenador	5000 Bs	100%	11 meses
Sensor Kinect	1200 Bs	80%	9 meses
Conector USB Kinect	200 Bs	80%	9 meses
Coste Total	6400 Bs		

Software

A continuación se detalla el coste del software que ha sido necesario utilizar durante el desarrollo del proyecto.

Tabla 28 Costos de Software

Descripción	Licencias	Coste/Licencia	Coste
Kinect for Windows SDK v1.8.	1	0.00 Bs	0.00 Bs
Visual Studio 2013 Community	1	0.00 Bs	0.00 Bs
Unity3D 5 free	1	0.00 Bs	0.00 Bs
Plugin Kinect with MS-SDK 1.12	1	0.00 Bs	0.00 Bs
Microsoft Office 2013 Student	1	0.00 Bs	0.00 Bs
Total			0.00 Bs

Coste Total

A continuación se resumirán los diferentes costes, mencionados y se calculará el coste total dentro de toda la planificación realizada.

Tabla 29 Costos Totales

Concepto	Coste Total
Equipos	6400.00 Bs
Software	0.00 Bs
Total	6400.00 Bs

De este modo, el coste total del proyecto asciende a 6400.00 Bs.

Fase 3: Elaboración

Diseño

El objetivo de este capítulo será el de solucionar los problemas planteados en el análisis anterior. Para ello, se describirá de forma detallada la arquitectura del sistema y sus diferentes módulos. El diseño se realizará para todos los componentes de la aplicación, de forma que ésta quede bien definida. Además, definiremos los prototipos de las interfaces que serán utilizadas en la ejecución del videojuego.

Arquitectura

A continuación se mostrara la arquitectura que se utiliza para la realización del videojuego. De este modo, podremos ver de una forma más precisa la estructura que sigue el videojuego y que facilitara el proceso de implementación.

En la siguiente imagen, se muestra la arquitectura general de la aplicación, de modo que se pueden observar las interacciones realizadas entre la aplicación para Kinect realizada y los elementos externos que forman parte del videojuego. Así, vemos como el jugador que utiliza el videojuego, se comunica con nuestro videojuego Kinect mediante gestos, teniendo en este punto

dos objetivos, diferenciados. Por un lado, encontramos la interfaz del videojuego y, por otro, el control de nuestro videojuego.

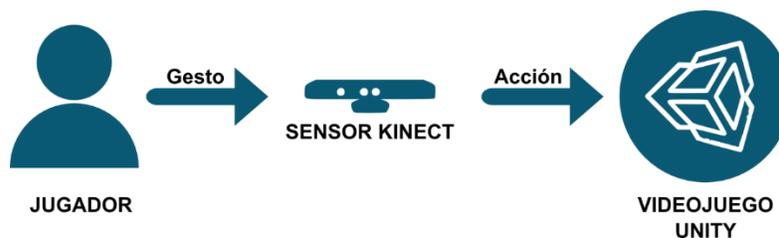


Figura 12 Arquitectura General del Videojuego

El siguiente paso, será describir de forma más detallada la estructura interna de nuestro videojuego en sí, de modo que se puedan conocer los diferentes módulos de los que se componen la aplicación para el sensor Kinect. A continuación se explicara de más específica la función y componentes de los que dispone cada uno de estos módulos. Para cada uno de ellos se describirá su propósito y las funciones que debe cumplir para que su funcionamiento sea completo y correcto.

Tabla 30 Modulo 0 Adaptación del Plugin de Kinect

Modulo 0	Adaptación del Plugin de Kinect
Propósito	El objetivo es reutilizar y adecuar los metodos y campos que tiene el plugin para sacar mejor provecho en el desarrollo.
Funciones	<ol style="list-style-type: none"> 1. Debera poder instanciar distintos tipos de gestos y posturas. 2. Podra reconocer si se completo el estado de un gesto o postura 3. Control de Posible errores.

Tabla 31 Modulo 1 Menú Principal

Modulo 1	Menu Principal
Propósito	Su objetivo es el de proporcionar al usuario una interfaz gráfica en la que pueda seleccionar las opciones de videojuego disponibles.
Funciones	<ol style="list-style-type: none"> 1. Deberá proporcionar una interfaz de usuario sencilla y fácil de aprender 2. Deberá proporcionar la opción de comenzar el videojuego. 3. Deberá proporcionar la opción de configuración. 4. Deberá permitir seleccionar las distintas opciones presentes en el menú principal.

Tabla 32 Modulo 2 Configuración

Modulo 2	Configuración
Propósito	Permitira al usuario ajustar el grado de inclinación del Sensor para un mejor uso
Funciones	<ol style="list-style-type: none"> 1. Permitira cambiar la inclinación del sensor 2. El control de esta escena es realizada por medio del teclado.

Tabla 33 Modulo 3 El agua es el hogar de los peces

Modulo 3	El agua es el hogar de los peces
Propósito	Permitir al usuario recolectar todo tipos de peces dentro del mar y la importancia que tienen.
Funciones	<ol style="list-style-type: none"> 1. El usuario podra moverse de Izquierda para recolectar los distintos tipos de peces. 2. El usuario tendra un tiempo limite para recolectar el mayor numero de peces. 3. Existira un contador encargado de controlar la cantidad de peces que tiene.

Desarrollo del Guion

Para comenzar el desarrollo del videojuego se procedió a la elaboración del guion, el cual ayudaría a realizar el videojuego.

Para lo cual se definió el siguiente guion:

Juan es el personaje principal del videojuego, él vive en una pequeña aldea donde cuidan de su lago ya que este es el hogar de sus peces, un día Juan se percata de que en el lago alguien lo ha estado contaminando. Juan necesita sacar los objetos que contaminan el lago, para que los peces puedan vivir dentro del lago de nuevo.

Para ello él se pondrá manos a la obra, para sacar todos los objetos que contaminan el lago de los peces, con la ayuda de los peces el deberá de recolectar la mayor cantidad de objetos para poder limpiar el Lago de toda la contaminación.

Juan tendrá el deber de limpiar el lago para que los peces puedan vivir tranquilos de nuevo.

Implementación

En esta sección se explicaran detalladamente los aspectos más relevantes de la implementación de la aplicación para Kinect. Haciendo uso de los puntos anteriores en los que se especifican los diferentes requisitos que se deben cumplir, así como el diseño que debe poseer el sistema, se implementarán las diferentes funcionalidades que harán nuestro videojuego de la manera que se espera.

La implementación se basa en dos clases fundamentales. Por un lado, la que nos permite interactuar con el sensor Kinect y la de controlar las funcionalidades dentro de cada escena. En este caso, como veremos a continuación, la creación de la propia interfaz se realiza de forma visual en el entorno de desarrollo, y es este entorno el que nos genera el código para cada escena. El trabajo de implementación de este nuestro videojuego se encontrar es permitir crear una instancia que permanezca activa en el transcurso del videojuego. De modo que será el núcleo de la implementación.

Módulo 0: Adaptación del Plugin de Kinect

Para el avance de este videojuego es necesario adecuar algunos métodos y campos definidos en el Plugin. A continuación se muestra la colección de clases que tiene definida el Plugin:



Figura 13 Diagrama de clases del Plugin

Kinect con MS-SDK es un conjunto de ejemplos de Kinect v1 que utiliza varios guiones principales, agrupados en una carpeta. Se muestra cómo utilizar avatares controlados Kinect, gestos detectados Kinect u otras cosas relacionadas con el Kinect en sus propios proyectos de Unity. Este Plugin utiliza el Kinect SDK/Runtime proporcionado por Microsoft.

El Plugin cuenta con un determinado número de funciones ya definidas para su uso, así mismo se pueden definir gestos y posturas personalizadas.

La estructura definida para la creación de gestos con el Plugin pasa por el siguiente flujo:

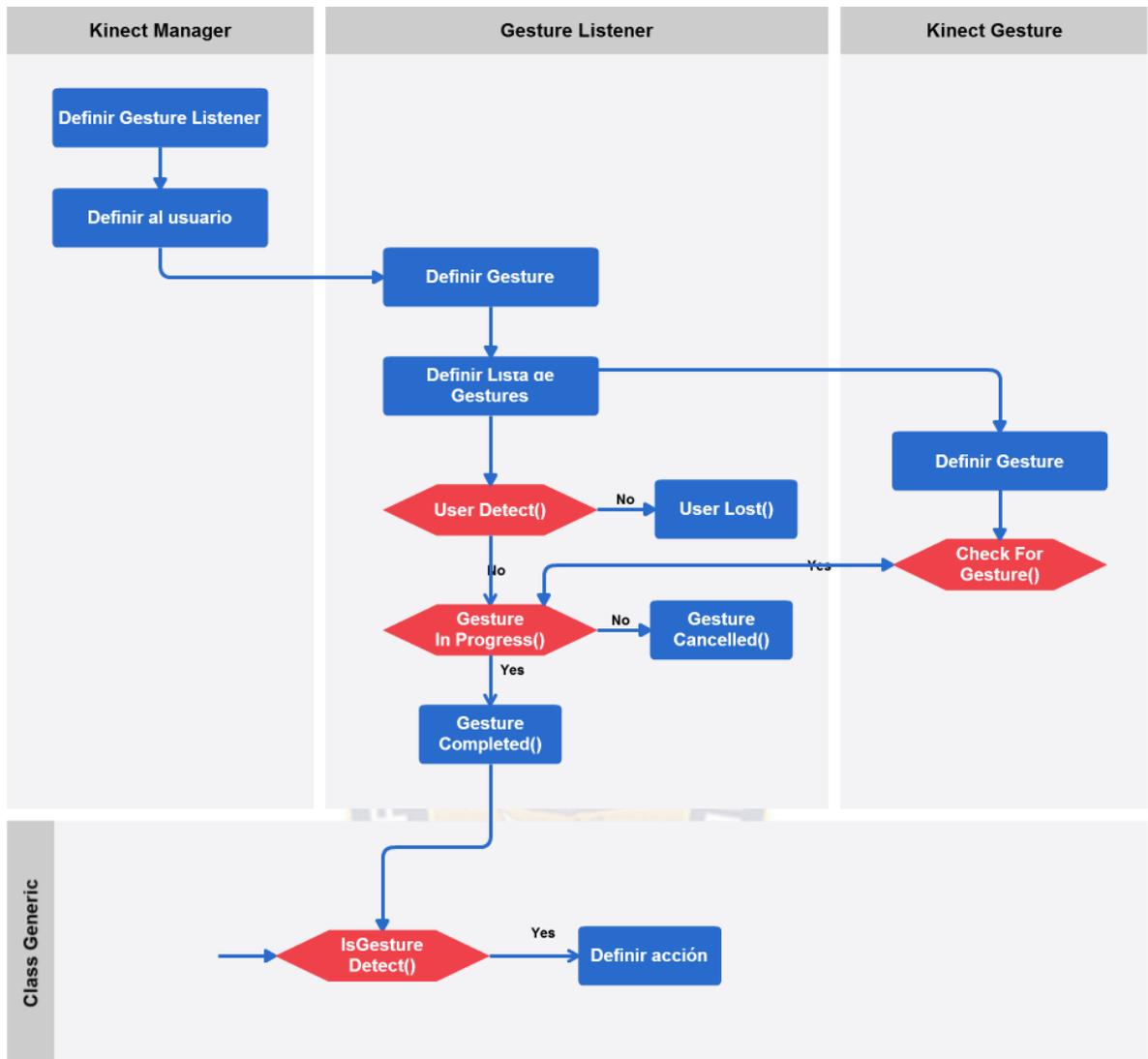


Figura 14 Flujo de detección de Gestos

Este Plugin fue liberado para la versión 1.5 del SDK de Kinect y además para demos dentro del escenario de videojuegos en Unity3d 4.0, por lo cual se pasaron a realizar algunas adaptaciones del Plugin para que no presente fallas con el uso del Kinect, las mejoras que se realizaron para el uso del Plugin no fueron muy grandes, pero si necesarias para su uso entre las cuales podemos mencionar:

- Detener el Kinect al Cerrar la aplicación y definir el ángulo de inclinación a 0.
- Buscar si ya existe un Kinect activado dentro de la escena, ya que este lo instanciaba en cada escenario.

Los gestos y posturas definidas dentro del Plugin son las siguientes:

- Zoom In (Acercar)

- Zoom Out (Alejar)
- Jump (Saltar)
- TPose (Postura en forma de T)
- Swipe Down/Up (Deslizamiento hacia Arriba y Abajo)

A continuación se definen los gestos elaborados para su uso:

- Swipe Righth (deslizamiento hacia la derechas)

```

case 0: // gesture phase 1 = Detect
  if (jointsTracked[leftHandIndex] &&
      jointsTracked[hipCenterIndex] &&
      jointsTracked[shoulderCenterIndex] &&
      jointsTracked[leftHipIndex] &&
      jointsTracked[rightHipIndex] &&
      jointsPos[leftHandIndex].y >= gestureBottom &&
      jointsPos[leftHandIndex].y <= gestureTop &&
      jointsPos[leftHandIndex].x >= gestureLeft &&
      jointsPos[leftHandIndex].x < gestureRight)
  {
    SetGestureJoint(ref gestureData, timestamp, leftHandIndex, jointsPos[leftHandIndex]);
    gestureData.progress = 0.1f;
  }
  break;

```

Figura 15 SwipeRight - State Detect

```

case 1: // gesture phase 2 = complete
  if ((timestamp - gestureData.timestamp) < 1.5f)
  {
    bool isInPose = jointsTracked[leftHandIndex] &&
                    jointsTracked[hipCenterIndex] &&
                    jointsTracked[shoulderCenterIndex] &&
                    jointsTracked[leftHipIndex] &&
                    jointsTracked[rightHipIndex] &&
                    jointsPos[leftHandIndex].y >= gestureBottom &&
                    jointsPos[leftHandIndex].y <= gestureTop &&
                    jointsPos[leftHandIndex].x > gestureRight;

    if (isInPose)
    {
      Vector3 jointPos = jointsPos[gestureData.joint];
      CheckPoseComplete(ref gestureData, timestamp, jointPos, isInPose, 0f);
    }
    else if (jointsPos[leftHandIndex].x >= gestureLeft)
    {
      float gestureSize = gestureRight - gestureLeft;
      gestureData.progress = gestureSize > 0.01f ? (jointsPos[leftHandIndex].x - gestureLeft) / gestureSize : 0f;
    }
  }
  else
  {
    // cancel the gesture
    SetGestureCancelled(ref gestureData);
  }
  break;

```

Figura 16 SwipeRight - State Completed

- Swipe Left (deslizamiento hacia la izquierda)

```

case 0: // gesture phase 1 = Detect
    if (jointsTracked[rightHandIndex] &&
        jointsTracked[hipCenterIndex] &&
        jointsTracked[shoulderCenterIndex] &&
        jointsTracked[leftHipIndex] &&
        jointsTracked[rightHipIndex] &&
        jointsPos[rightHandIndex].y >= gestureBottom &&
        jointsPos[rightHandIndex].y <= gestureTop &&
        jointsPos[rightHandIndex].x <= gestureRight &&
        jointsPos[rightHandIndex].x > gestureLeft)
    {
        SetGestureJoint(ref gestureData, timestamp, rightHandIndex, jointsPos[rightHandIndex]);
        gestureData.progress = 0.1f;
    }
    break;

```

Figura 17 SwipeLeft - State Detect

```

case 1: // gesture phase 2 = complete
    if ((timestamp - gestureData.timestamp) < 1.5f)
    {
        bool isInPose = jointsTracked[rightHandIndex] && jointsTracked[hipCenterIndex] &&
            jointsTracked[shoulderCenterIndex] &&
            jointsTracked[leftHipIndex] &&
            jointsTracked[rightHipIndex] &&
            jointsPos[rightHandIndex].y >= gestureBottom && jointsPos[rightHandIndex].y <= gestureTop &&
            jointsPos[rightHandIndex].x < gestureLeft;

        if (isInPose)
        {
            Vector3 jointPos = jointsPos[gestureData.joint];
            CheckPoseComplete(ref gestureData, timestamp, jointPos, isInPose, 0f);
        }
        else if (jointsPos[rightHandIndex].x <= gestureRight)
        {
            float gestureSize = gestureRight - gestureLeft;
            gestureData.progress = gestureSize > 0.01f ? (gestureRight - jointsPos[rightHandIndex].x) / gestureSize : 0f;
        }
    }
    else
    {
        // cancel the gesture
        SetGestureCancelled(ref gestureData);
    }
    break;

```

Figura 18 SwipeLeft - State Completed

- Stop (Postura definida para detener el flujo del juego)

```

case 0: // gesture detection
    if (jointsTracked[rightHandIndex] && jointsTracked[rightHipIndex] &&
        (jointsPos[rightHandIndex].y - jointsPos[rightHipIndex].y) < 0.1f &&
        (jointsPos[rightHandIndex].x - jointsPos[rightHipIndex].x) >= 0.4f)
    {
        SetGestureJoint(ref gestureData, timestamp, rightHandIndex, jointsPos[rightHandIndex]);
    }
    else if (jointsTracked[leftHandIndex] && jointsTracked[leftHipIndex] &&
        (jointsPos[leftHandIndex].y - jointsPos[leftHipIndex].y) < 0.1f &&
        (jointsPos[leftHandIndex].x - jointsPos[leftHipIndex].x) <= -0.4f)
    {
        SetGestureJoint(ref gestureData, timestamp, leftHandIndex, jointsPos[leftHandIndex]);
    }
    break;

```

Figura 19 Stop - State Detect

```

case 1: // gesture complete
bool isInPose = (gestureData.joint == rightHandIndex) ?
(jointsTracked[rightHandIndex] && jointsTracked[rightHipIndex] &&
(jointsPos[rightHandIndex].y - jointsPos[rightHipIndex].y) < 0.1f &&
(jointsPos[rightHandIndex].x - jointsPos[rightHipIndex].x) >= 0.4f) :
(jointsTracked[leftHandIndex] && jointsTracked[leftHipIndex] &&
(jointsPos[leftHandIndex].y - jointsPos[leftHipIndex].y) < 0.1f &&
(jointsPos[leftHandIndex].x - jointsPos[leftHipIndex].x) <= -0.4f);

Vector3 jointPos = jointsPos[gestureData.joint];
CheckPoseComplete(ref gestureData, timestamp, jointPos, isInPose, KinectWrapper.Constants.PoseCompleteDuration);
break;

```

Figura 20 Stop - State Completed

Módulo 1: Menú Principal

La creación del menú principal se elaboró por medio de recursos 3D, para una mejor presentación, así mismo se implementaron distintas funciones, para darle una animación adecuada al escenario como por ejemplo, el movimiento de traslación hacía, otro objeto, de la misma forma se hizo uso del Plugin para la interacción con el Kinect, para lo cual se elaboró la clase LerpManager el cual tiene como objetivo el de controlar la interacción dentro de este escenario para lo cual se definieron los siguiente campos:

- Objects: Es el total de objetos dentro del escenario para poder avanzar.
- Count: Este tiene la tarea de trabajar como índice dentro del escenario.
- GestureListener: Esta clase es donde se definen todos los gestos habilitados para el escenario.
- LerpObject: Esta clase es la encargada de realizar la animación de movimiento del objeto en distintas direcciones.

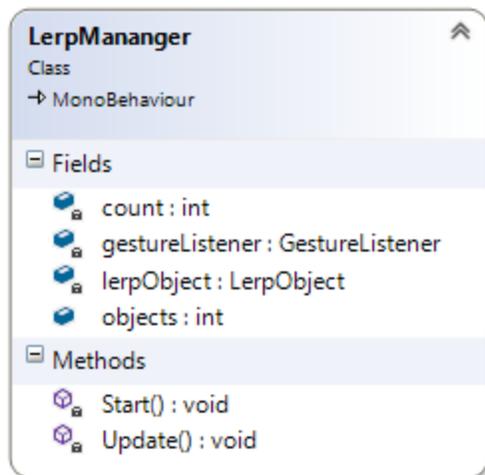


Figura 21 Clase LerpManager

La clase LerpManager es la intermediaria entre para la interacción con el Plugin para un mejor control, así mismo tiene definida la Clase LerpObject la cual nos permite movernos de una manera más óptima y mejor:

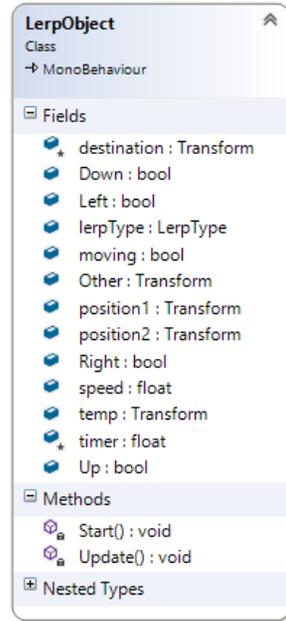


Figura 22 Clase LerpObject

La clase LerpObject tiene como tarea la de mover de una posición 1 a la posición 2, de manera más interactiva.



Figura 23 Menu Principal - Game

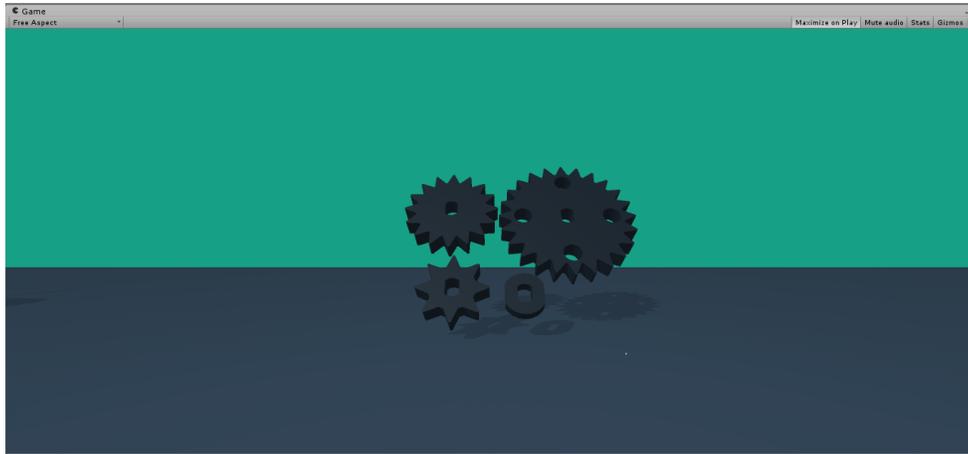


Figura 24 Menu Principal - Configuración

Módulo 2: Configuración

El objetivo de la configuración es el de ajustar el ángulo de inclinación del sensor para que este sea más cómodo para el usuario.

Además es relevante hacer mención del recurso que se utilizó para el desarrollo del videojuego como lo solicita la página para su uso de los modelos.

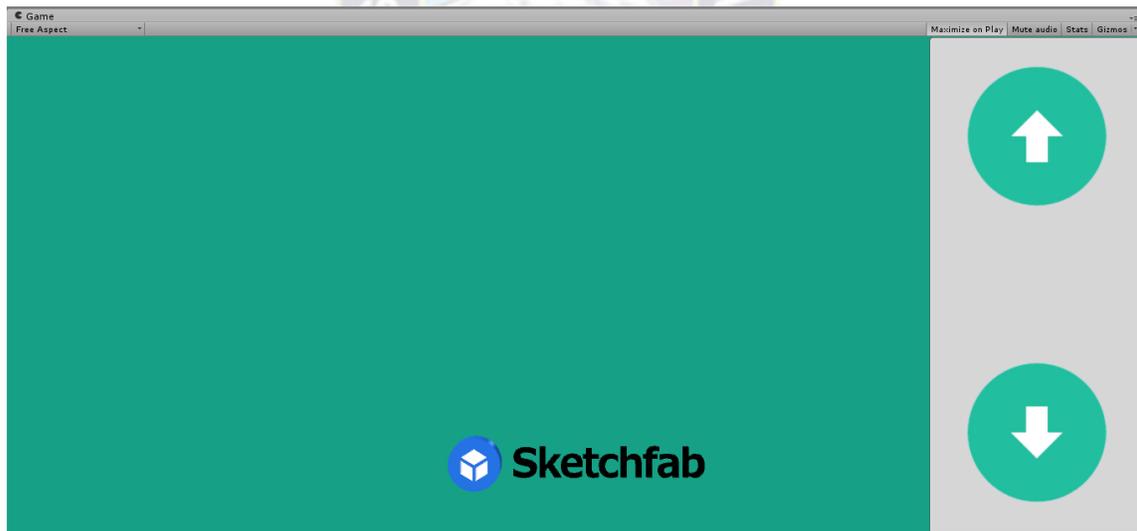


Figura 25 Configuración

Este escenario solo es utilizado por medio de los controles del teclado ya que si este era utilizado de por medio de gestos, al tener un ángulo de inclinación fuera de enfoque este ya no podría utilizarse, por lo cual se optó por el uso del teclado para un mejor uso.

Módulo 3: El agua es el hogar de los peces

El agua es el hogar de muchos peces por lo cual tenemos que cuidar de su casa, para que estos puedan vivir de manera pacífica dentro del mundo.

El juego consiste en poder observar la cantidad de peces que viven dentro del mar y de poder recolectar algunos para nuestra colección.

Para lo cual se utilizó la clase controller avatar del Plugin el cual permite seleccionar un modelo específico y copiar sus movimientos por medio del Kinect. Para que controller avatar tenga un mejor resultado este tiene que cumplir con el esqueleto definido correctamente según el modelo de prueba del Plugin.

Por medio del gesto stop el usuario podrá detener el juego, de la misma forma podrá reanudar para continuar con este.

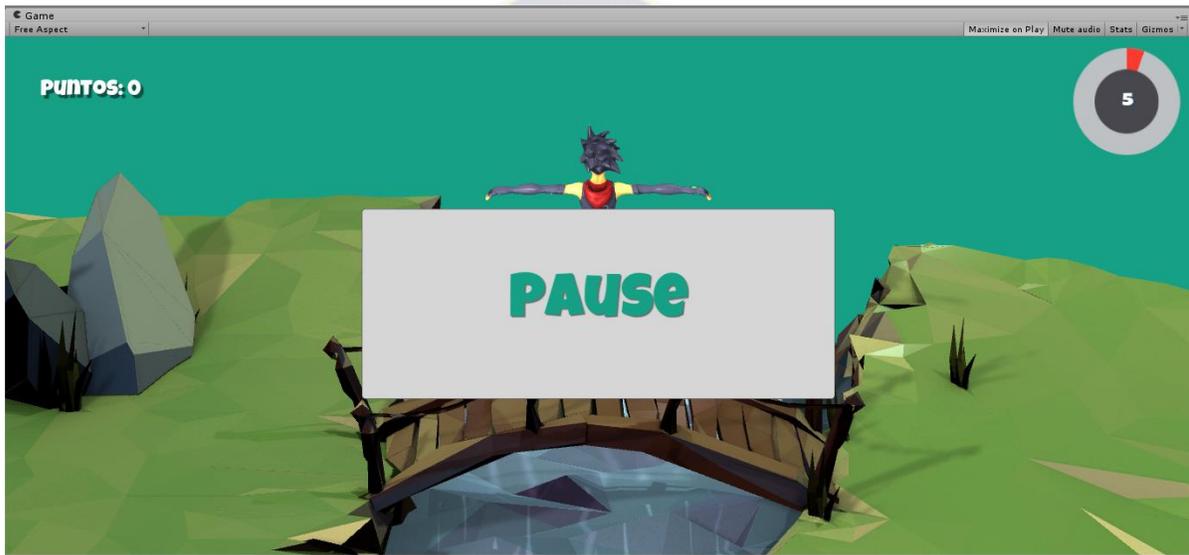


Figura 26 Juego El agua es el hogar de los peces, en estado de detenido

Dentro de este escenario se usaron modelos 3D para una mejor interfaz.

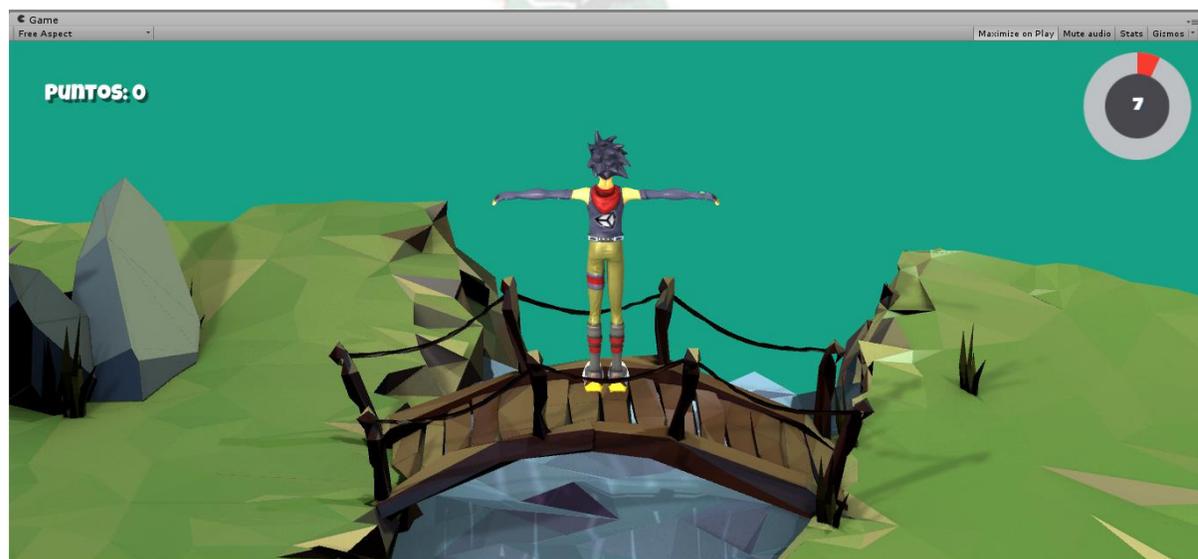


Figura 27 Juego El agua es el hogar de los peces, en ejecución

Los modelos de esta escena fueron obtenidos de la página sketchfab (<http://sketchfab.com/>) la cual tiene un repositorio de modelos de paga y gratis.

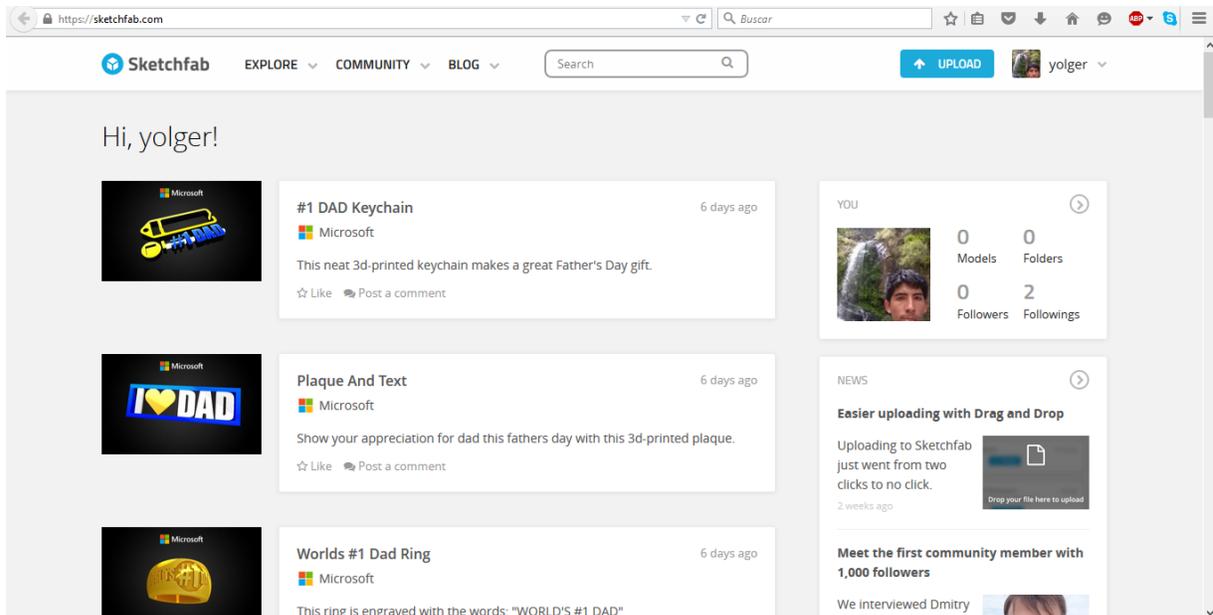


Figura 28 Pagina principal de Sketchfab

El juego consiste en obtener la mayor cantidad de peces posibles, el usuario podra moverse por medio del modelo, para recolectar los peces de la misma forma no solo serán los peces que aparezcan ya que aparece distintas objeto lo cuales tendra que evadir para no perder puntos, para la generación de objeto se implemento la clase GenerateFish.

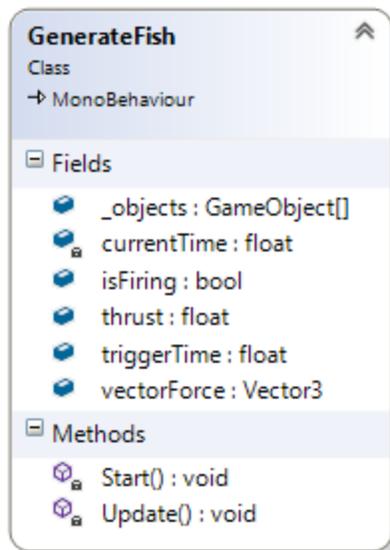


Figura 29 Clase Generate Fish

La clase GenerateFish tiene como objetivo el de crear objetos además de simular una trayectoria de lanzamiento, para que estos objetos puedan simular el efecto de lanzamiento estos tiene tener definida la propiedad Rigidbody la cual nos permite dar a nuestro objeto la propiedad de caer por medio de la fuerza de gravedad.



Para que este efecto se cumpla se definieron las siguientes valores dentro de la clase GenerateFish:

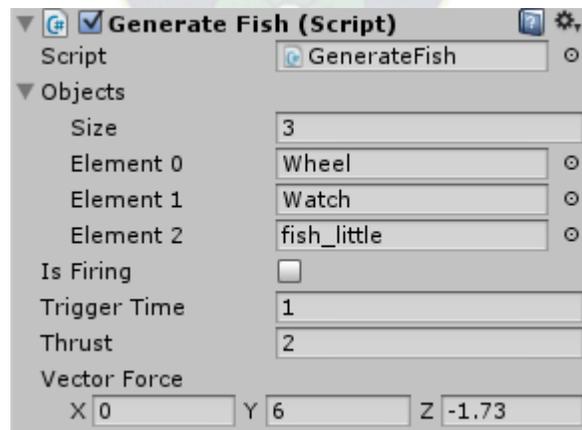


Figura 30 Valores definidos en la clase GenerateFish

Esta clase además cuenta con tres propiedades las cuales nos ayudaran para un mejor control:

- IsFiring, la cual permite que los objetos se creen o no se creen.
- TriggerTime, la cual se encarga de definir el tiempo para crear cada objeto.
- Thrust, es la fuerza con la que el objeto es arrojada.

Para este modulo se implemento un contador de tiempo el cual se encargara de controlar el tiempo que el usuario tiene antes de que termine. Para lo cual se implemento la clase ControlTime la cual se encarga de actualizar cada frame.

Una vez terminado el tiempo del usuario nos trasladara al escenario de score donde nos mostrara el puntaje que obtubimos y el puntaje maximo del videojuego.

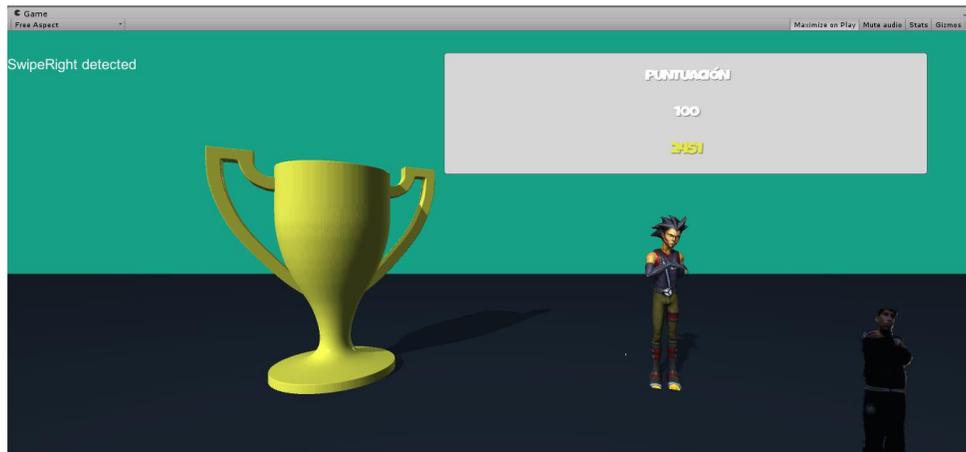


Figura 31 Escenario de Puntuación de la partida

Fase 4: Beta

Al completar todas las iteraciones dentro de la fase de elaboración se procede a liberar la primera versión para realizar las pruebas del videojuego, estas pruebas son detalladas a continuación en el capítulo 4.

Fase 5: Cierre

Una vez completada la versión beta se proceda al poner a disposición de distintos usuarios para la evaluación del desarrollo del videojuego.

Al obtener una evaluación satisfactoria de la versión beta del videojuego podemos cerrar el proyecto como primera versión.

Capítulo 4: Pruebas

Las pruebas con el videojuego, fueron realizadas en la U.E. Copacabana, en esta escuela se selección un curso al azar para realizar las pruebas correspondientes. El curso evaluado fue el 4to de Primaria con niños de edad 9 a 10 años, la población fue de 31 niños presentes.



Figura 32 Presentación con el curso

Al comenzar la demostración se observó la atención de los niños en cuanto al videojuego, también se observó el entusiasmo por usarlo.



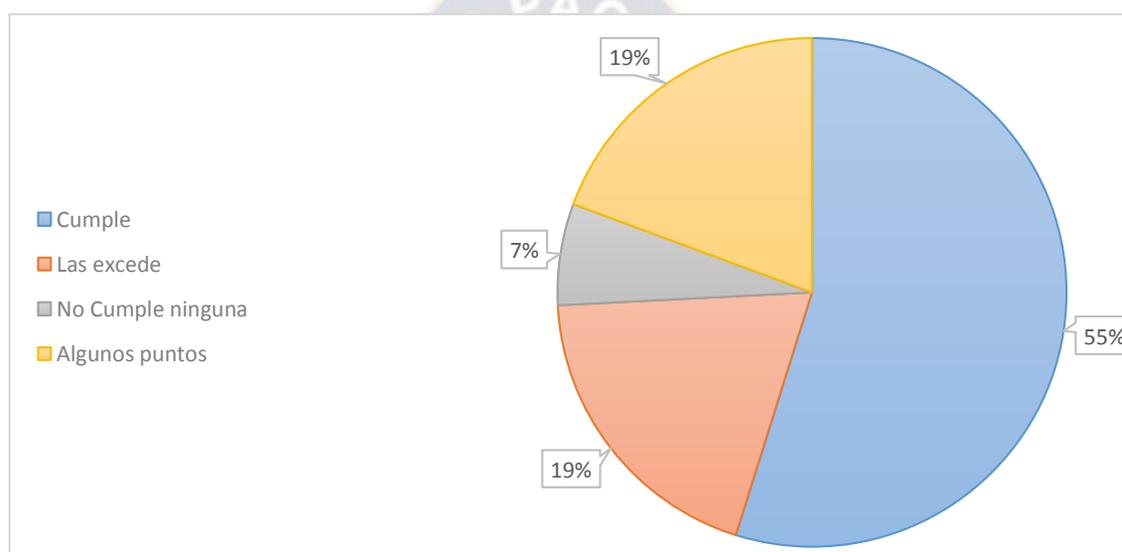
Figura 33 Participación del curso

De las pruebas realizadas con los niños se obtuvieron los siguientes resultados.

¿Qué expectativas tuvo el videojuego con los niños?

Tabla 34 Expectativas del Videojuego con los niños

	Frecuencia	Porcentaje
Solo se utilizó algunas funciones	6	19 %
No Cumple ninguna	2	7 %
Las excede	6	19 %
Cumple	17	55 %
	31	100 %

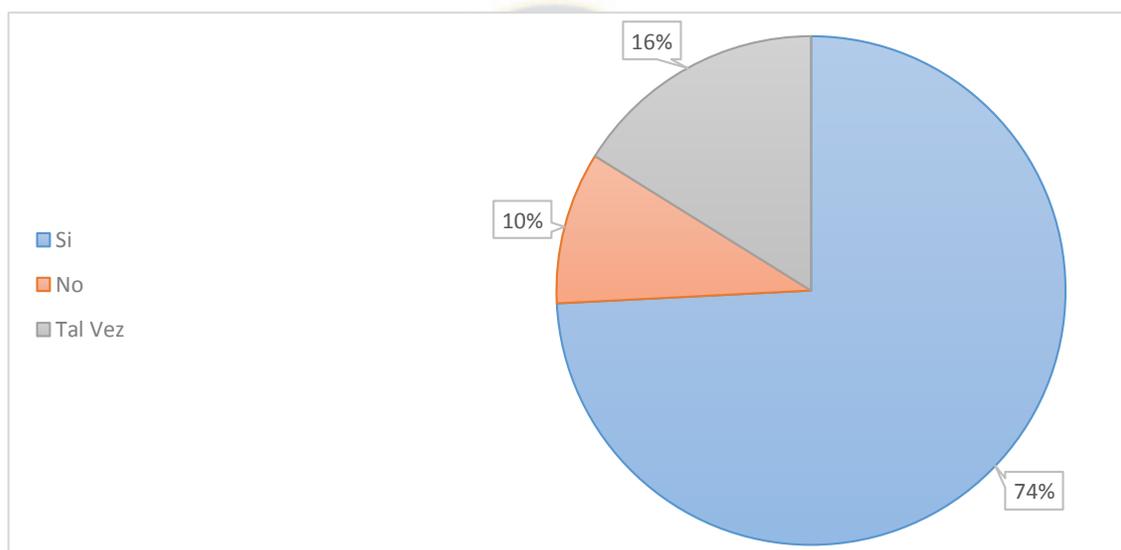


El videojuego tuvo una buena expectativa por parte de los niños ya que se obtuvo 55% del uso de sus funciones al momento de utilizarlo, así mismo se comprobó que un 19% excedió sus expectativas dentro del videojuego, un 7% no mostró mucha importancia a las funciones del videojuego y un 19% no utilizó las funciones del videojuego ya que solo participaron en el videojuego y se sentaron.

¿El videojuego fue aceptado por los niños?

Tabla 35 Aceptación del Videojuego por los niños

	Frecuencia	Porcentaje
Si	23	74 %
No	3	10 %
Tal vez	5	16 %
	31	100 %

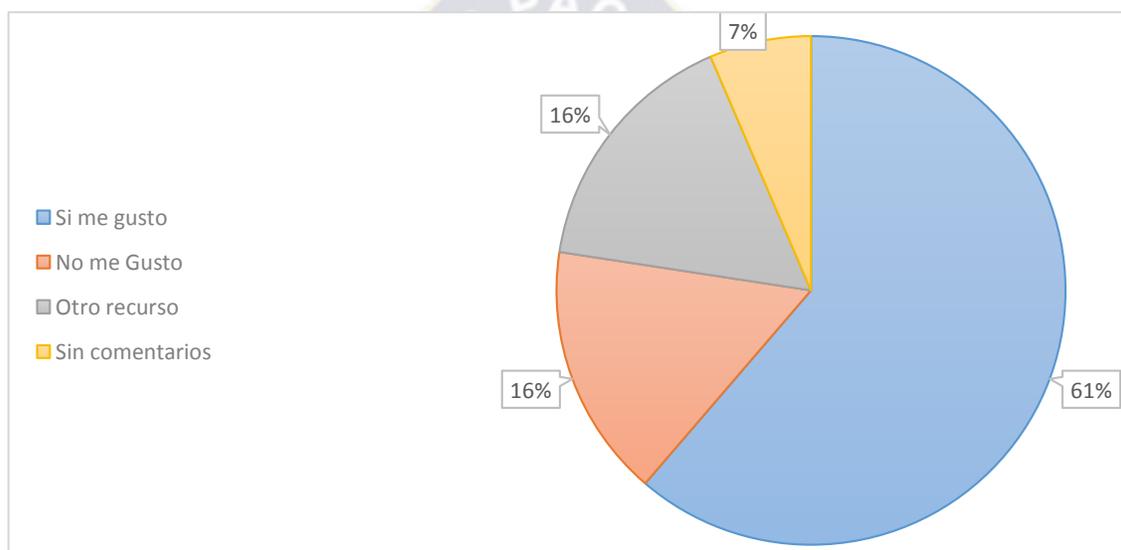


El videojuego fue del agrado de los niños ya que una mayoría mostro un 74% de participación, también se pudo que ver que existió un 10% que no mostro un interés al videojuego, un 5% no mostro tanto interés como el resto de los niños, pero si su participación al momento de probarlo.

¿Kinect fue un buen recurso para la concientización del cuidado del agua?

Tabla 36 Kinect un recurso Educativo

	Frecuencia	Porcentaje
Si me gusto	19	61 %
No me gusta	5	16 %
Otro recurso	5	16 %
Sin comentario	2	7 %
	31	100 %

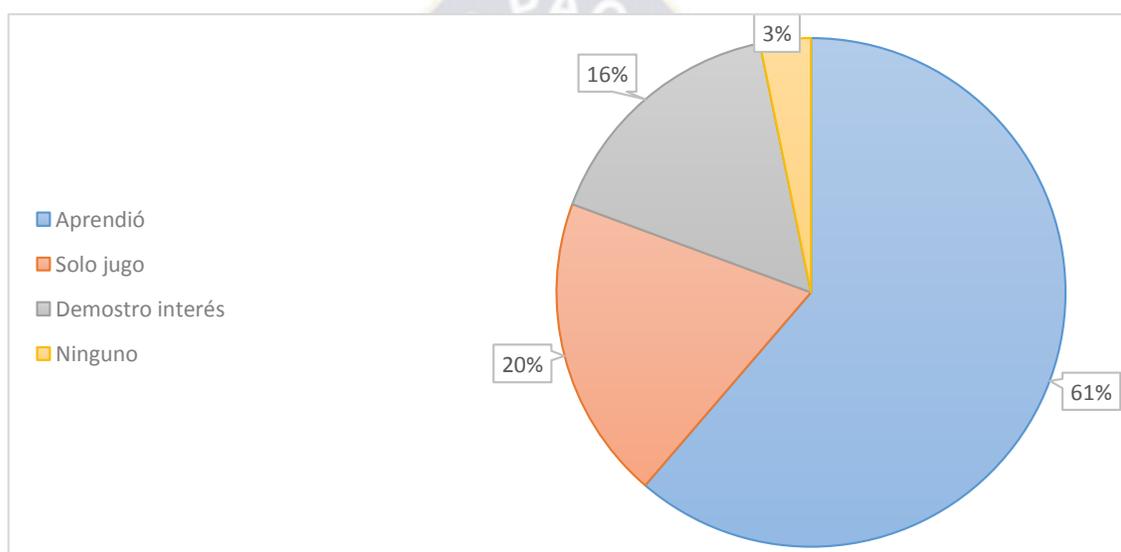


Se verifico que los niños se divirtieron al jugar con el Kinect ya que un 61% le gusto la experiencia de este recurso, un 16% no mostro interés al nuevo recurso, un 16% comento que si es videojuego podría utilizarse con otro recurso, estos mencionaron el Nintendo Wii y el Play Station. Un 7% no comento sobre el uso del Kinect como recurso.

¿Cómo fue la concientización de los niños con el videojuego?

Tabla 37 Resultados de concientización con los niños

	Frecuencia	Porcentaje
Aprendió	19	61 %
Solo jugo	6	20 %
Demostró interés	5	16 %
Ninguno	1	3 %
	31	100 %



Al finalizar el videojuego se mencionó que se podía rescatar del juego, se observó que un 61% comento sobre la importancia del agua, un 20% de los niños solo jugo y no menciono nada sobre tema, el 16% apoyo las respuesta de sus compañeros y un 3% no comento nada.

Resultados

Los datos que fueron totalmente principales, son el hecho que el videojuego cumple con las expectativas esperadas ya que se obtuvo un 55%, otro dato que resalta es el de la aceptación del videojuego ya que se obtuvo un 74% como resultado, el resultado que más impacto que se obtuvo es el de aprendizaje, ya que se obtuvo un 61% de resultado, no llegando al 75%. como se esperaba.

Así mismo se pudo observar el gran interés de los niños en estas nuevas tecnologías y su uso, también se observó el interés de los niños hacia el diseño de videojuegos, ya que muchos de ellos, comenzaron a realizar preguntas sobre el tiempo que se tomó hacerlo y si estos son difíciles de realizarlo.



Capítulo 5: Conclusiones y Recomendaciones

5.1. Conclusiones

El objetivo general que se describe en el Marco Referencial, se cumplió correctamente, ya que se utilizó una nueva tecnología para cumplir con dicho objetivo y cumpliendo con la tarea de concientizar a las personas sobre el cuidado del agua.

En cuanto a los objetivos secundarios:

El primero de ellos se cumple en el Marco Referencial, donde se realiza el análisis de distintos proyectos o tesis que hayan revisado el tema sobre el cuidado ambiental o sobre Kinect for Windows v. 1.8.

El segundo objetivo se cumple en el Marco Aplicativo dentro de la elaboración del videojuego, donde se cumplió completando un guion para realizar el seguimiento de videojuego, bajo el tema de cuidado del agua.

El tercer objetivo se cumple en el Marco Referencial: Herramientas, donde se definieron los softwares y Hardware que se utilizaron para completar el desarrollo del videojuego.

El último objetivo, se cumplió desarrollando la aplicación bajo la metodología propuesta y se han realizado las pruebas de funcionalidad, que se revisaron en el Marco Aplicativo.

5.2. Recomendaciones

Debido a que las tecnologías van en proceso de crecimiento mayor cada día y mejorando las herramientas para el desarrollo de videojuegos.

En el tiempo de selección de hardware, Microsoft presento en su Build del 2015 los lentes de holografía HoloLens, el cual promete ser una nueva herramienta dentro de la programación y realidad aumentada, aunque hasta la fecha no se tiene un precio estimado, esta herramienta promete mejorar la interacción Háptica del usuario con el equipo.

Así mismo no se descarta Kinect v2, la cual es la mejora de la Kinect v1 y que tiene un mejor reconocimiento del jugador, una mejor cámara, reconocimiento de sonido y de profundidad.

El videojuego presentado en esta tesis, solo tiene como plataforma dirigida a los equipos que tengan instalados sistemas operativos Windows 7, 8 o 10. El motor de desarrollo Unity 5 usada en el desarrollo del videojuego, permite la exportación de proyectos a otras plataformas, se recomienda por tanto que la aplicación se re factorizada para el uso en otras plataformas.

Capítulo 6: Bibliografía

- Acebo. (2012 de Agosto de 2012). *alt1040.com*. Obtenido de <http://alt1040.com/2011/03/las-ventas-del-kinect-sobrepasan-las-10-millones-de-unidades-vendidas>
- Acerenza, N., Coppers, A., Mesa, G., Viera, A., Fernández, E., Lorenzo, T., & Vallespir, D. (2010). *Una Metodología para Desarrollo de Videojuegos*. Uruguay.
- Adafruit. (10 de Noviembre de 2010). *WE HAVE A WINNER*. Obtenido de <http://www.adafruit.com/blog/2010/11/10/we-have-a-winner-open-kinect-drivers-released-winner-will-use-3k-for-more-hacking-plus-an-additional-2k-goes-to-the-eff/>
- Barchini, G. E., Sosa, M., & Herrera, S. (s.f.). *La informática como disciplina científica. Ensayo de mapeo disciplinar*.
- Berreti, F. (2014). *Navegabilidad asistida para personas con disminución visual*. Paraguay.
- Catuhe, D. (2013). *Programming With the Kinect for Windows Software Development Kit*. Readmond: Microsogt Press.
- Corresponsables/Servimedia. (18 de Marzo de 2013). *Corresponsables.com*. Obtenido de <http://www.corresponsables.com/actualidad/la-fundacion-aqualogy-lanza-una-aplicacion-movi-para-concienciar-sobre-los-problemas-de>
- Cortés, D. C. (2014). *Reconocimiento de Gestos de Manos como Mecanismo de Interacción Humano-Computador*. Bogotá D.C.
- Dios, A. G. (2014). *Desarrollo y Evaluación de un Sistema de Teleoperación de Robots basado en Reconocimiento de Gestos usando el sensor Kinect*. España.
- Duncan, G. (20 de Diciembre de 2011). *Channel 9*. Obtenido de <http://channel9.msdn.com/coding4fun/kinect/Kinect--3D--Fusion4D>
- educaLAB. (11 de Junio de 2015). *Actualidad de educaLAB*. Obtenido de <http://educalab.es/-/necesitamos-de-un-sistema-educativo-que-utilice-metodos-que-estimulen-la-creatividad-gamificacion->
- Eguía Gómez, J. L., Contreras Espinosa, R., & Solano Albajes, L. (2012). Videojuegos: Conceptos, Historia y su portencial como herramientas para la Educación. *3 Ciencias*, 14.
- Escobar, A., Quintero, D., & Serradas, D. (12 de Julio de 2006). EL Reciclaje como instrumento para la concientización de la conservación del ambiente, en el preescolar "Mi Casita de Colores".

- ESFAP. (2013). *Escuela Superior Fuerza Superior del Perú*. Obtenido de <http://www.esfap.fap.mil.pe/jetpo/docsPDF/MetodoSistemico.pdf>
- Fernández Sánchez, E. (5 de Septiembre de 2012). Control de Software Educativo Mediante Kinect. España.
- Guayas, B. P. (28 de Marzo de 2015). *Exacto, Revista Digital*. Obtenido de <http://www.exactodigital.com/noticias/guayas-expertos-de-microsoft-constatan-el-exito-del-programa-kodu-en-el-ctp>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2012). *Metodología de la Investigación*. McGraw-Hill.
- Jana, A. (2012). *Kinect for Windows SDK Programming Guide*. Birmingham, Mumbai: [PACKT] Publishing.
- Lainez Sanz, B., Alvarez Garcia, J. J., Vilches Amado, M., Alvarez Jimenez, J. M., & Palomar Sanchez, J. P. (2014). *Estrategias Didácticas para educar en el respeto al medio ambiente desde el área de Tecnología*. Ecuador.
- Luna, G., & Luna, L. (2001). *Manual de EDUCACIÓN AMBIENTAL para Comunidades de la RAAS*. Nicaragua: Edición Grafica Producciones EMCOR, s.a.
- Mendoza Lopez, P. (2010). *Educación y las nuevas Tecnologías*. Madrid.
- Ortega Carrillo. (s.f.). *Análisis crítico de los valores que transmiten los videojuegos: Descubriendo su potencial seductor de naturaleza subliminal*. Obtenido de www.ugr.es/~sevimeco/documentos/iaplicadas/a_valores.doc
- Records, G. W. (03 de Enero de 2011). www.guinnessworldrecords.com. Obtenido de <http://www.guinnessworldrecords.com/records-9000/fastest-selling-gaming-peripheral/>
- Sue, B. (2014). *Unity for Absolute Beginners*. Technology in Action.
- Thorn, A. (2014). *Learn unity for 2D Game*. Technology in Action.
- Ucedo, E. R. (2012). Implementación de estrategias de educación ambiental en la escuela superior de formación de maestros Bautista Saavedra de Santiago de Huata del departamento de La Paz. La Paz, Bolivia.
- Vallejos Fernández, D., Gonzáles Morcillo, C., Villa Alises, D., Jurado Monroy, F., Moya Fernández, F., Albusac Jiménez, J. A., . . . Gonzáles Sánchez, J. L. (2014). *Desarrollo de Videojuegos: Un Enfoque Práctico*. Madrid, España: EdLibrix.
- W. Murray, J. (2014). *C# Game Programming Cookbook for unity 3D*. CRC Press.

Wikipedia. (28 de Mayo de 2015). *Unity (Software)*. Obtenido de http://es.wikipedia.org/wiki/Unity_%28software%29

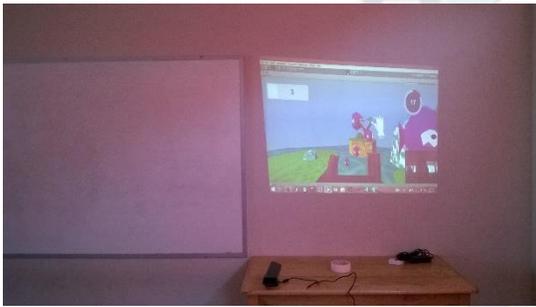
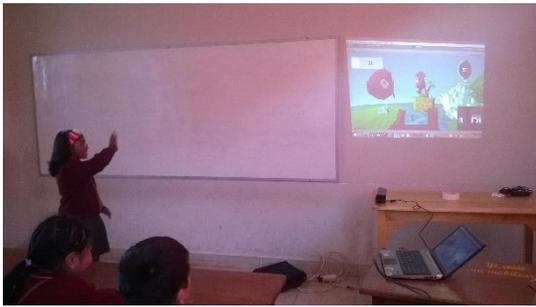
Windows, K. f. (2013). *Human Interface Guidelines v 1.8*. Microsoft Corporation.

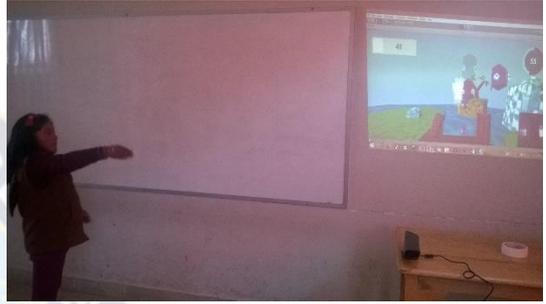
Wood, J. (28 de Junio de 2011). *Kinect Hackers Are Changing the Future of Robotics*. Obtenido de http://www.wired.com/2011/06/mf_kinect/



Capítulo 7: Anexos

Anexo 1: Evaluación del Videojuego





Anexo 2: Modelo de Negocio Canvas para Videojuegos

Modelo de Negocios Canvas				
Plataformas 1. El videojuego estara disponible para plataformas de escritorio. 2. Como una nueva fase se considera el diseño del mismo para plataformas mobiles.	Mecánica 1. Por medio de los componentes tradicionales el videojuego tendra el mismo comportamiento. 2. Con el uso del Kinect se necesita realizar una demostración de su uso. 3. El grado de dificultad v avanzado cada vez que el tiempo dentro del videojeugo se acorta.	Dinámica 1. Como dinámica propuesta de nastro videojuego se propone el uso de las nuevas tecnologías. 2. Como nueva tecnología se empleara el uso del Kinect de la consola de XBOX 360. 3. Se propone la interacción entre Videojuego y Jugador por medio del Kinect para una mejor experiencia.	Estética 1. El uso del Kinect promete llamar la atención de los usuarios. 2. La interción con el videojuego es de manera sencilla y rapida.	Jugadores 1. El videojuego esta orientado a niños de 8 a 10 años. 2. El videojuego apoyara a la concientización del cuidado del agua
	Componentes 1. Dentro del videojuego se contara con los controles comunes de juegos: contadores de puntuación, menu, contadores de tiempo. 2. Asi mismo cabe destacar cabe resaltar que gracias a la ayuda del Kinect, se utilizaran nuevos medios para el control de cada escenario.		Comportamiento 1. El jugador tendra la neccesidad de superar la puntuación maxima acumulada. 2. La necesidad de que el jugador consiga acumular más puntos. 3. la formas de acumular puntos dentro del videojuego son las mejores formas para que un videojuego tenga exito.	
Costos 1. Se utilizarn licencias gratuitas para el desarrollo de todo el videojuego. 2. Solo el costo de Kinect y el conector para la PC, seran necesarios para el desarrollo del videojuego.		Ingresos 1. Poder concientizar a los niños sobre el cuidado del agua. 2. Por medio de una evaluación sobre lo aprendido o destacado dentro del videojuego. 3. Esperamos llamar la atención de los niños asi mismo enseñar sobre el cuidado del agua.		